École doctorale
de l'École polytechnique

INRIA Sophia-Antipolis
Projet de recherche Asclepios

Processing Data in Lie Groups: An Algebraic Approach. Application to Non-Linear Registration and Diffusion Tensor MRI.

Thèse présentée pour obtenir le grade de

DOCTEUR DE L'ÉCOLE POLYTECHNIQUE

Spécialité Mathématiques et Informatique

par

Vincent ARSIGNY

Soutenue publiquement le 29 novembre 2006
devant le Jury composé de :

M. Nicholas AYACHE (Directeur de Thèse, D.R. INRIA)
Mme Isabelle BLOCH (Rapporteur, Prof. ENST)
M. Olivier FAUGERAS (Examinateur, D.R. INRIA, Ac. Sciences)
M. Jean GALLIER (Rapporteur, Prof. Univ. Pennsylvannie)
M. Stéphane MALLAT (Président, Prof. École polytechnique)
M. Xavier PENNEC (Co-Directeur, C.R. INRIA)
M. Marc SIGELLE (Invité, Maître Conf. ENST)

Research work carried out at the INRIA Sophia Antipolis - Asclepios Research Project.
2004, route des Lucioles - BP 93. 06902 Sophia Antipolis Cedex, France.

A mon épouse Angèle et mon fils Vincent-Thibault,
ainsi qu'à nos enfants futurs.

Résumé

Traitement de données dans les groupes de Lie : une approche algébrique. Application au recalage non-linéaire et à l'imagerie du tenseur de diffusion.

Ces dernières années, le besoin de cadres mathématiques rigoureux de traiter des données appartenant à des espaces non-linéaires s'est développé considérablement dans le domaine de l'imagerie médicale. Pour aborder cette question, nous nous sommes concentrés au cours de cette thèse sur la mise au point de cadres généraux pour traiter plusieurs types de données non-linéaires. Contrairement à la tendance actuelle dans notre communauté, nous ne nous sommes pas appuyés dans tous les cas sur la géométrie riemannienne, qui peut être coûteuse en temps de calcul ou bien encore ne pas avoir certaines propriétés souhaitables. Au lieu de cela, nous avons généralement employé des approches de type *algébrique*, c'est-à-dire des approches basées sur les propriétés algébriques des espaces non linéaires que nous avons considérés, qui sont tous des groupes de Lie.

Tout d'abord, nous présentons dans ce travail un cadre de traitement général pour les matrices symétriques et définies positives (également appelées ici “tenseurs” par abus de langage). Ce cadre, nommé *log-euclidien*, est très simple à utiliser et a d'excellentes propriétés théoriques. En particulier, ce cadre permet de mener des calculs invariants par similitude et par inversion matricielle, et la moyenne log-euclidienne est une généralisation aux tenseurs de la moyenne géométrique des nombres strictement positifs. L'utilisation de ce cadre puissant est illustrée dans le cas du traitement des images de tenseurs de diffusion obtenues par résonance magnétique.

En second lieu, nous avons proposé plusieurs cadres, appelés *polyaffines*, pour paramétrer avec un nombre restreint de degrés de liberté flexibles des transformations localement rigides ou affines, d'une manière qui garantit leur inversibilité et assure d'excellentes propriétés théoriques. L'utilisation de ces cadres est exemplifié avec succès dans le cas du recalage localement rigide de coupes histologiques et du recalage 3D localement affine d'IRM du cerveau humain. De manière remarquable, le problème de la fusion de plusieurs transformations linéaires locales en une transformation globale inversible est étroitement lié au problème du calcul d'une valeur moyenne dans les groupes de Lie des transformations rigides ou affines.

Ceci nous a menés à proposer deux cadres généraux originaux pour le calcul de statistiques dans les groupes de Lie en dimension finie : d'abord le cadre log-euclidien, qui généralise notre travail sur les tenseurs. Et en second lieu, un cadre basé sur la notion nouvelle de *moyenne bi-invariante*, dont les propriétés d'invariance généralisent aux groupes de Lie celles de la moyenne arithmétique dans les espaces euclidiens. Sont présentés ici une théorie générale de la moyenne bi-invariante ainsi que ses propriétés remarquables dans un certain nombre de cas spécifiques (en particulier les transformations rigides).

Enfin, nous avons généralisé notre cadre log-euclidien aux déformations géométriques régulières et inversibles (c'est-à-dire les difféomorphismes) afin de fournir une manière simple de calculer des statistiques sur ce type spécifique de données. Des résultats préliminaires prometteurs sont présentées à la fin de ce travail, ce qui ouvre la voie à un cadre général et cohérent pour les statistiques en anatomie computationnelle.

Abstract

In recent years, the need for rigorous frameworks to process data belonging to non-linear spaces has grown considerably in the medical imaging community. To address this issue, we have focused during this thesis on proposing novel and general frameworks to process several types of data living in non-linear spaces. Contrary to the current trend in our community, we have not relied in all cases on Riemannian geometry, which can either be quite computationally expensive or even lack desirable properties in a number of situations. Instead, we have generally used *algebraic-oriented* approaches, i.e. approaches rooted in the algebraic properties of the non-linear spaces we have considered, which are all Lie groups.

First, we present in this work a general processing framework for symmetric positive-definite matrices (also called here "tensors" by abuse of language). This framework, named *Log-Euclidean*, is very simple to use and has excellent theoretical properties. In particular, it allows to perform computations which are invariant with respect to similarities and matrix inversion, and the Log-Euclidean mean is a generalization to tensors of the geometric mean of positive numbers. The use of this powerful framework is exemplified in the case of diffusion tensor MRI.

Second, we have proposed several frameworks, called *polyaffine*, to parameterize with a small number of flexible degrees of freedom locally rigid or affine transformations, in a way that guarantees their invertibility and ensures as many natural properties as possible. The use of these frameworks is exemplified successfully with the locally rigid registration of histological slices and with the locally affine 3D registration of MR scans of the human brain. Interestingly, fusing several local linear transformations into a global invertible transformation is closely linked to the problem of computing a mean value in the Lie groups of rigid or affine transformations.

This has led us to propose two novel general frameworks for computing statistics in finite-dimensional Lie groups: first a Log-Euclidean framework, which generalizes our work on tensors. Second, a framework based on the novel notion of *bi-invariant mean*, whose invariance properties generalize to Lie groups those of the arithmetic mean in Euclidean spaces. A general theory of the bi-invariant mean as well as its remarkable properties in a number of specific cases (in particular rigid transformations) are presented.

Finally, we have generalized our Log-Euclidean framework to invertible geometrical deformations (i.e., diffeomorphisms) in order to provide simple tools to compute statistics on this special type of data. First experiments showing promising results are presented briefly at the end of this work, which opens the way to a consistent integrative framework for statistics in computational anatomy.

Remerciements

Je remercie en tout premier lieu ma famille, et plus particulièrement mon épouse Angèle et mon fils Vincent-Thibault pour leur soutient permanent. Sans eux, ce travail de thèse n'aurait pas été posssible.

J'adresse mes chaleureux remerciements à mon directeur de thèse, Nicholas AYACHE, qui m'a accueilli avec enthousiasme dans son équipe et m'aura permis de m'investir dans des projets scientifiques passionnants au plus haut niveau de la recherche mondiale. Merci à lui d'avoir assurré mon encadrement, ainsi qu'à Xavier PENNEC, qui a co-encadré cette thèse, et qui m'a initié à la géométrie Riemannienne.

Bien sûr, je remercie très cordialement les membres de mon jury de thèse, qui m'ont fait l'amitié de consacrer du temps à l'examen attentif de mes travaux. Merci tout particulièrement à Isabelle BLOCH et Jean GALLIER d'avoir assumé la lourde tâche de rapporteurs. Merci à Stéphane MALLAT de m'avoir fait l'honneur d'être le Président de mon jury, et à Olivier FAUGERAS et Marc SIGELLE d'avoir accepté d'en être les examinateurs.

Merci également aux membres présents et passés du projet de recherche ASCLEPIOS, pour ces années vécues dans une ambiance généralement chaleureuse. Ma gratitude va tout particulièrement à ceux qui auront eu la gentillesse d'assurer une part de ma formation en se montrant disponibles pour répondre à mes multiples questions : à ce sujet, merci tout spécialement à Marc TRAINA pour son aide précieuse concernant divers problèmes d'ordre informatique. Un grand merci en outre à Pierre FILLARD et Olivier COMMOWICK, pour nos très fructueuses collaborations.

Je tiens aussi à rendre hommage aux instances dirigeantes de l'INRIA, qui m'ont permi d'effectuer mes années de doctorat dans un environnement particulièrement stimulant, et d'excellentes conditions. Ma gratitude va en particulier à M. Gilles KAHN, décédé si prématurément en 2006, et qui s'était personnellement déplacé en 2002 à l'ENST pour présenter l'INRIA à ma promotion d'ingénieurs. Tous mes remerciements également aux autres P.-D.G. de l'INRIA durant mon séjour à l'institut : MM. Bernard LAROUTUROU et Michel COSNARD.

Il est bien évident que je n'aurai pu parvenir à un tel niveau d'études supérieures sans l'exemple ou les encouragements d'un certains nombre d'enseignants dont l'influence a été pour moi déterminante. Citons parmi eux en particulier M. Alain CAMBROUZE, professeur de Mathématiques au Lycée Saint Louis, M. Stéphane MALLAT, professeur à l'Ecole polytechnique et M. Yves MEYER, professeur à l'Ecole nomale supérieure de Cachan.

Contents

1 Introduction **1**
1.1 Data Processing in Lie Groups and Medical Imaging 1
1.1.1 Tensor Processing 2
1.1.2 Parameterization of Geometrical Transformations 5
1.1.3 Statistics on Invertible Linear Transformations 9
1.1.4 Statistics on Diffeomorphisms 10
1.2 Manuscript Organization 11
1.3 Global Picture 13

2 Fundamental Mathematical Tools **15**
2.1 Notations 16
2.2 First Order Ordinary Differential Equations 17
2.2.1 Definition 17
2.2.2 Life Span of a Solution to an ODE 17
2.2.3 Flow of a First Order ODE 18
2.2.4 One-Parameter Subgroups 18
2.3 Lie Groups 19
2.3.1 Definition of Lie groups 19
2.3.2 Examples 19
2.3.3 Lie Algebra and Adjoint Representation 20
2.3.4 Matrix Exponential and Logarithm 21
2.3.5 Lie Group Exponential and Logarithm 21
2.3.6 Baker-Campbell-Hausdorff Formula 23
2.4 Riemannian Geometry 23
2.4.1 Riemannian Metrics 25
2.4.2 Riemannian Geodesics 25
2.4.3 Riemannian Exponential and Logarithm 26
2.4.4 Fréchet Mean 26

3 Log-Euclidean Metrics on Tensors **29**
3.1 Introduction 31
3.2 Preliminaries 32
3.2.1 Differential Properties of Matrix Exponential 33
3.2.2 Algebraic Properties of SPD Matrices 33
3.2.3 Differential Properties of SPD Matrices 34
3.2.4 Compatibility Between Algebraic and Differential Properties 34
3.3 Log-Euclidean Means 35
3.3.1 Multiplication of SPD Matrices 35
3.3.2 Log-Euclidean Metrics on the Lie Group of SPD Matrices 37

3.3.3 A Vector Space Structure on SPD Matrices . . . 39
3.3.4 Log-Euclidean Mean . . . 39
3.4 Comparison with the Affine-Invariant Mean . . . 40
3.4.1 Elementary Metric Operations and Invariance . . . 40
3.4.2 Affine-Invariant Means . . . 40
3.4.3 Geometric Interpolation of Determinants . . . 41
3.4.4 Criterion for the Equality of the Two Means . . . 42
3.4.5 Larger Anisotropy in Log-Euclidean Means . . . 42
3.4.6 Linear and Bilinear Interpolation of SPD Matrices . . . 45
3.5 Probabilities and Statistics with Log-Euclidean Metrics . . . 46
3.5.1 General Riemannian Statistical Framework . . . 47
3.5.2 Random Tensors . . . 47
3.5.3 Fréchet Means and Covariances with Log-Euclidean Metrics . . . 48
3.5.4 General Log-Euclidean Statistical Framework . . . 50
3.6 Conclusions . . . 50

4 Log-Euclidean Processing of Diffusion Tensors 53
4.1 Introduction . . . 54
4.1.1 The Defects of Euclidean Calculus . . . 55
4.1.2 Riemannian Metrics . . . 55
4.2 Theory . . . 56
4.2.1 Matrix Exponential, Logarithm and Powers . . . 56
4.2.2 Definition of Log-Euclidean Metrics . . . 57
4.2.3 Invariance Properties of Log-Euclidean Metrics . . . 58
4.2.4 Log-Euclidean Computations on Tensors . . . 58
4.2.5 Comparison of the Affine-Invariant and Log-Euclidean Frameworks . . . 59
4.3 Methods . . . 60
4.3.1 Interpolation . . . 60
4.3.2 Regularization . . . 60
4.3.3 Absolute Value of a Symmetric Matrix . . . 61
4.3.4 Materials . . . 61
4.4 Results . . . 62
4.4.1 Interpolation . . . 62
4.4.2 Regularization . . . 64
4.5 Discussion and Conclusions . . . 69
4.5.1 The Defects of Euclidean Calculus . . . 69
4.5.2 Conclusions and Perspectives . . . 70

5 Original Polyrigid and Polyaffine Transformations 71
5.1 Introduction . . . 72
5.2 Theory of Polyrigid and Polyaffine Transformations . . . 73
5.2.1 Regions of Influence and Interpolation of Sparse Data . . . 73
5.2.2 A Framework with ODEs . . . 75
5.2.3 Theoretical Properties of Polyrigid Transformations . . . 76
5.2.4 Extension to Polyaffine Transformations . . . 76
5.2.5 Summary of the properties of Polyrigid Transformations . . . 77
5.3 Implementation of Polyrigid Transformations . . . 78
5.3.1 Discretization Schemes . . . 78
5.3.2 Implementation with the Insight Toolkit . . . 79

5.4 Registration of Histological Slices 81
5.4.1 Object of the Study and Experimental Setup 81
5.4.2 Limitations of the First-Order Gradient Descent. 84
5.4.3 Registration Results using a Levenberg-Marquardt Algorithm 88
5.4.4 Alternating Optimization 88
5.5 Results with more Complex Regions 90
5.5.1 The Shape of the Regions of Influence. 90
5.5.2 Results Obtained with Three Anchor Points 90
5.6 Conclusions and Perspectives 93
5.7 Appendix: First Derivatives of Original Polyrigid Transformations 94

6 Log-Euclidean Polyaffine Transformations 97
6.1 Introduction 98
6.2 A Log-Euclidean Polyaffine Framework 99
6.2.1 Previous Polyaffine Framework 99
6.2.2 Simpler Velocity Vectors for Affine Transformations 103
6.2.3 Log-Euclidean Polyaffine Transformations 104
6.3 Fast Polyaffine Transform 107
6.3.1 Matrix Exponential and the ‘Scaling and Squaring’ Method 111
6.3.2 A ‘Scaling and Squaring’ Method for LEPTs 111
6.3.3 2D Synthetic Experiments 114
6.3.4 3D Registration Example 122
6.4 A Log-Euclidean Framework for Rigid and Affine Transformations 125
6.4.1 Log-Euclidean Metrics 126
6.4.2 Invariance Properties 127
6.4.3 Regularization of Locally Affine Transformations 128
6.4.4 Log-Euclidean Framework for General Lie Groups 129
6.4.5 Numerical Implementation of Matrix Logarithm. 131
6.5 Conclusion and Perspectives 132

7 Bi-Invariant Means in Lie Groups 133
7.1 Introduction 135
7.2 Means in Lie Groups 136
7.2.1 Means and Algebraic Invariance 136
7.2.2 Bi-Invariant Fréchet Means via Invariant Metrics in Lie Groups 137
7.2.3 Absence of Bi-Invariant Metrics for Rigid Transformations 139
7.3 Advanced Properties of the Exponential and Logarithm 141
7.3.1 Preliminary Result 141
7.3.2 Group Geodesics. 141
7.4 Bi-Invariant Means in Lie Groups 143
7.4.1 A Geometric Definition of the Mean 143
7.4.2 Stability of the Classical Iterative Scheme 144
7.4.3 Convergence: Special Case 146
7.4.4 Convergence: General Case 147
7.4.5 Higher Order Moments 148
7.5 Bi-Invariant Means in Simple Cases 149
7.5.1 Bi-Invariant Mean of Two Points 149
7.5.2 Scalings and Translations in 1D 150
7.5.3 The Heisenberg Group 152

7.5.4 On a Subgroup of Triangular Matrices . . . 154
7.6 Linear Transformations . . . 155
7.6.1 General Rigid Transformations . . . 155
7.6.2 2D Rigid Transformations . . . 159
7.6.3 General Linear Transformations . . . 160
7.6.4 Tensors . . . 161
7.7 Left-Invariant Polyaffine Transformations . . . 161
7.7.1 Polyaffine Transformations . . . 161
7.7.2 A Novel Type of Polyaffine Transformations . . . 162
7.8 Conclusions and Perspectives . . . 165

8 Statistics on Diffeomorphisms in a Log-Euclidean Framework 167
8.1 Introduction . . . 168
8.2 A Log-Euclidean Framework for Diffeomorphisms . . . 169
8.3 Computation of Exponentials and Logarithms . . . 171
8.4 Statistics on 3D Diffeomorphisms . . . 173
8.5 Conclusions and Perspectives . . . 175

9 Conclusions and Perspectives 177
9.1 Contributions and Publications . . . 177
9.1.1 General Log-Euclidean Framework for Tensors . . . 178
9.1.2 Diffusion Tensor Processing . . . 179
9.1.3 Structure Tensor Processing . . . 179
9.1.4 Statistics on Anatomical Variability via Tensors . . . 180
9.1.5 Statistics on Deformation Tensors in Non-Linear Registration . . . 181
9.1.6 Polyrigid and Polyaffine Transformations . . . 182
9.1.7 Statistics in Finite-Dimensional Lie Groups . . . 184
9.1.8 Statistics on Diffeomorphisms . . . 185
9.2 Short Term Perspectives . . . 185
9.2.1 Tensor Processing . . . 185
9.2.2 Locally Rigid or Affine Transformations . . . 186
9.2.3 General Statistics in Lie Groups . . . 187
9.2.4 Statistics on Diffeomorphisms . . . 187
9.3 More Global Perspectives . . . 187
9.3.1 Natural Extensions of this Work . . . 187
9.3.2 Bi-Invariant Framework vs. Riemannian Geometry . . . 190
9.3.3 Medical Image Registration . . . 191
9.4 Epilogue . . . 191

Chapter 1

Introduction

Contents

1.1 Data Processing in Lie Groups and Medical Imaging 1

1.1.1 Tensor Processing 2

1.1.2 Parameterization of Geometrical Transformations 5

1.1.3 Statistics on Invertible Linear Transformations 9

1.1.4 Statistics on Diffeomorphisms 10

1.2 Manuscript Organization 11

1.3 Global Picture 13

1.1 Data Processing in Lie Groups and Medical Imaging

In recent years, the need for rigorous frameworks to process data belonging to non-linear spaces has grown considerably in the medical imaging community.

First, a number of imaging modalities, like diffusion tensor MRI (DTI or DT-MRI) [Le Bihan 91, Basser 02, Özarslan 03], provide researchers and clinicians with data which do not belong to a linear space, and nonetheless require post-processing (re-sampling, regularization, statistics, etc.). See for instance [Lenglet 06b] for examples of Riemannian statistics on diffusion tensors.

Second, the one-to-one registration of medical images naturally deals with data living in non-linear spaces, since many types of invertible geometrical deformations belong to *groups* of transformations, which are not vector spaces. Some of these groups are finite-dimensional, as in the case of rigid or affine transformations; see for instance [Boisvert 06] for statistics on rigid transformations, in the context of the analysis of the statistical properties of the human scoliosis. Some groups of transformations relevant in medical imaging can also be infinite-dimensional, as in the case of groups of diffeomorphisms parameterized with time-varying velocity vector fields [Trouvé 98].

During our Ph.D. thesis, we have focused on proposing novel and general mathematical frameworks to rigorously process several types of data living in non-linear spaces. Firstly, we have sought to define a general processing framework for symmetric positive-definite matrices, often referred to as 'tensors' by abuse of language in this work. Secondly, we have concentrated on parameterizing invertible geometrical transformations with a small number of flexible degrees of freedom, with a local rigid or affine behavior. This is closely related to the question of computing a mean value in the Lie groups of rigid or affine transformations, which is a topic we have investigated in detail. This has led to the generalization to linear transformations of our framework for tensors, as well as the definition of a novel general notion of mean in Lie groups, called the bi-invariant mean. Last but not least, we have generalized our ideas on tensors to *general* (i.e., non-linear) invertible geometrical deformations, in order to provide *simple* tools to compute statistics on this special type of data.

The various types of non-linear data we have been studying have one feature in common: they all belong to mathematical spaces which are called *Lie groups*. These very remarkable entities are both smooth manifolds (i.e. nearly identical to vector spaces locally, but possibly very 'curved' globally) and algebraic groups (i.e. are endowed with a nice 'multiplication' operation between elements of the space).

One should note that a classical and very general framework to process data in smooth manifolds such as Lie groups is provided by *Riemannian geometry*, which has been increasingly used in our community in recent years (see Chapter 2, Section 2.4 for more details on this topic). However, contrary to the current trend, we have not relied in all cases on Riemannian geometry. As we have found in this work, this remarkable framework can either be quite computationally expensive or even lack desirable properties in a number of situations. Instead, we have generally used *algebraic-oriented* approaches, i.e. approaches rooted in the algebraic properties of the non-linear spaces we have considered.

Essentially, we have followed this original approach in order to find a satisfying balance between the remarkable mathematical properties of our frameworks and their simplicity, which is of paramount importance from a practical point of view. Fortunately, these two antagonistic characteristics can sometimes be relatively easy to conciliate: our Log-Euclidean framework for tensors presented in Chapter 3 is particularly striking in this respect.

In the rest of this Section, we detail the different situations of data processing we have considered, as well as their respective importance in medical imaging.

1.1.1 Tensor Processing

Symmetric positive-definite matrices (or SPD matrices) of real numbers, also called by abuse of language 'tensors' in this work, appear in many contexts, as illustrated in Fig. 1.1.

In medical imaging, their use has become common during the last ten years with the growing interest for *Diffusion Tensor* Magnetic Resonance Imaging (DT-MRI or simply DTI) [Le Bihan 91]. In this imaging technique based on Nuclear Magnetic Resonance (NMR), the assumption is made that the random diffusion of water molecules at a given position in a biological tissue is Gaussian. As a consequence, a diffusion tensor image is an SPD matrix-valued image in which each volume element (or voxel) is the covariance matrix of the local diffusion process. *Variability tensors* also provide a powerful framework to model the local anatomical variability of organs such as the brain, as shown in [Fillard 05c]. More generally, *structure tensors* are widely used in image analysis, especially for segmentation, grouping, motion analysis and texture segmentation [Broxand 03], and can also be used in certain regularization techniques for medical image registration [Clatz 05,Stefanescu 04]. Tensors are also used intensively in mechanics, for example with stress or deformation tensors [Salencon 01]. *Deformation tensors* can be also used to inject priors in non-linear medical image registration as in [Commowick 05,Pennec 05]. Moreover, *metric tensors* are becoming a common tool in numerical analysis to generate adapted meshes to reduce the computational cost of solving partial differential equations (PDEs) in 3D [Mohammadi 97].

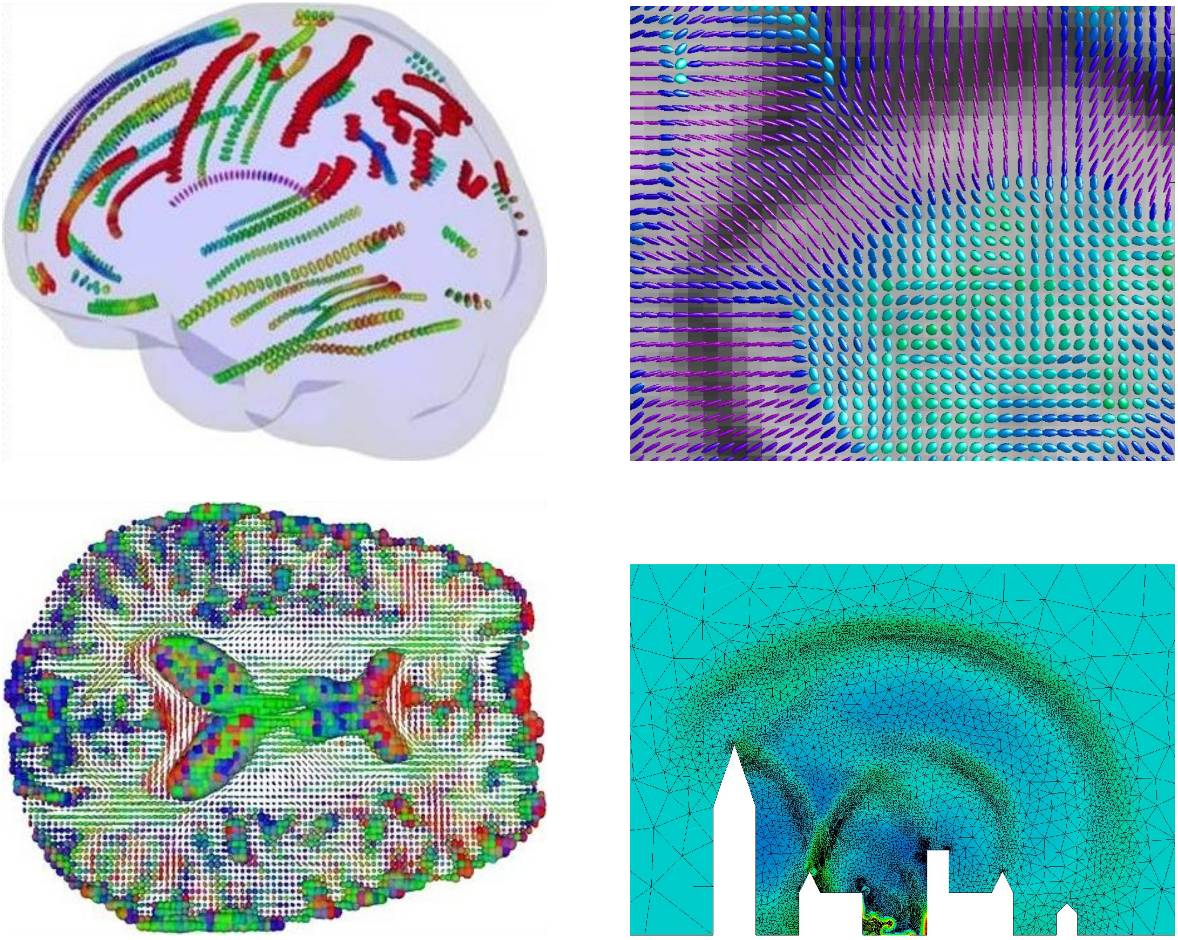

Figure 1.1: **Examples of the use of tensors. Top left: variability tensors** describing the local variability of the human brain, sampled along sulcal lines on the cortex [Fillard 05c]. **Top right: structure tensors** in a 2D slice extracted from a MR scan of the brain in [Clatz 05]. Structure tensors describe the local structure of intensity variations in images; here, the intensity of the image is displayed in grey levels below tensors. **Bottom right: diffusion tensors** describing the local characteristics of the diffusion of water molecules in the brain, on a slice of diffusion tensor MRI (DT-MRI). **Bottom left:** adapted mesh generated via the use of **metric tensors** to solve more efficiently a wave propagation partial differential equation [Alauzet 03]. In three of these images, a tensor S is represented via a translated version of the ellipsoid of equation $x^T.S^{-2}.x = 1$ and is colored according to the direction of its first eigenvector.

As a consequence, there has been an increasing need to carry out computations on these objects, for instance to interpolate, restore, enhance images of symmetric positive-definite matrices. To this end, one needs to define a complete operational framework. This is necessary to fully generalize to the SPD case the usual statistical tools or PDEs on vector-valued images. The framework of Riemannian geometry [Gallot 93], briefly presented in Chapter 2, is particularly well-adapted to this task, since many statistical tools [Pennec 06a] and PDEs available for vectors can be generalized to non-linear spaces in this framework.

One can directly use a Euclidean structure on square matrices to define a metric on the space of SPD matrices. This is straightforward, but unfortunately, although Euclidean distances are well-adapted to general square matrices, they are unsatisfactory for tensors, which are very specific matrices, for two main reasons.

First, symmetric matrices with null or negative eigenvalues typically appear on clinical DT-MRI data as soon as we perform on tensors Euclidean operations which are non-convex. Example of such situations are the estimation of tensors from diffusion-weighted images, the regularization of tensors fields, etc. To avoid obtaining non-positive eigenvalues, which are difficult to interpret physically, it has been proposed to regularize only *features* extracted from tensors, like first eigenvectors [Coulon 04] or orientations [Chefd'hotel 04]. This is only partly satisfactory, since such approaches do not take into account *all* the information carried by tensors.

Second, an SPD matrix corresponds typically to a *covariance matrix*, and the value of its determinant is a direct measure of the dispersion of the associated Gaussian multivariate random variable (MRV). Indeed, for a given MRV, we know with a confidence say of 95% that a sample of the MRV will be located within a region called a *confidence region*, which is the multidimensional equivalent of a confidence interval. For Gaussian MRVs, the confidence regions are ellipsoids, and the volumes of these ellipsoids are proportional to the square root of the *determinant* of the covariance matrix. But the Euclidean averaging of SPD matrices leads often to a *swelling effect*: the determinant of the Euclidean mean can be strictly larger than the original determinants. The reason is that the induced interpolation of determinants is polynomial and *not* monotonic in general. In DTI, diffusion tensors are assumed to be covariance matrices of the local Brownian motion of water molecules. Introducing more dispersion in computations amounts to introducing more diffusion, which is physically unacceptable. For illustrations of this effect, see [Feddern 04, Chefd'hotel 04]. As a consequence, the determinant of a mean of symmetric positive-definite matrices should remain bounded by the values of the determinants of the averaged matrices.

To fully circumvent these difficulties, more sophisticated Riemannian metrics have been recently proposed for tensors, called affine-invariant [Pennec 06b, Fletcher 04a, Lenglet 06a, Moakher 05]. With these metrics, symmetric matrices with negative and null eigenvalues are at an infinite distance from any tensor. The swelling effect is not present, and the symmetry with respect to inversion is respected. But the price paid for this success is a high computational burden in practice, essentially due to the curvature induced on the space of symmetric positive-definite matrices. This leads in many cases to slow and hard to implement algorithms (especially for PDEs) [Pennec 06b]. We have proposed in this work a *simpler and less computationally expensive alternative* to the affine-invariant framework, while preserving as many natural properties as possible. Our framework, called *Log-Euclidean*, is presented in Chapter 3 and its application to the processing of diffusion tensors is described in Chapter 4. Interestingly, our approach is based on a *novel Lie group structure for tensors*, which can be smoothly extended into a novel vector space structure *compatible* with many of the usual *algebraic properties* of this remarkable matrix space.

1.1.2 Parameterization of Geometrical Transformations

In Chapters 5 and 6, we focus on the parameterization of locally rigid or affine geometrical deformations with a small number of flexible degrees of freedom. In order to emphasize the relevance of this type of parameterization of geometrical deformations in medical imaging, we will first briefly present the general problem of medical image registration and the different types of parameterization of transformations which exist in the literature. For more details on medical image registration, the interested reader can refer to the reviews of medical image registration of [Hill 01] and [Maintz 98].

Medical Image Registration. In medical imaging, one of the most fundamental problems is that of finding *correspondences* between some *representations* of two (or more) anatomies (i.e. images, volumes, surfaces, sets of lines or points, etc.). This is very useful to compare anatomies, for example to analyze the anatomical differences existing between normal and pathological subjects, or to quantify the evolution of a disease, or to build *anatomical atlases*, i.e. models incorporating various types of prior knowledge about a particular type of anatomy (i.e., an organ or a set of organs, etc.). In this context, the general process of finding corresponding points is called *registration*. In many cases, correspondences are obtained via a *geometrical transformation*: most registration algorithms look for an optimal way of geometrically 'deforming' one of the representations into the others. We only consider this quite general case in this work.

If the representations to be registered correspond to the *same* anatomy (i.e. come from the same subject), possibly imaged with two different modalities, the registration is referred to as *intra-subject*. In this case, assuming that no deformations or changes have been introduced between the acquisitions of these two representations (e.g., in the case of the brain, no surgical operation has been carried out, and little time has elapsed between the acquisitions, since aging modifies the size of the ventricles, etc.), then true correspondences exist, and can be found only via the search of the true *rigid transformation* between the two representation.

But in the case of representations from *different* anatomies, the registration is called *inter-subject*, and the problem of finding 'good' correspondences between these anatomies is much more difficult, possibly meaningless in certain areas (e.g., some folding patterns of the cortex of the human brain, called *sulci*, exist for certain individuals and not in others, see [Rivière 02, Rivière 00]). In the inter-subject case, a rigid registration is obviously not sufficient between most anatomies (e.g., human brains, livers or bones vary considerably in size and shape). Thus, existing algorithms resort to *non-rigid* registration: they non-rigidly deform one anatomy to fit the other. Here, a crucial issue is certainly the *amount of non-rigidity* introduced by the algorithm. On the one hand, the fewer the *parameters* or equivalently *degrees of freedom* (DOFs), the smaller the risk of introducing artifactual deformations during the registration. But on the other hands, many DOFs are needed to take into account the fine variations that can exist from one anatomy to another. Finding an adequate number of adapted DOFs for a specific application can thus be quite a challenging task, and the question of the *parameterization* of the correspondences to be established during a registration procedure is therefore central in medical image registration. Chapters 5 and 6 of this thesis are mainly devoted to this topic.

Other key issues are the following: how can one quantify the 'goodness' of a given set of correspondences? For a chosen measure of goodness, how can one compute a transformation optimizing this goodness? Also, what should we expect *a priori* in terms of deformations, i.e. are some of them more likely than others? Thus, in the field of medical image registration, much effort is still currently devoted to the following problems:

- Defining proper **parameterizations** of the classes of geometrical transformations used for medical image registration. This topic will be detailed below.

- Finding adequate **measures of similarity** (or *similarity criteria*) between anatomies, which are scalar-valued functions of the representations of the anatomies and of the correspondences found between them. Intuitively, similarity measures quantify the 'goodness' of correspondences with respect to the registered anatomies. See for example [Holden 00] for a comparison between 8 similarity measures in the context of the rigid registration of 3D MR brain scans.

- Designing efficient **optimization methods**, adapted to the similarity measure and the parameterization used. For example, in Chapter 5, we use a Levenberg-Marquardt gradient descent strategy to optimize a image similarity measure called the 'sum of squared differences' (SSD) with respect to the parameters of a polyrigid transformation. Completely different optimization methods can be used depending on the circumstances. For instance, linear algebraic techniques can be used to compute optimal solutions in closed forms in the case of the rigid or affine image registration algorithm based on block-matching presented in [Ourselin 00]. Whereas in situations where the gradient of the simularity measure is too expensive to compute, methods such as Powell's multidimensional direction set method can be relied on, as in the case of the maximization of the mutual information to rigidly register 3D images of different modalities [Maes 97].

- Defining relevant **priors** on correspondences, depending on the precise type of registration considered. Indeed, in most non-linear registration algorithms, some kind of *regularization* technique is used to prevent irregular deformations from being obtained, which are implicitly assumed to be anatomically very unlikely. Currently, a substantial effort is being devoted to finding efficient ways of using *statistical* priors on deformations, instead of arbitrary mathematical priors chosen for purely technical reasons. See [Pennec 05, Commowick 05, Stefanescu 04] for two examples of strategies to inject statistical priors in non-linear registration. Finally, one should note that in the case of a single anatomy which has been mechanically deformed (for example during a surgical operation), it makes most sense to rely on realistic mechanical models to constrain deformations, as done in [Clatz 05].

Parameterization of Geometrical Deformations. We now present the various types of parameterization that exist in the literature of medical image registration, since this specific topic is at the heart of the results presented in Chapters 5 and 6.

At the beginning of the spectrum, we have simple parametric transformations such as rigid or affine transformations, which have a very small number of degrees of freedom, and can be efficiently used for intra-patient registration.

For both inter-subject or intra-subject registration, in order to introduce more DOFs, geometrical deformations are often parameterized with splines or radial basis functions (RBFs), such as B-Splines [Rueckert 99], Thin-Plate-Splines [Bookstein 99], Geodesic Interpolating Splines [Camion 01], or more recently Polyharmonic Clamped-Plate Splines [Marsland 03] and radially symmetric basis functions [Rohde 03]. An alternative is provided by finite elements models, as in [Ferrant 99, Ashburner 99] or more recently [Clatz 05] in the context of the use of a mechanical deformable model of the human brain. Another way of introducing more DOFs is to model deformations locally with a few intuitive parameters which can code for a large variety of local deformations such as translations, rotations, swelling, etc. Different types of locally rigid or affine deformations have been proposed (also called 'piecewise' rigid or affine), as in [Little 96], which was recently used in [Pitiot 06] in the context of the registration of histological slices, in [Hellier 01] in an adaptative hierarchical framework for general medical image registration, and in [du Bois d'Aische 05] for the locally rigid registration of articulated structure. The family of novel transformations described in this thesis, called *polyrigid and polyaffine transformations*, belong to this category, and have inspired another variants of such transformations, such as the other local rigid or affine parameterization of deformations presented in [Narayanan 05] and the local parameterization of articulated rigid structures of [Papademetris 05]. Another type of local parameterization derived from the principles of fluid mechanics and based on 'vortex particles', limited to 2D deformations, was also presented in [Cuzol 05]. Interestingly, one should also note that a number of authors increase locally and adaptively the number of DOFs during the registration procedure. This can be done for example with B-splines [Noblet 05, Park 03, Schnabel 01], RBFs [Rohde 03] or local affine components [Hellier 01].

At the end of the spectrum, *dense deformation fields* defining a displacement at every voxel [Thirion 98, Cachier 02, Cachier 03, Chefd'hotel 02, Hermosillo 02, Stefanescu 04], and *diffeomorphisms* generated by time-varying vector fields defining a velocity vector at every voxel and at each (discretized) instant between times 0 and 1 [Trouvé 98, Trouvé 00, Camion 01, Beg 03, Miller 03, Guo 04, Joshi 00, Beg 05, Trouvé 05b, Glaunes 04, Allassonnière 05] exhibit the highest number of degrees of freedom, and can be used for inter-subject registration.

Each of the above-mentioned transformations has its particular domain of application. In the case of anatomical structures incorporating rigid elements (such as bone articulations, or structures which are subject to simple local deformations, like histological slices), rigid and affine transformations clearly do not have enough degrees of freedom. On the contrary, deformation fields have too many DOFs and thus can be easily misled by local minima of the similarity criterion. For the existing intermediate transformations, e.g. B-Splines, the degrees of freedom of the transformation are not really adapted since many control points are required to reconstruct several locally rigid behaviors, especially when rotations are substantial. Clearly, in this case or in the case of structures which are subject to simple local deformations, locally rigid or affine transformations like the ones described in Chapter 5 and 6 are particularly well adapted. Fig. 1.2 illustrates the interest of this type of geometrical deformations in the case of the correction of the posture of the human head and neck.

Guaranteeing the Invertibility of Geometrical Deformations. A desirable property of geometrical deformations between anatomies is *invertibility*, which is not guaranteed by many algorithms based on the parameterizations mentioned above. Although there is no reason to believe that a *true* one-to-one correspondence exists between *any* two anatomies, guaranteeing the invertibility of geometrical deformations between two anatomies constitutes a powerful safeguard against nonsensical results (e.g., transformations with numerous local foldings) and uncontrolled changes of topology. In the absence of a satisfying way of modeling and evaluating the local uncertainty of correspondences between two anatomies, it makes good sense to guarantee the invertibility of the transformations used for the inter-subject registration.

In our review of the literature, we have found three main ways of guaranteeing the invertibility of the geometrical deformations yielded by a non-rigid registration algorithm. They are the following:

- **Explicit constraints on parameters:** although parameterizations such as B-splines or RBFs do not yield invertible transformations in general, it is possible to guarantee the invertibility of the result by imposing explicit *constraints* on the parameters of

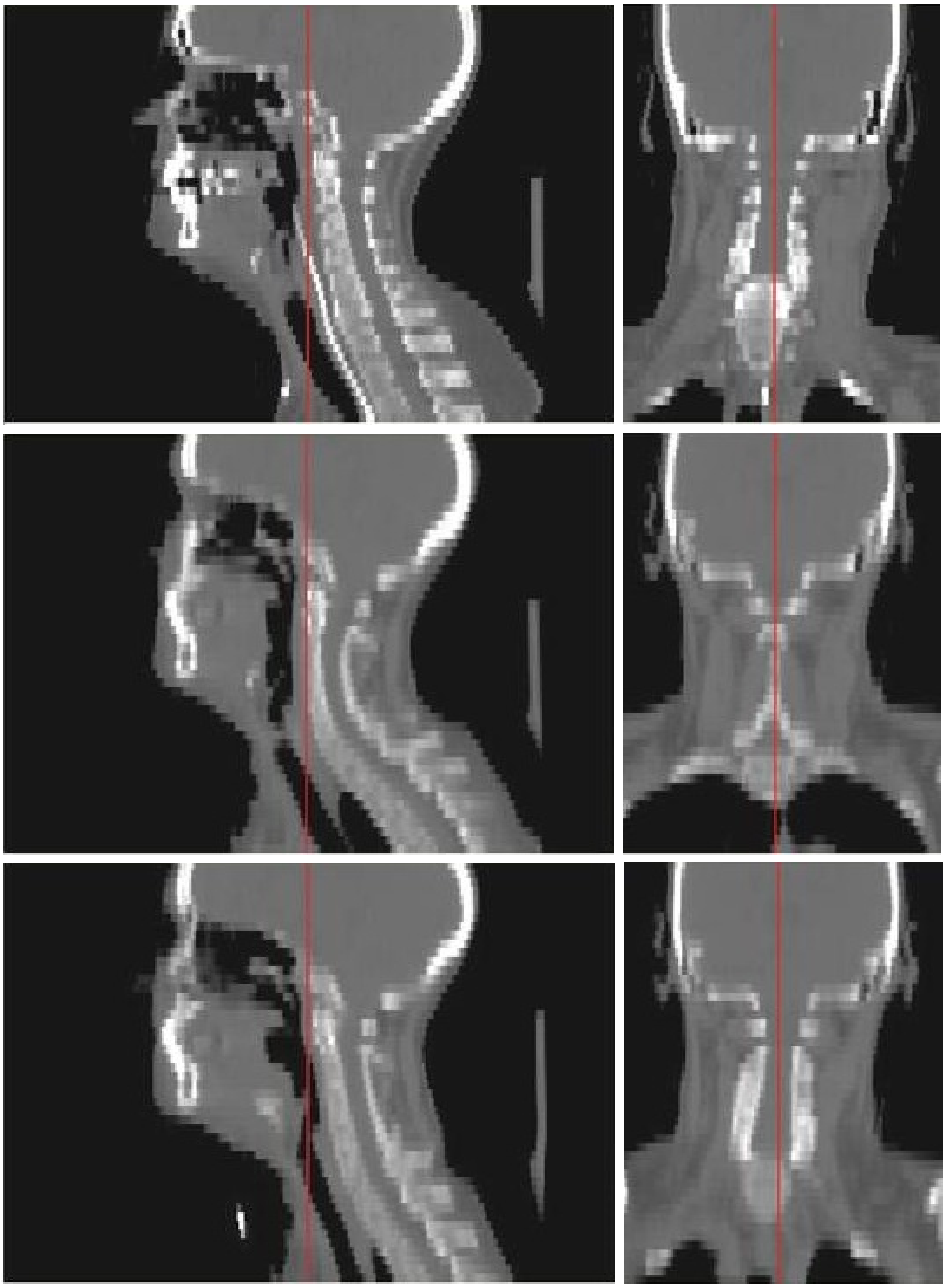

Figure 1.2: **Posture correction for CT scans of head and neck with a locally affine transformation. Top:** first subject (on the left: sagittal view, on the right: coronal view). **Middle:** second subject. **Bottom:** second anatomy after posture correction with a transformation with two affine components: one for the neck and one for the head. Note how the posture is largely corrected with only a few degrees of freedom (only 2×12); to facilitate the visual comparison between the anatomies, and in particular the position of bones, a vertical red line is drawn in each of the images at the same position. The registration was carried out with the algorithm of [Commowick 06a].

these transformations. For example, imposing values of parameters corresponding to small deformations will always result in invertible transformations. See [Noblet 05] and [Choi 00] for sufficient conditions on B-splines coefficients to guarantee invertibility, and [Rohde 03] for similar conditions imposed on RBFs coefficients. The locally rigid or affine parameterization proposed in [Narayanan 05] also yields invertible deformations, provided some upper bound on the translation part is respected.

- **Penalization of non-invertibilities:** in [Pennec 05] and [Ashburner 99], the resulting deformations are guaranteed to be invertible because non-invertible transformations are heavily 'penalized': they are given an infinite energy (i.e., intuitively, a zero probability of appearing) in an energy minimization framework. Interestingly, this strategy is *independent* of the parameterization used.

- **Intrinsic invertibility:** some types of parameterization *intrinsically* guarantee the invertibility of geometrical deformations, which means that regardless of the value chosen for the parameters, the result will always be invertible (well, at least from a theoretical point of view; of course, from a numerical point of view, this can be more difficult to ensure). With just a few DOFs, this is the case of rigid and affine transformations, and with a very large number of DOFs, this is also the case of the diffeomorphisms obtained via the integration of time-varying vector fields, as in [Trouvé 98]. Intuitively, in this setting, large invertible deformations are obtained *little by little* via the composition of small invertible deformations, an idea which was also used in the context of the 'fluid' registration algorithm of [Christensen 96]. This principle is now quite widely used to generate invertible transformations, see for example [Chefd'hotel 02], [Stefanescu 04] and [Grenander 06]. Between linear transformations and general diffeomorphisms, apart from the locally linear transformations described in this thesis, only Geodesic Interpolating Splines [Camion 01] guarantee the invertibility of resulting deformations for any choice of parameters with a small or moderate number of degrees of freedom.

We have seen that there are a number of situations in medical image registration where the use of locally rigid and affine geometrical deformations is particularly relevant, especially if these transformations are invertible. In this thesis, we have addressed this issue from the following angle: how is it possible to *fuse* local rigid or affine deformations into a global *invertible* transformation? Moreover, to what extent is it possible to devise of fusion technique which is *compatible* with the *algebraic* properties of the Lie groups of rigid and affine transformations? In Chapters 5, 6 and 7, we present several answers to these questions, along with adapted numerical methods to compute and optimize our novel transformations in practice.

1.1.3 Statistics on Invertible Linear Transformations

As we will see in this thesis, the question of the fusion of several *local* rigid or affine components into a *global* invertible transformation is closely linked to the problem of computing a mean value in the Lie groups of rigid or affine transformations. This is the reason why we have investigated this general question in this work. This is quite an important topic for statistics in Lie groups, since the mean is certainly the most fundamental statistic which can be computed, and is in many cases the first step to compute more elaborate statistics (e.g., higher order moments). In the medical imaging community, computing statistics on rigid or affine transformations are an important task, for example in order to evaluate quantitatively the performances of linear registration algorithms with respect to a *bronze standard* as in [Nicolau 03, Glatard 06], or to better characterize some pathologies as in [Boisvert 06] in the context of the analysis of the statistical properties of the human scoliosis.

In differential geometry, for a Lie group endowed with a Riemannian metric, the natural choice of mean is called the *Fréchet mean* [Pennec 06a]. But this Riemannian approach is fully compatible with the group operations (i.e., multiplication and inversion) *only* if a bi-invariant metric exists, which is for example the case for compact groups such as rotations [Pennec 06a, Moakher 02]. The bi-invariant Fréchet mean enjoys many desirable *invariance properties*, which generalize to the non-linear case the properties of the arithmetic mean: it is invariant with respect to left- and right-multiplication, as well as inversion. Unfortunately, bi-invariant Riemannian metrics do not always exist. In particular, in this work, we prove in Chapter 7 that such metrics do not exist in any dimension for *rigid transformations*, which form the simplest Lie group involved in medical image registration.

In this thesis, we have tried to overcome the lack of existence of bi-invariant Riemannian metrics for many Lie groups. To this end, we have proposed to define a *bi-invariant mean* generalizing the Fréchet mean induced by bi-invariant metrics, even in cases when such metrics *do not* exist. Our theory of bi-invariant means is presented in Chapter 7, as well as the way of computing higher order moments in this context. Alternatively, we have also proposed in this work a simpler approach to statistics of linear invertible transformations, provided by a Log-Euclidean framework for these transformations, presented in Section 6.4. Interestingly, for matrices, the bi-invariant and Log-Euclidean means are *both* generalizations of the *geometric mean* of positive numbers, since their determinants are both exactly equal to the geometric mean of the determinant of the data. The Log-Euclidean mean is much simpler to compute, but has fewer invariance properties and is limited to transformations not too far away from the identity, contrary to the bi-invariant mean.

1.1.4 Statistics on Diffeomorphisms

Currently, a large variety of non-linear registration algorithms have been proposed to deal with the non-rigid registration of medical images [Hill 01]. However, much work remains to be done to quantitatively compare these algorithms. To this end, having a consistent framework to compute statistics on general invertible transformations would be very useful. For instance it could lead to the non-linear generalization of *bronze standard* techniques such as the one presented in [Nicolau 03] in the case of rigid transformations.

Alternatively, statistics on invertible transformations could prove very valuable to inject better priors in non-linear registration algorithms. Computing statistics on *global* deformations would allow to introduce much more information than in [Commowick 05] and [Pennec 05] with *local* statistics on deformation tensors.

The computation of statistics is closely linked to the issue of the *parameterization* of diffeomorphisms. Many algorithms, as in [Chefd'hotel 02,Pennec 05], provide dense transformations which are guaranteed to be diffeomorphic, and parameterize them via their displacement field. This simple parameterization provides a simple way of performing statistics, which was used in [Charpiat 05b] to compute second-order moments, and in a similar way in [Rueckert 03], this time on B-splines coefficients. But Euclidean means of displacement fields do not necessary yield invertible deformations and such statistics are therefore not entirely satisfactory for diffeomorphisms. In [Marsland 04], it was proposed to parameterize arbitrary diffeomorphisms with the control points of Geodesic Interpolating Splines [Camion 01], and then to perform Euclidean operations on these low-dimensional parameters. However, although this guarantees the invertibility of the results, this may not be adequate for the whole variety of invertible transformations used in medical imaging.

To fully take into account the group structure of diffeomorphisms, it has been proposed to parameterize dense deformations with Hilbert spaces of *time-varying* velocity vector fields, which yield geometrical deformations via the integration of an Ordinary Differential Equation (ODE) during one unit of time [Trouvé 98, Beg 05]. In [Vaillant 04], it is suggested that the linear space of initial momenta of the geodesics in these spaces could provide an appropriate setting for statistics on diffeomorphisms. However, this is illustrated in [Vaillant 04] only in the low-dimensional and very specific case of landmark matching. To our knowledge, this statistical framework has not been used yet in the *general case*, certainly because of the *iterative nature* of the computation of the mean in this setting, which requires very stable numerical algorithms to converge.

In Chapter 8, we present the generalization to diffeomorphisms of our Log-Euclidean framework. Such a simple framework would be potentially very useful, since it would allow to perform statistics *vectorial* way on the logarithms of transformations. Without relying on Riemannian geometry, contrary to the approach of [Vaillant 04], such a framework would yield simple processing algorithms compatible with a number of algebraic properties of diffeomorphisms, in particular inversion, since Log-Euclidean frameworks are *inversion-invariant*.

1.2 Manuscript Organization

In this thesis, our contributions are detailed in six Chapters, which essentially correspond to the journal articles and research reports we have published so far, except for the last one, which contains promising preliminary results that have been presented in October at the international conference MICCAI'2006 in Copenhagen.

In Chapter 2, before we actually begin to detail our novel results, the fundamental mathematical tools used throughout this thesis are presented. First, we will present the properties of the basic tools we use to define novel types of invertible geometrical transformations: *first order differential equations (ODEs)*. Then, we detail the the notion of *Lie group*, which is particularly useful to analyze the properties of transformations groups such as rigid or affine transformations. Finally, we briefly present the powerful framework of Riemannian geometry, which is widely used to generalize many types of classical processing to non-linear spaces.

Then, we focus on the processing of symmetric positive-definite matrices, also called by abuse of language in this thesis 'tensors'. This type of data appears in many contexts, and in the past years the need for a general processing framework for tensors has grown continuously.

In Chapter 3, we present the theoretical properties of a novel and general Riemannian processing framework for tensors, called *Log-Euclidean*. Our approach is based on a *new Lie group structure* for tensors, which can smoothly be extended into a vector space structure. Remarkably, the original *algebraic structure* we propose here for symmetric and positive-definite matrices is *compatible* with the usual algebraic properties of this set: the inverse of a tensor is its usual inverse and the matrix exponential is the group exponential of our Lie group structure. Interestingly, our novel framework does not introduce any superfluous complexity, since Log-Euclidean Riemannian computations can be converted into Euclidean ones once tensors have been transformed into their matrix logarithms, which makes classical Euclidean processing algorithms particularly simple to recycle.

In Chapter 4, we focus on the application of our Log-Euclidean framework to the processing of *diffusion tensors*. In the medical imaging community, this particular case is of great importance since diffusion tensor imaging is an emerging imaging modality whose importance has been growing considerably. In particular, most attempts to reconstruct non-invasively the connectivity of the brain are based on DTI. Theoretical aspects of the Log-Euclidean framework are presented this time shortly and from a practical point of view. The Euclidean, affine-invariant and Log-Euclidean frameworks are compared experimentally, with interpolation and regularization tasks on synthetic and clinical 3D DTI data.

After tensors, we put the emphasis on geometrical transformations in the rest of this thesis.

In Chapter 5, we introduce two novel classes of transformations, called *polyrigid* and *polyaffine*, that we originally proposed in 2003 to parameterize and locally rigid or affine *diffeomorphic* deformations with a small number of flexible degrees of freedom. They can describe compactly large rigid or affine movements, unlike most free-form deformation classes. Very flexible, this tool can be readily adapted to a large variety of situations, simply by tuning the number of rigid or affine components and the number of parameters describing their regions of influence. The whole framework is exemplified successfully with the registration of histological slices.

In Chapter 6, we propose an alternative to our original polyrigid and polyaffine transformations. The novel framework presented here is called *Log-Euclidean polyaffine*, and overcomes the limitations of our original transformations in terms of *invariance properties*. Moreover, the remarkable properties of Log-Euclidean polyaffine transformations (LEPTs) allow the fast computation of these transformations on regular grids. Essentially, these nice results are obtained thanks to a much better *compatibility* between LEPTs and the *algebraic* properties of the Lie groups of rigid or affine transformations. Interestingly, this compatibility is obtained via the implicit use of a *Log-Euclidean* framework for linear transformations, which is the generalization to rigid or affine transformations of the framework presented for tensors in Chapter 3. The Log-Euclidean framework for linear transformations, as well as its generalization to abstract Lie groups, is presented. The results obtained here on the 3D locally affine registration of MR scans of the human brain suggest that our Log-Euclidean polyaffine framework provides a general and efficient way of fusing local rigid or affine components into a global invertible transformation without introducing artifacts, independently of the way local deformations are first estimated.

The Log-Euclidean polyaffine fusion of local rigid or affine components presented in Chapter 6 is implicitly based on a Log-Euclidean *averaging* of the local linear components. This has led us to investigate the problem of computing mean values of data living in general Lie groups.

In Chapter 7, we define a novel and general notion of mean in finite-dimensional Lie groups (e.g. the rigid or affine groups) which is this time *fully compatible* with the algebraic structure of these groups, contrary to the Log-Euclidean mean which is only invariant with respect to inversion and to the adjoint representation of the group. Indeed, the *bi-invariant mean* we propose generalizes to any Lie group the invariance properties of the arithmetic mean. Interestingly, we do not rely on Riemannian geometry but on the algebraic properties of Lie groups to define this mean. In fact, going beyond Riemannian metrics was unavoidable: we prove that there is no bi-invariant Riemannian metric for rigid transformations, in any dimension. Also, we briefly present the way to compute higher order moments based on this novel type of mean. Finally, we use the bi-invariant mean to define a last class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components *arbitrarily far away from the identity.*

In Chapter 8, we briefly present the generalization to diffeomorphisms of our Log-Euclidean framework. This framework allows to perform statistics on diffeomorphisms in a simple *vectorial* way via the logarithms of these transformations, with excellent theoretical properties such as inversion-invariance.

In Chapter 9, we summarize and discuss our contributions and detail our publications. And last but not least, we put into light the various perspectives of our research work.

1.3 Global Picture

In this work, we present several approaches for the processing of multiple types of data, in different contexts. This complexity makes it difficult to have a clear view of the structure of this thesis. However, we have found that most of our contributions can be presented in a simple way within a two-column table. These columns stand for the two main families of processing frameworks we have proposed: the Log-Euclidean and the bi-invariant one. Each line of this table stands for a specific type of data. Fig. 1.3 presents this table, which puts into light the various interconnections between the Chapters of this thesis.

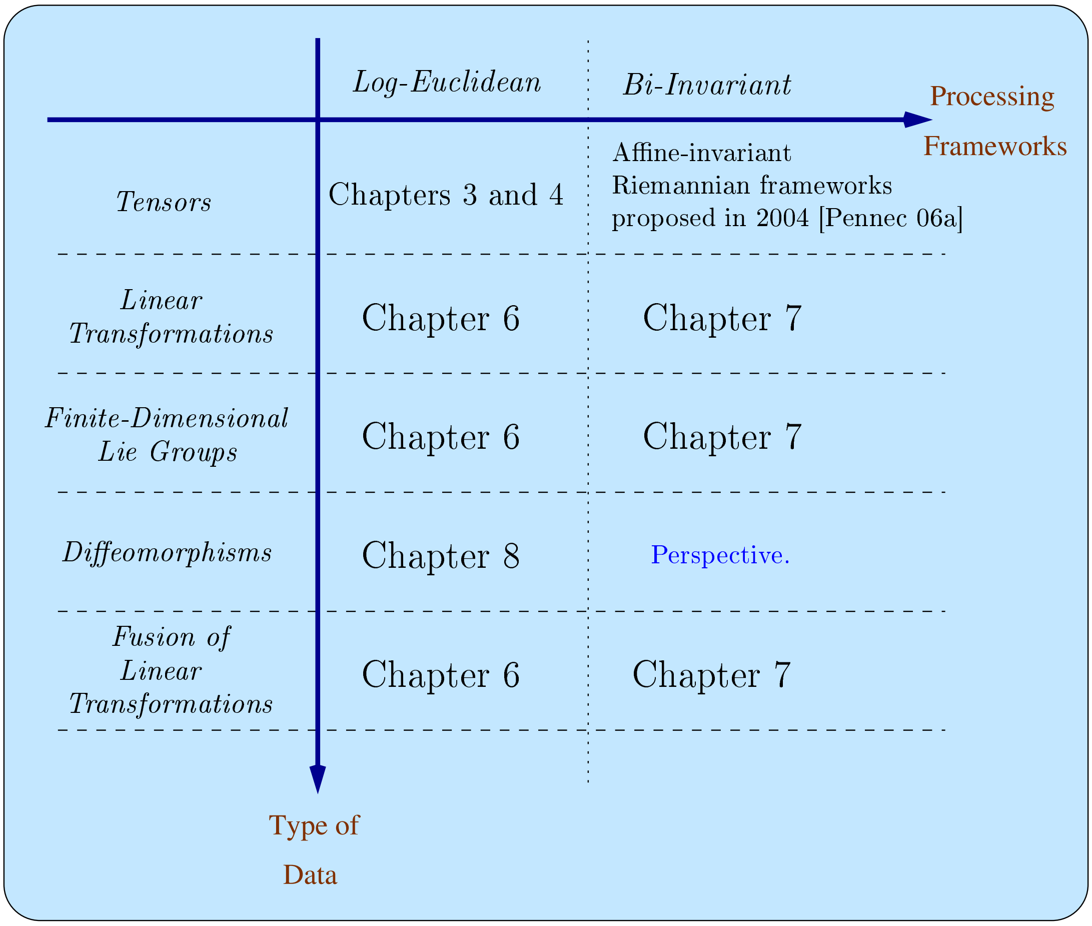

Figure 1.3: **Global view of this thesis**. **Lines: various types of data**, and **columns: processing frameworks** (either the Log-Euclidean one or the bi-invariant one). Chapter 5 does not appear in this table, because the polyaffine fusion of linear transformations presented in this chapter belongs neither to the Log-Euclidean nor the bi-invariant framework.

Chapter 2

Fundamental Mathematical Tools

In this chapter, we introduce the mathematical tools used in this thesis. First order differential equations are the basic tool we use to construct invertible geometrical transformations. The theory of Lie groups is particularly useful to analyze the properties of transformations groups such as rigid or affine transformations. And Riemannian geometry is a very powerful framework used more and more to process data belonging to non-linear spaces.

Although some of the notions and mathematical properties presented here are quite advanced, we have tried to make our presentation as *intuitive* as possible. Rather than technical details, we hope that the usefulness and vision behind each tool will be the most salient items in this short presentation.

Contents

2.1 Notations 16
2.2 First Order Ordinary Differential Equations 17
2.2.1 Definition 17
2.2.2 Life Span of a Solution to an ODE 17
2.2.3 Flow of a First Order ODE 18
2.2.4 One-Parameter Subgroups 18
2.3 Lie Groups 19
2.3.1 Definition of Lie groups 19
2.3.2 Examples 19
2.3.3 Lie Algebra and Adjoint Representation 20
2.3.4 Matrix Exponential and Logarithm 21
2.3.5 Lie Group Exponential and Logarithm 21
2.3.6 Baker-Campbell-Hausdorff Formula 23
2.4 Riemannian Geometry 23
2.4.1 Riemannian Metrics 25
2.4.2 Riemannian Geodesics 25
2.4.3 Riemannian Exponential and Logarithm 26
2.4.4 Fréchet Mean 26

2.1 Notations

Notations. In the sequel of this work, we will use a number of notations, which are listed below. We begin with notations for usual matrix groups and submanifolds:

- $GL(n)$ is the group of real invertible $n \times n$ matrices, and more generally, for any (finite dimensional) vector space E, $GL(E)$ will be the group of invertible linear operations acting on E.
- $SL(n)$ is the *special linear group*, i.e. the subgroup of matrices of $GL(n)$ whose determinant is equal to 1.
- $O(n)$ is the group of *orthogonal* transformations, i.e. square matrices satisfying $R.R^T = Id$, where Id is the identity matrix and R^T is the transposed matrix of R.
- $SO(n)$ is the *special orthogonal group*, better known has the group of *rotations*. It is the subgroup of $O(n)$ whose elements satisfy $det(R) = 1$.
- $SE(n)$ is the group of *special Euclidean* transformations, i.e. the group of rigid displacements.
- $M(n)$ is the space of real $n \times n$ square matrices.
- $Sym_\star^+(n)$ is the space of symmetric positive-definite real $n \times n$ matrices.
- $Sym(n)$ is the vector space of real $n \times n$ symmetric matrices.
- $\mathrm{Diag}(\lambda_1, ..., \lambda_n)$ will be the diagonal matrix constructed with the real values $(\lambda_i)_{i \in 1...n}$ in its diagonal.
- For any square matrix M, $Sp(M)$ is the *spectrum* of M, i.e. the set of its eigenvalues.

General (abstract) Lie group and differential calculus notations:

- When $\mathcal{G}$ is a Lie group, its neutral element will be written e, and a typical element of $\mathcal{G}$ will be m. The Lie algebra of $\mathcal{G}$ will be written $\mathfrak{g}$.
- The tangent space of $\mathcal{G}$ at point m will be referred to as $T_m\mathcal{G}$, which can be intuitively thought of as the linear space 'best approximating' $\mathcal{G}$ around m. For general differentiable manifolds, the same notation will be used (i.e. we will write T_mE for the tangent space of a manifold E at a given point m).
- We denote L_m (resp. R_m) the left- (resp. right-)multiplication by an element $m \in \mathcal{G}$. Furthermore, we will let $Inv : \mathcal{G} \to \mathcal{G}$ be the inversion operator.
- Let $\phi : E \to F$ be a differentiable mapping between two smooth manifolds (which are not necessarily Lie groups). Its differential (or *tangent map*) at a point $m \in E$ will be written $D_m\Phi$, which is a linear mapping from T_mE to $T_{\Phi(m)}F$. This means that to a tangent vector located at m (which is basically an infinitesimal displacement) it associates a tangent vector at $\Phi(m)$ (another infinitesimal displacement, 'living' in a different vector space).

  In the sequel of this thesis, we will let the action of $D_m\Phi$ on an infinitesimal displacement v in T_mE be written $D_m\Phi.v$.

2.2 First Order Ordinary Differential Equations

A classical way of obtaining *invertible* smooth geometrical transformations is to use first order ordinary differential equations (ODEs) [Tenenbaum 85]. The fundamental idea is that composing iteratively small invertible deformations guarantees that the resulting (possibly large) deformations will remain invertible. This was for example noted in the work of Christensen *et al.* on fluid deformations, who used the expression 'topology-preserving' rather than 'invertible'. Indeed, on page 1439 of [Christensen 96], one can read:

"We note that the above procedure assures that the concatenated transformation of the template into the study preserves the topology of the template. This is because each of the propagated template transformations preserve topology (due to the fact that the Jacobian of the transformation is positive globally) and the concatenation of topology-preserving transformations produces a topology-preserving transformation."

2.2.1 Definition

To obtain invertible smooth geometrical deformations with first order ODEs, any point x of $\mathbb{R}^n$ can be displaced little by little (in fact, infinitesimally) and in a reversible manner via the continuous integration during one unit of time of an evolution equation. A first order ODE is an equation of the following form:

$$\dot{x}(s) = V(x, s). \tag{2.1}$$

Of course, the integration of (2.1) is a well-defined mathematical operation only if the *velocity vector field* $V(x, s)$ is smooth enough (for instance $\mathcal{C}^1$) with respect to x. The principle of obtaining geometrical transformations via the continuous addition of infinitesimally small deformations during one unit of time is illustrated in Fig. 2.1.

A particular case of interest is when $V(x, s)$ *does not depend* on s. Then, the ODE is called *stationary* or equivalently *autonomous*. For instance, the ODE defining Log-Euclidean polyaffine transformations in Chapter 6 is stationary.

In the work of Trouvé, Younes, Miller, Joshi and others [Trouvé 98, Trouvé 00, Camion 01, Beg 03, Miller 03, Guo 04, Guo 04, Joshi 00, Beg 05, Trouvé 05b, Glaunes 04, Allassonnière 05], ODEs are used to generate very general *diffeomorphisms*[1], with a *very high number* of degrees of freedom (theoretically infinite).

On the contrary, we mainly use ODEs in this work to define novel classes of diffeomorphisms with a *small number* of flexible degrees of freedom. These transformations, called *polyrigid* and *polyaffine*, are described in Chapters 5 (original polyrigid transformations), 6 (Log-Euclidean polyaffine transformations) and 7 (left-invariant polyaffine transformations). We rely in Chapter 8 on autonomous first order ODEs to generalize the notion of logarithm to diffeomorphisms.

2.2.2 Life Span of a Solution to an ODE

In order to define a geometrical transformation via the ODE (2.1), it is necessary to prove first that the position at time 1 exists, whatever the initial position may be. This is what we do in the next chapters for polyrigid and polyaffine transformations. This is a necessary precaution since for an arbitrary ODE, the existence is not always insured, however smooth the velocity function V may be. Consider for instance, the 1D evolution

$$\dot{y}(s) = V(y) = y^2.$$

[1] i.e. smooth invertible transformations, whose inverse is also smooth

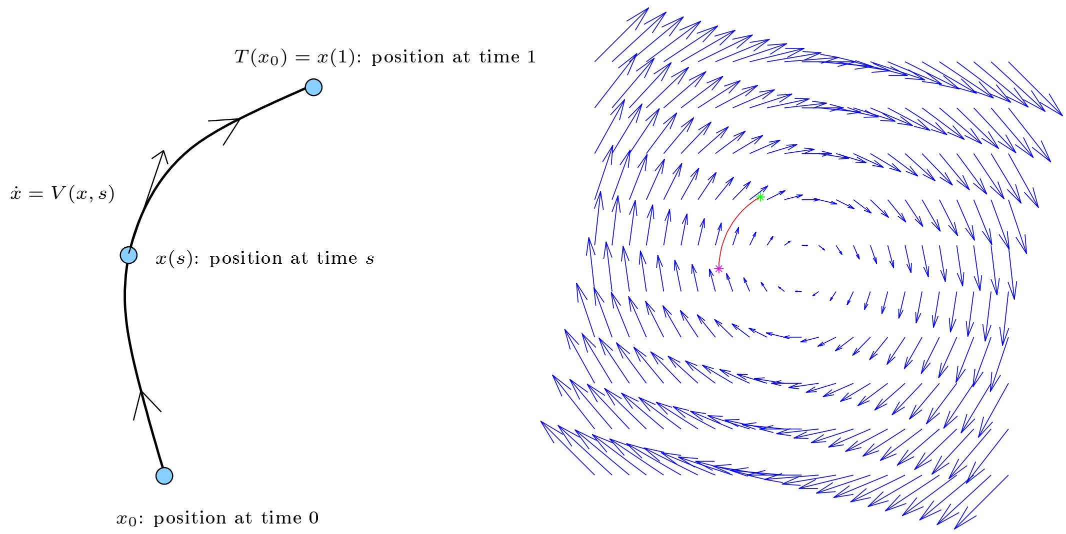

Figure 2.1: **Integration of velocity vector fields. Left:** integration of a vector field between time 0 and time 1. The value at a point x_0 of the global transformation is given by $x(1)$. **Right:** example of a single rotation. Velocity vectors are displayed in blue. The magenta point corresponds to the initial condition and the green point is the position reached at time 3 (not time 1 so that the trajectory be longer and thus more visible).

Its solution with an initial position y_0 is $y(s) = \frac{1}{1/y_0 - s}$. Thus, we see that for $1/y_0 > 0$, the life-span of the solution only extends between $-\infty$ and $\frac{1}{y_0}$, and if $\frac{1}{y_0} < 1$, then the position at time 1 is absolutely undefined, the particle having gone to infinity before that.

2.2.3 Flow of a First Order ODE

Definition 2.1. *The flow associated to a first order ODE is the family of mappings $\Phi(., s) : \mathbb{R}^n \to \mathbb{R}^n$ parameterized by a time parameter $s \in \mathbb{R}$, such that for a fixed x_0, $s \mapsto \Phi(x_0, s)$ is the unique solution of $\dot{x} = V(x)$ with initial condition x_0 at time 0.*

Intuitively, for a fixed s, the mapping $x \mapsto \Phi(x, s)$ gives the way the ambient space is deformed by the integration of the ODE during s units of time. It is always a *diffeomorphism*, i.e. a differentiable one-to-one mapping between the ambient space and itself, whose inverse is also differentiable. This invertibility property simply comes from the fact that all deformations induced by the ODE are reversible since one can go back in time by simply multiplying the velocity vector $V(x)$ by -1! The smoothness of the flow comes from the smoothness of $V(x)$.

2.2.4 One-Parameter Subgroups

Definition 2.2. *Let $(G, .)$ be a group (i.e., the multiplication '.' is associative and there exists a neutral element e and each element of G has a unique inverse). Then a family of elements $g(s)$ of G parameterized by $s \in \mathbb{R}$ is called a one-parameter subgroup of G if and only if:*

$$(a)\ \ g(0) = e, \ \ \textit{i.e. the neutral element of } G$$
$$(b)\ \ \textit{for all } s, t \textit{ in } \mathbb{R}: \ \ g(s).g(t) = g(s + t).$$

Furthermore, if one knows how to differentiate functions valued in G (e.g., G is a Lie group), and if $g(s)$ is differentiable at 0, then $\frac{dg}{dt}(0)$ is called the infinitesimal generator of the subgroup.

One-parameter subgroups are particularly useful to describe a crucial property of the flow of stationary first order ODEs. Furthermore, they are closely linked to the notion of *group exponential*, described in the next Section. These remarkable subgroups will also be at the heart of our definition of bi-invariant means in Lie groups in Chapter 7.

Interestingly, the infinitesimal generator of a one-parameter subgroup contains all the information about the subgroup, and can generate it entirely, which explains its name. See for example [Godement 82] for more details on this topic in the case of Lie groups.

We have the following result: the flow $\Phi(., s)$ of an *autonomous* first-order ODE is a *one-parameter subgroup of the group of diffeomorphisms*. In other words: $\Phi(., s) \circ \Phi(., t) = \Phi(., s + t)$. This implies in particular that the deformations of space given at time 1 by $\Phi(., 1)$ are twice that observed at time 0.5 via $\Phi(., 0.5)$. The infinitesimal generator of the flow is simply $V(x)$. This is not surprising since it is quite clear how $V(x)$ infinitesimally generates the flow: this is done precisely by integrating the associated ODE. These results will be used in Chapter 6 to prove the particularly nice properties of Log-Euclidean polyaffine transformations and in Chapter 8 to generalize the notion of logarithm to diffeomorphisms.

2.3 Lie Groups

2.3.1 Definition of Lie groups

We start by recalling the basic properties of Lie groups, along with the convenient notions which are classically used to describe these properties. Typical examples of such groups are groups of geometrical transformations (e.g., rigid or affine transformations), where the multiplication is the composition of mappings. Such groups are particularly important in medical imaging, since the two most basic types of transformations used to register medical images, i.e. rigid and affine transformations, are both Lie groups.

In simple terms, a Lie group is first a group in the *algebraic* sense, i.e. a set of elements in which a *multiplication* between elements is defined. This multiplication is assumed to have neat and intuitive properties: it is associative (i.e., $(a.b).c = a.(b.c)$), has a neutral element e and each element a has a unique inverse a^{-1}.

Second, a Lie group has a structure of (smooth, i.e. $\mathcal{C}^\infty$) *differential manifold*. This means that it is locally similar to a vector space, but can be quite 'curved' globally.

Third, the algebraic and differential structures are *compatible*: inversion and (left- and right-) multiplications are *smooth* mappings. This means that it is possible to indefinitely differentiate them. For more formal definitions and more details, please refer to classical differential geometry books like [Sternberg 64] or [Gallot 93].

2.3.2 Examples

Many usual spaces can be viewed as Lie groups. Namely:

- *Vector spaces* (with their addition)
- *Multiplicative matrix groups*: $GL(n)$, $O(n)$, $SO(n)$, etc., with the usual matrix multiplication.
- *Geometric transformation groups* such as rigid transformations, similarities, affine transformations... which can anyway also be looked upon as matrix groups via their 'faithful' representation based on homogeneous coordinates.

- Infinite-dimensional *Lie groups of diffeomorphisms* have also been recently gaining much importance in computational anatomy [Trouvé 98].

2.3.3 Lie Algebra and Adjoint Representation

We will need a number of notions classically used to describe the properties of Lie groups. They are the following:

- To vector v tangent to $\mathcal{G}$ at the identity can be associated in a one-to-one manner a *left-invariant* vector field defined by $X_v(m) = D_e L_m.v$, i.e. simply by left-multiplying v.

- $(T_e\mathcal{G}, +, .)$ is by construction a *vector space*, and can also be given a structure of *Lie algebra*, i.e. one can give it an *extra* algebraic operation which is a bi-linear inner product called the *Lie bracket*, denoted here $[\,,]$. This operation closely reflects the multiplicative properties of the group $\mathcal{G}$. In particular, for commutative Lie groups, the Lie bracket is always null.

  We can identify $T_e\mathcal{G}$ with the set of left-invariant vector fields, and since this set is a *Lie subalgebra* of vector fields on $\mathcal{G}$, the inner product $[\,,]$ on $T_e\mathcal{G}$ can be actually defined from the general Lie bracket on smooth vector fields.

  As mentioned previously, the notation for the Lie algebra of $\mathcal{G}$ will be in this thesis $\mathfrak{g}$. It has a number of remarkable algebraic properties (in addition to its associativity and bi-linearity) which are the following:

  i) $\forall a, b \in \mathfrak{g},\ [a,b] = -[b,a]$ ('anti-commutativity'), which implies $[a,a] = 0$

  ii) $\forall a, b, c \in \mathfrak{g},\ [a,[b,c]] + [c,[a,b]] + [b,[c,a]] = 0$ (Jacobi identity).

  Simple examples of Lie brackets are given by $GL(n)$ and its multiplicative subgroups, like $SL(n)$ or $SO(n)$. In these cases, the Lie algebra is a *vector space of square matrices*, and the Lie bracket between two elements M and N of this algebra is the *commutator* of these two matrices, i.e. $[M,N] = M.N - N.M$. In particular, the Lie algebra of $GL(n)$ is $M(n)$, that of $SL(n)$ is the vector subspace of $M(n)$ of matrices with a trace equal to zero, and the Lie algebra of $SO(n)$ is the vector space of skew symmetric matrices. For a complete account on Lie Algebras, see [Bourbaki 89].

- $\mathcal{G}$ can be 'represented' by a group of matrices acting on $\mathfrak{g}$, via what is called its *adjoint representation*, $Ad(\mathcal{G})$. The properties of this representation and the existence of bi-invariant metrics for the group $\mathcal{G}$ are highly linked.

  The expression 'representation' means that one can map each element of the group into a *linear operator* (i.e., a matrix) which acts on the Lie algebra. More precisely, an element m of $\mathcal{G}$ acts on an element v of $\mathfrak{g}$ by $Ad(m).v =$ '$m.v.m^{-1}$' [2]. This operation is called a representation in the sense of *representation theory* (see [Lang 04] for a complete treatment), which means that this mapping is *compatible* with the Lie group structure of $\mathcal{G}$. This compatibility consists of the following properties:

[2] To be completely rigorous, one has to resort to the (more complicated) differentials of left- and right-multiplication. This yields: $Ad(m).v =$ ' $m.v.m^{-1}$ ' $= D_{m^{-1}}L_m.D_eR_{m^{-1}}.v = D_mR_{m^{-1}}.D_eL_m.v$ by associativity of the group multiplication. In the matrix case, we have the (simple this time) formula: $Ad(R).M = R.M.R^{-1}$, which only uses two matrix multiplications and one matrix inversion.

i) $Ad(e) = Id$

ii) $\forall m \in \mathcal{G},\ Ad(m^{-1}) = Ad(m)^{-1}$

iii) $\forall m, n \in \mathcal{G},\ Ad(m.n) = Ad(m).Ad(n)$

iv) $Ad : \mathcal{G} \to GL(\mathfrak{g})$ is smooth.

This amounts to saying that Ad is a *smooth group homomorphism* (or Lie group homomorphism).

2.3.4 Matrix Exponential and Logarithm

The Matrix Exponential and Logarithm. Before we present the general group exponential and logarithm, let us recall the fundamental properties of the *matrix* exponential and logarithm, which correspond to the group exponential and logarithm of the Lie group of $n \times n$ invertible matrices, $GL(n)$. They are the generalization to matrices of the well-known scalar exponential and logarithm.

Definition 2.3. *The exponential* $\exp(M)$ *of a matrix* M *is given by* $\exp(M) = \sum_{n=0}^{\infty} \frac{M^k}{k!}$*. Let* $G \in GL(n)$*. If there exists* $M \in M(n)$ *such that* $G = \exp(M)$*, then* M *is said to be a logarithm of* N*.*

In general, the logarithm of a real invertible matrix may not exist, and if it exists it may not be unique. The lack of existence is a general phenomenon in connected Lie groups. One generally needs *two* exponentials to reach every element [Wüstner 03]. The lack of uniqueness is essentially due to the influence of rotations: rotating of an angle α is the same as rotating of an angle $\alpha + 2k\pi$ where k is an integer. Since the logarithm of a rotation matrix directly depends on its rotation angles (one angle suffices in 3D, but several angles are necessary when $n > 3$), it is not unique.

Principal Logarithm of a Matrix. When a real invertible matrix has no (complex) eigenvalue on the (closed) half line of negative real numbers, then it has a unique real logarithm whose (complex) eigenvalues have an imaginary part in $]-\pi, \pi[$ [Cheng 01]. In this case this particular logarithm is well-defined and called *principal*. We will write $\log(M)$ for the principal logarithm of a matrix M whenever it is defined.

Numerical Computation. Thanks to their remarkable *algebraic* properties, and essentially their link with one-parameter subgroups (see next Section for more details on this topic), matrix exponential and logarithms can be quite efficiently numerically computed.

In practice, we have used in this work the popular 'Scaling and Squaring Method' [Higham 05] to compute numerically matrix exponentials, as well as the 'Inverse Scaling and Squaring Method' [Cheng 01] to compute matrix logarithms. More details on these methods can be found in Chapter 6, Sections 6.3 and 6.4.5. Interestingly, we generalize these methods in Chapter 8 to compute this time the exponential of a vector field and the logarithm of a diffeomorphisms.

2.3.5 Lie Group Exponential and Logarithm

Let us now define the general group exponential and logarithm in Lie groups. For more details on these properties, see [Godement 82]. Basically, these properties are very similar to those of the matrix exponential and logarithm, which are a particular case of such mappings. One should note that this particular case is actually quite general, since most classical Lie groups can be looked upon as matrix Lie groups anyway [Hall 03]. But all Lie groups are *not* (at least directly) multiplicative matrix groups, as in the case of the Lie group structure we have recently proposed by symmetric positive-definite matrices, which is described in Chapter 3. This is the reason why we do not limit ourselves to the matrix case.

Definition 2.4. *Let $\mathcal{G}$ be a Lie group and let v be an tangent vector at the identity, i.e. an element of the Lie Algebra $\mathfrak{g}$. The group exponential of v, denoted* $\exp(v)$*, is given by the value at time* 1 *of the unique function $g(t)$ defined by the following ordinary differential equation (ODE):*

$$\begin{cases} \frac{dg}{dt} = D_e L_{g(t)}.v \\ g(0) = e. \end{cases} \tag{2.2}$$

Eq. (2.2) has particularly nice properties. $g(t)$ is in fact defined for all t, and yields a continuous *one-parameter subgroup* (also called one-parameter *Lie* subgroup), which means that $g(0) = e$, $g(t+t') = g(t).g(t') = g(t').g(t)$. The velocity vector v is called the *infinitesimal generator* of this subgroup. See [Gallot 93], pages 27-29 for proofs of these properties. In fact, Eq. (2.2) is the equivalent of the matrix differential equation, which is a nice and classical *linear ODE*:

$$\begin{cases} \frac{dG}{dt} = G.V \\ G(0) = Id, \end{cases}$$

whose solution is well-known from classical ODE theory to be $G(t) = \exp(tV)$, where exp is the matrix exponential [Tenenbaum 85].

One-Parameter Subgroups vs. Group Exponential. We have just seen that for all v belonging to $\mathfrak{g}$, $\exp(t.v)$ is a one-parameter subgroup of $\mathcal{G}$: the additive subgroup $(t.V)_t$ of $\mathfrak{g}$ is mapped into a multiplicative subgroup of $\mathcal{G}$ by the exponential. Conversely, we have the interesting result that all continuous one-parameter subgroups of $\mathcal{G}$ are of this form ([Sternberg 64], Section V, Theorem 3.1, page 223). *This provides a simple way of computing the group exponential in situations where one-parameter subgroups are easy to obtain.*

The Exponential as a Local Diffeomorphism. Very much like the exponential map associated to a Riemannian metric, the group exponential is diffeomorphic locally around 0. More precisely, we have the following theorem:

Theorem 2.1. *The group exponential is a diffeomorphism from a open neighborhood of* 0 *in* $\mathfrak{g}$ *to a open neighborhood of e in $\mathcal{G}$, and its differential map at* 0 *is the identity.*

Proof. Since the exponential is a smooth mapping, the fact that its differential map is invertible at e allows for the use of the 'Implicit Function Theorem', which guarantees that it is a diffeomorphism from some open neighborhood of 0 to a open neighborhood of $\exp(0) = e$. For more details, see [Gallot 93], page 28. □

This theorem implies that one can define *without ambiguity* a logarithm in a open neighborhood of e: for every g in this open neighborhood, there exists a unique v in the open neighborhood of 0 in $\mathfrak{g}$, such that $g = \exp(v)$. In the following, we will write $v = \log(g)$ for this logarithm, which is the (abstract) equivalent of the (matrix) *principal* logarithm.

2.3.6 Baker-Campbell-Hausdorff Formula

Before presenting briefly the principles of Riemannian geometry, let us focus on a last fundamental property of the group exponential and logarithm: the Baker-Campbell-Hausdorff formula (or BCH formula). Intuitively, this formula shows how much $\log(\exp(v).\exp(w))$ deviates from $v + w$ due to the (possible) non-commutativity of the multiplication in $\mathcal{G}$. Remarkably, this deviation can be expressed only in terms of Lie brackets between v and w [Godement 82].

Theorem 2.2. *Series form of the BCH formula ([Godement 82], Chapter VI). Let v, w be in $\mathfrak{g}$. If they are small enough, then the logarithm of the product* $\exp(v).\exp(w)$ *is always well-defined and we have the following development:*

$$\begin{aligned} \log(\exp(v).\exp(w)) \quad = \quad & v + w + 1/2([v,w]) \\ & +1/12([v,[v,w]] + [w,[w,v]]) \\ & +1/24([[v,[v,w]],w]) + O((\|v\| + \|w\|)^5). \end{aligned} \tag{2.3}$$

Following [Godement 82], let us write $H : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ the mapping defined near 0 such that $H(v,w) = \log(\exp(v).\exp(w))$. A fundamental property of this function is the following: it is not only $\mathcal{C}^\infty$ but also *analytic* around 0, which means that $H(v,w)$ (near 0) is the sum of an absolutely converging multivariate infinite series (the usual multiplication is replaced here by the Lie bracket). This implies in particular that all the (partial) derivatives of this function are also analytic.

We use twice the BCH formula in the sequel of this thesis, namely in Chapter 7 to prove the existence and uniqueness of the bi-invariant mean in any Lie group, and in Chapter 3 to compare the traces of the Log-Euclidean and affine-invariant means of symmetric positive-definite matrices.

2.4 Riemannian Geometry

The geometry of non-linear spaces based on Riemannian metrics is extremely powerful, and allows the generalization to these spaces of many type of data processing techniques such as statistics [Pennec 96, Pennec 06a] or non-linear filtering [Pennec 06b]. Intuitively, the idea behind this geometry is to endow a smooth manifold with a distance between points *compatible* with its smooth differentiable structure. With this type of distance, one can compute shortest paths between points (i.e. geodesics), generalize the notion of arithmetic mean (with the so-called Fréchet mean, see Chapter 7, Section 7.2 for details), generalize usual differential operators such as the Laplacian (for a presentation of the Laplace-Beltrami operator, see for example [Gallot 93], Chapter IV), give a precise definition to the intuitive notion of curvature, etc.

For the processing of medical images and of related geometrical transformations or data, Riemannian geometry has been increasingly used in the last decade, and particularly in the past few years. One can find below a non-exhaustive list of situations where the use of Riemannian geometry has been particularly fruitful in our domain and in related research areas:

- **Statistics on rotations**: bi-invariant Riemannian metrics are particularly useful to compute statistics on rotations [Pennec 96, Pennec 98a, Moakher 02].

- **Statistics on rigid transformations**: left-invariant Riemannian metrics are quite straightforward to use in order to compute in 2D and 3D [Pennec 98b], either in the context of 2D images mosaicing [Vercauteren 06] or the statistical study of human spine scoliosis [Boisvert 06].

- **Processing of symmetric positive-definite matrices (so-called 'tensors')**: in 2004, several teams have proposed to use affine-invariant Riemannian metrics to process diffusion-tensor images (DT-MRI) [Moakher 05, Fletcher 04a, Batchelor 05, Pennec 06b, Lenglet 06a], which allow for the estimation, regularization, interpolation or segmentation of this type of images with particularly nice properties. In Chapters 3 and 4, we present an alternative family of Riemannian metrics, called Log-Euclidean, which are particularly well-adapted to the processing of diffusion tensors and of symmetric positive-definite matrices in general.

- **Metrics on parametric probability density functions**: in [Lenglet 06a], affine-invariant Riemannian metrics on tensors are defined via the use of a Fisher-Rao Riemannian metric on the space of Gaussian probability functions of mean zero. Intuitively, this choice of metric is based on the idea that the larger the intrinsic local uncertainty on the estimation of parameters is, the 'closer' parameters should be locally. This approach can be used in more general situations than simple Gaussians. Indeed, in [Peter 06], a Fisher-Rao Riemannian metric is used on a subspace of mixtures of 2D Gaussians. This allows to define a metric which is invariant under re-parameterization on N-tuples of points modeling the shapes of anatomical structures.

- **Metrics on shape spaces of N-tuples of points**: much effort has been devoted to analyze mathematically the properties of *shape spaces*, i.e. of the information contained in sets of points which is invariant with respect to any rigid transformation and scaling. D.G. Kendall was the founder of the theory of shape spaces of N-tuples of points [Kendall 84], and his framework is particularly useful for computing statistics on shapes [Kendall 89, Le 93]. It has been widely used in the Computer Vision and medical image processing communities. See for example [Davies 02] for an advanced information theory-based optimization strategy to find 'good' correspondences between continuous shape instances, in order to apply Kendall's theory to discretized versions of continuous shapes. Shape theory can be used in practice to study many types of shapes; in particular, the application of shape spaces theory to the statistical analysis of the shape of the human ear canal is presented in [Paulsen 04]. For a very pleasant introduction to shape spaces, see [Small 96].

- **Metrics on shape spaces of continuous objects**: theoretically, shape spaces of continuous objects are far more complex to deal with than N-tuples of points, except in a few exceptional cases such as in the following. For instance, a large variety of Riemannian frameworks have been proposed to deal with *2D closed curves* [Klassen 03, Joshi 05, Michor 06]. Very recently, an extension of some of these approaches to 3D closed curves has been presented in [Klassen 06]. An alternative framework to deal with 2D closed curves, although outside the framework of Riemannian geometry, can be found in [Charpiat 05a], where differentiable approximations of the Hausdorff distance on compact sets are used to compute first and second-order moments of 2D closed curves of arbitrary topology in a level set framework. A Riemannian framework has also been proposed for a particular finite-dimensional parameterization of 3D shapes called 'M-reps' (for 'medial-representations') [Fletcher 04b]. For advanced statistical tests on M-reps taking into account the specific direct product structure of this non-linear space, see [Terriberry 05].

- **Metrics on diffeomorphisms**: this time in a infinite-dimensional setting, right-invariant or left-invariant metrics on subgroups of diffeomorphisms have been increasingly used in computational anatomy, thanks to the work of Trouvé, Younes, Miller and others. See for example [Miller 06, Holm 04b, Holm 04a] for recent accounts on the remarkable properties of geodesics in this infinite-dimensional setting. Also, see [Vaillant 04] for statistics on low-dimensional diffeomorphisms in this framework. One should note that these metrics defined on diffeomorphisms are not strictly speaking Riemannian metrics, since no classical differential structure (only a "weak" one) has been identified yet for the groups of diffeomorphisms considered in this type of approaches, and the collection of scalar products in the spaces "playing the role of tangent spaces" is therefore not "smooth" in any fully rigorous sense [Trouvé 95]. However, we are very close to Riemannian geometry and many of the usual properties of Riemannian geometry apply in this infinite-dimensional setting.

- **Metrics on images and shapes via semi-direct product spaces with diffeomorphisms**: more recently, it has been proposed to induce metrics on images and continuous shapes from the existing metric structures on diffeomorphisms. See [Trouvé 05b, Trouvé 05a] for rigorous presentations of the theoretical aspects of this framework, and [Xu 06, Lorenzen 05, Davis 04] for practical applications, such as computing mean images of the human brain. In this context, a trade-off between geometrical deformations and differences in shape or image intensities has to be chosen.

For a general introduction to Riemannian geometry, see for instance [Gallot 93] or [do Carmo 92].

2.4.1 Riemannian Metrics

Riemannian metrics (or distances) are *compatible* with the differentiable structure of differentiable manifolds. In Riemannian geometry, the idea is to define *smoothly* in each tangent space T_mE a *scalar product* $< .,. >_{T_mE}$. The distance between two points m and n is then obtained by computing the minimal length of a smooth curve $c(t)$ joining them in one unit of time. We recall that the length $l(c)$ of c is classically given by:

$$l(c) = \int_0^1 \left\| \frac{dc}{dt}(t) \right\|_{T_{c(t)}E} dt.$$

Thanks to the smoothness of the scalar product, Riemannian distances are indefinitely differentiable.

2.4.2 Riemannian Geodesics

A smooth curve of minimal Riemannian length between two points is called a *geodesic*, and if there exists a unique geodesic between two points, this curve is called a *minimizing geodesic*. Interestingly, for any given point in a smooth manifold endowed with a Riemannian metric, there exists an *open neighborhood* of this point which is *weakly geodesically convex*, i.e. where any couple of points can be joined by a minimizing geodesic. Interestingly, geodesics satisfy a second-order differential equation, and thus can be practically computed via the numerical integration of an ODE (see [Gallot 93], page 80, for more details on this ODE, which can be neatly expressed in terms of *first order Christoffel symbols*).

When two points are 'far' apart, there can fail to be any geodesic between them (think of a set with several connected components) or on the contrary several geodesics (possibly an infinity) can join these points (think of antipodal points on a sphere). For details on the existence and possible uniqueness of geodesics, see for example [Gallot 93].

2.4.3 Riemannian Exponential and Logarithm

In Section 2.3.5, we recalled the general notions of exponential and logarithm in Lie groups. Remarkably, these notions can be generalized in the framework of Riemannian geometry without relying on any Lie group structure.

Definitions. At a given point m of E, the Riemannian exponential $\exp_m$ is defined as the mapping $T_mE \to E$ which associates to a velocity vector v of T_mE the position obtained at time 1 by the unique geodesic starting at time 0 at m with the initial speed v. In fact, contrary to the group exponential, the Riemannian exponential $\exp_m$ is only defined in general in an open neighborhood of 0 in T_mE, since Riemannian geodesics may go out of the manifold in finite time (e.g., when there are 'holes'), which is *not* the case of continuous one-parameter subgroups. Riemannian manifolds such that for all m, $\exp_m$ is defined on the whole tangent space T_mE are called *geodesically complete*.

Exactly as in the Lie group case, $\exp_m$ is a *local diffeomorphism*, and is diffeomorphic in an open neighborhood of 0 in T_mE, which is mapped into a open neighborhood of m in E. As a consequence, its inverse mapping, the *Riemannian logarithm* $\log_m$, is well-defined locally in an open neighborhood of m. $\log_m$ associates to a given point n near m the unique velocity vector v close to 0 in T_mE such that $\exp_m(v) = n$. Moreover, the norm of v in T_mE is equal to the length of the geodesic between m and n. In this sense, the logarithm map $\log_m$ locally 'unfolds' (near m) the manifold E into the tangent space T_mE, while conserving distances along geodesics starting from m, which are mapped by $\log_m$ in T_mE into the straight lines passing through 0.

The notions of Riemannian exponential and logarithm are exemplified on the unit sphere in Fig. 2.2.

Riemannian Exponential vs. Group Exponential. In Lie groups endowed with a Riemannian metric, the group and Riemannian exponentials are closely related in a number of situations. In particular, in the cases where bi-invariant Riemannian metrics exist, geodesics are given by the translated versions of one-parameter subgroups, and in particular, the Riemannian exponential at e is simply the group exponential (possibly rescaled). See Chapter 7, Section 7.2 for more details on this subject.

From a practical point of view, the most favorable situation is certainly when the Riemannian exponential is a function of the matrix exponential, which allows the direct use of very efficient numerical algorithms, like the ones described in Section 2.3.4. Particularly nice examples of such favorable situations are given by left-invariant metrics on rigid transformations [Pennec 96] and affine-invariant metrics on tensors [Pennec 06b] (where the space of symmetric positive-definite matrices is looked upon as an *homogeneous space*). Otherwise, the numerical integration of the second-order ODE satisfied by geodesics can be more difficult, and the computation of the Riemannian logarithm even harder.

2.4.4 Fréchet Mean

The framework of Riemannian geometry allows to generalize many usual statistical tools in non-linear spaces. For a presentation of this topic, the interested reader can refer to [Pennec 06a].

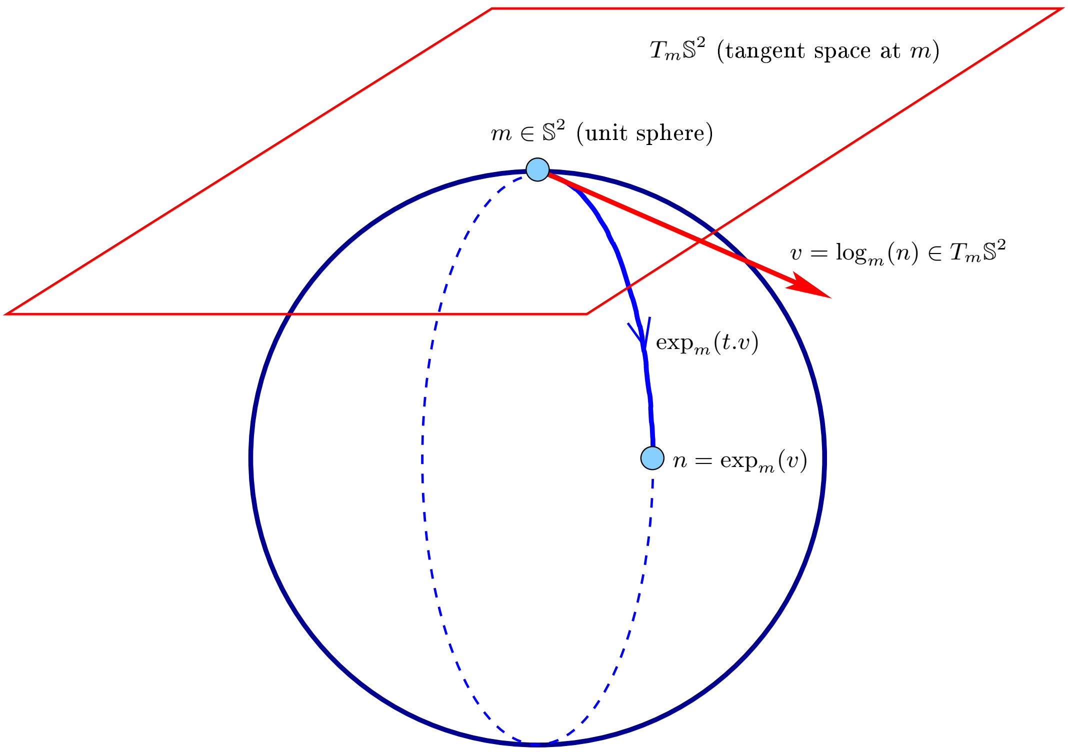

Figure 2.2: **Riemannian exponential and logarithm on the unit sphere** $\mathbb{S}^2$, where the geodesics associated to the usual Riemannian metric are all great circles. Let $v = \log_m(n)$. When t varies between 0 and 1, $\exp_m(t.v)$ moves smoothly between m and n along the great circle between them. The norm of v, drawn in red, is equal to the length of the geodesic between m and n, which is a portion of great circle (called an *orthodrome*), in blue.

Among statistics, the most fundamental is certainly the *mean*. In the general setting of *metric spaces*, i.e. the sets on which a *distance* is defined (e.g., Riemannian manifolds), one can generalize the classical arithmetic mean in a straightforward way.

Indeed, to define a notion of mean value *compatible* with the metric, one can rely on the intuitive idea of *minimal variance or dispersion* to define the mean, because a metric provides a way of quantifying how close (or far away) two elements are from each other in a metric space $(E, dist)$. More precisely, the mean can be defined as the point $\mathbb{E}(X)$ which minimizes some kind of dispersion of the data $(X_i)_{i=1}^N$ around itself (with respect to some non-negative normalized weights (w_i)), for example:

$$\mathbb{E}(X_i) = \underset{Y \in E}{arg\ min} \sum_i w_i.dist(X_i, Y)^\alpha. \tag{2.4}$$

The case $\alpha = 2$ corresponds in vector spaces to the arithmetic mean, and in other spaces to their generalization, called the *Fréchet* mean. For $\alpha = 1$, one obtains the generalization of the *median*. One should note that the dispersion may have several minimizers. For instance, this is classically the case in vector spaces for $\alpha = 1$, essentially because in this case the dispersion it is not strictly convex, contrary to the case $\alpha > 1$. Even when $\alpha > 1$, the dispersion of the data X_i should not be too high in order to guarantee that the dispersion has a unique global minima [Pennec 06a].

Chapter 3

Log-Euclidean Metrics on Tensors

We concentrate in this Chapter and the next on the processing of symmetric positive-definite matrices (SPD matrices), also called by abuse of language 'tensors' in this work. This type of data appears in many contexts, and in the past years the need for a general processing framework for tensors has grown continously.

One should note that although the tensor space is non-linear, it is one of the most simple cases of non-linear manifold that exists. It can be seen as a multidimensional generalization of positive numbers, and has many specific properties that make it extremely close to a vector space. So close that we even show in this Chapter that it *can* be looked upon as a vector space from an non-classical point of view! This remarkably simple non-linear space is therefore a particularly nice pedagogical entry point for the beginner into the harduous topic of Riemannian geometry.

In this Chapter, we present the theoretical properties of a novel and general Riemannian processing framework for tensors, called *Log-Euclidean*. Our approach is based on a *novel Lie group structure* for tensors, which can smoothly be extended into a vector space structure. The novel *algebraic structure* we propose here for symmetric and positive-definite matrices is *compatible* with the usual algebraic properties of this set: the inverse of a tensor is its usual inverse and the matrix exponential is the group exponential of our Lie group structure.

Our novel framework does not introduce any superfluous complexity, since Log-Euclidean Riemannian computations can be converted into Euclidean ones once tensors have been transformed into their matrix logarithms, which makes Euclidean processing algorithms particularly simple to recycle.

A general device based on this principle has been patented by the INRIA [Arsigny 05d] (which does *not* restrict the use of our framework for either research or teaching purposes).

Contents

3.1 Introduction **31**

3.2 Preliminaries **32**

3.2.1 Differential Properties of Matrix Exponential 33

3.2.2 Algebraic Properties of SPD Matrices 33

3.2.3 Differential Properties of SPD Matrices 34

3.2.4 Compatibility Between Algebraic and Differential Properties 34

3.3 Log-Euclidean Means **35**

3.3.1 Multiplication of SPD Matrices 35

3.3.2 Log-Euclidean Metrics on the Lie Group of SPD Matrices 37
3.3.3 A Vector Space Structure on SPD Matrices 39
3.3.4 Log-Euclidean Mean 39
3.4 Comparison with the Affine-Invariant Mean 40
3.4.1 Elementary Metric Operations and Invariance 40
3.4.2 Affine-Invariant Means 40
3.4.3 Geometric Interpolation of Determinants 41
3.4.4 Criterion for the Equality of the Two Means 42
3.4.5 Larger Anisotropy in Log-Euclidean Means 42
3.4.6 Linear and Bilinear Interpolation of SPD Matrices 45
3.5 Probabilities and Statistics with Log-Euclidean Metrics 46
3.5.1 General Riemannian Statistical Framework 47
3.5.2 Random Tensors 47
3.5.3 Fréchet Means and Covariances with Log-Euclidean Metrics 48
3.5.4 General Log-Euclidean Statistical Framework 50
3.6 Conclusions 50

Abstract. In this work we present a novel and general Riemannian processing framework for tensors, called *Log-Euclidean*. Our approach is based on two novel algebraic structures on symmetric positive-definite matrices. First, a Lie group structure which is compatible with the usual algebraic properties of this matrix space. Second, a new scalar multiplication that smoothly extends the Lie group structure into a vector space structure. The Riemannian metrics compatible with these novel algebraic structures are called *Log-Euclidean*.

To evaluate the relevance of a given Riemannian metric, the properties of the associated notion of *mean* are of great importance. Indeed, most computations useful in practice involve averaging procedures. This the reason why we detail in the Chapter the remarkable properties of the Fréchet mean associated to Log-Euclidean metrics, which is simply called the *Log-Euclidean mean*. Interestingly, this means corresponds to an arithmetic mean in the domain of matrix logarithms. We detail the invariance properties of this novel geometric mean and compare it to the recently-introduced affine-invariant mean. The two means have the same determinant and are equal in a number of cases, yet they are not identical in general. One can show that the Log-Euclidean mean has a larger trace whenever they are not equal. Furthermore, the Log-Euclidean mean is much easier to compute.

Finally, we present in this Chapter the Log-Euclidean statistical framework for tensors. As expected, it is particularly simple to use, since Log-Euclidean Riemannian statistics are simply Euclidean statistics on the logarithms of tensors.

Related Publications. The Log-Euclidean framework for tensors was presented at the international conference MICCAI'05 [Arsigny 05a]. The theoretical aspects of this work will be published in the SIAM Journal for Matrix Analysis and Applications [Arsigny 06e], and the application of this framework to diffusion MRI, described in Chapter 4, was published in the international journal Magnetic Resonance in Medicine [Arsigny 06f]. Much of this work was originally published in an INRIA research report [Arsigny 05b].

As a co-author, we have also used the Log-Euclidean framework for tensors to introduce statistics on deformation tensors in non-linear registration, within a framework called

'Riemannian Elasticity' [Pennec 05]. Our Log-Euclidean framework was also used indirectly in [Commowick 05] to better constraint non-linear registration with strain tensors, in an alternative way. Furthermore, our novel framework was used to design a novel framework to jointly estimate and regularize diffusion tensors from diffusion-weighted images, in [Fillard 06a, Fillard 05b, Fillard 06b].

Moreover, still as a co-author, we have contributed to better process and analyse *variability tensors*, in the context of the modeling of the local variability of the human brain. Using this time the affine-invariant framework for tensor processing, this work was presented in [Fillard 05c, Fillard 06c, Fillard 06d]. Still with the affine-invariant framework, we have also explored the interest of Riemannian processing on *structure tensors* [Fillard 05a].

Publications by other Authors. In recent months, our Log-Euclidean framework for tensors has begun to be used on a regular basis by several teams, in various contexts. For example, in our research team, it is used in [Peyrat 06b, Peyrat 06a] to study the statistical variability of the diffusion tensors of canine hearts. At Rutgers university, in [Norris 06], the Log-Euclidean framework is used to compute the isotropic material closest to a given anisotropic material, in the context of continuum mechanics. In [Goodlett 06], at the University of North Carolina, the Log-Euclidean framework is used to generate unbiased atlases of diffusion tensor images of the human brain. At the University of California in Los Angeles, the Log-Euclidean framework is used in [Lepore 06] to perform statistical tests on deformation tensors steming from a registration procedure to compare morphologically two groups of individuals (HIV/AIDS patients vs. normals).

3.1 Introduction

In Section 1.1.1, we have seen that symmetric positive-definite matrices of real numbers appear in many contexts and have been increasingly used in medical imaging, in particular in diffusion MRI. As a consequence, there has been a growing need to carry out computations with these objects, for instance to interpolate, restore, enhance images of symmetric positive-definite matrices. To this end, one needs to define a complete operational framework. This is necessary to fully generalize to the SPD case the usual statistical tools or PDEs on vector-valued images. The framework of Riemannian geometry [Gallot 93] is particularly adapted to this task, since many statistical tools [Pennec 06a] and PDEs can be generalized this framework.

As we have seen in Section 2.4.4, the classical generalization of the Euclidean mean to Riemannian manifolds is called the Fréchet mean. Most computations useful in practice involve averaging procedures, and the properties of the mean are therefore of great importance. Means are computed either explicited or implicited for example during the interpolation, the regularization and the extrapolation of SPD matrices. A simple illustration of this fact is given by the classical regularization technique based on the heat equation, which is equivalent to the convolution of the original data with Gaussian kernels.

To define a Riemannian metric on the space of SPD matrices, one can directly use the usual Euclidean structure on square matrices. This is straightforward, and in this setting, the Riemannian mean of a system of SPD matrices is their *arithmetic* mean, which is an SPD matrix since SPD matrices form a convex set. However, as we pointed out in the Introduction of this thesis, although Euclidean distances are well-adapted to general square matrices, they are unsatisfactory for tensors, for two main reasons. First, symmetric matrices with null or negative eigenvalues typically appear on clinical DT-MRI data as soon as we perform on tensors Euclidean operations which are non-convex, which is unacceptable in many cases, like DT-MRI. Second, Euclidean averaging of SPD matrices leads often to a *swelling effect*: the determinant of the Euclidean mean can be strictly larger than the original determinants, which is not acceptable in DT-MRI where this amounts to introducing more diffusion in the images.

To fully circumvent these difficulties, other metrics have been recently proposed for SPD matrices. With the affine-invariant metrics proposed in [Pennec 06b,Fletcher 04a,Lenglet 06a, Moakher 05], negative and null eigenvalues are at an infinite distance. The swelling effect has disappeared, and the symmetry with respect to inversion is respected. These new metrics lead to the definition of an affine-invariant generalization of the geometric mean of positive numbers on symmetric positive-definite matrices. But the price paid for this success is a high computational burden in practice, essentially due to the curvature induced on the space of symmetric positive-definite matrices. This leads in many cases to slow and hard to implement algorithms (especially for PDEs) [Pennec 06b].

We propose here a new Riemannian framework on tensors, which leads to a novel generalization of the geometric mean to SPD matrices. It fully overcomes these computational limitations while conserving excellent theoretical properties. This is obtained with a new family of metrics named *Log-Euclidean*. Such metrics are particularly simple to use. They result in classical Euclidean computations in the domain of matrix logarithms. As a consequence, there is a closed form for the Log-Euclidean mean, contrary to the affine-invariant case. It results in a drastic reduction of computation times: the Log-Euclidean mean is approximately 20 times faster to compute.

The remainder of this Chapter is organized as follows. In Section 3.2, we present some fundamental properties of the space of symmetric positive-definite matrices which are used afterwards. Then we proceed in Section 3.3 to the theory of Log-Euclidean metrics which is based on two novel algebraic structures on SPD matrices: a Lie group structure and a new scalar multiplication which complements the new multiplication to obtain a new vector space structure. The definition of the Log-Euclidean mean is deduced from these new structures. Contrary to the affine-invariant mean, there is a closed form for the Log-Euclidean mean and it is simple to compute. In Section 3.4 we highlight the similarities and differences between affine-invariant and Log-Euclidean means. They are quite similar, since they have the same determinant, which is the classical geometric mean of the determinants of the averaged SPD matrices. They even coincide in a number of cases, and yet are different in general. We prove that Log-Euclidean means are strictly more anisotropic when averaged SPD matrices are isotropic enough. Last but not least, we present the general statistical Log-Euclidean framework for tensors, which is particularly simple to use, since Log-Euclidean Riemannian statistics are simply Euclidean statistics on the logarithms of tensors.

3.2 Preliminaries

We begin by describing the fundamental properties of the tensor space used in this Chapter. First, we present the differential properties of the matrix exponential. Then, we examine the general properties of SPD matrices. These properties are of two types: algebraic and differential. On the one hand, SPD matrices have algebraic properties because they are a special kind of invertible matrices, and on the other hand they can be considered globally as a smooth manifold and therefore have differential geometry properties. These properties are not independent: on the contrary, they are compatible in a profound way. This compatibility is the heart of the approach developed here.

3.2.1 Differential Properties of Matrix Exponential

The exponential plays a central role in *Lie groups* (see [Bourbaki 89, Sternberg 64, Gallot 93]). We consider here only the matrix version of the exponential, which is a tool that we extensively use in the next sections. Its fundamental properties are detailled in Chapter 2, Section 2.3. In the following, we describe some of its classical differential properties.

Theorem 3.1. exp : $M(n) \to GL(n)$ *is a* $\mathcal{C}^\infty$ *mapping. Its differential map at a point* $M \in M(n)$ *acting on an infinitesimal displacement* $dM \in M(n)$ *is given by:*

$$D_M \exp .dM = \sum_{k=1}^{\infty} \frac{1}{k!} \left(\sum_{l=0}^{k-1} M^{k-l-1}.dM.M^l \right). \tag{3.1}$$

Proof. The smoothness of exp is simply a consequence of the uniform absolute convergence of its series expansion in any compact set of $M(n)$. The differential is obtained classically by a term by term derivation of the series defining the exponential. □

We see here that the non-commutativity of the matrix multiplication complicates seriously the differentiation of the exponential, which is much simpler in the scalar case. However, taking the trace in Eq. (3.1) yields:

Corollary 3.1. *We have the following simplification in terms of traces:*

$$\mathrm{Trace}(D_M \exp .dM) = \mathrm{Trace}(\exp(M).dM). \tag{3.2}$$

We shall also use in the following this property on determinants:

Proposition 3.1. *Let* $M \in M(n)$. *Then* $\det(\exp(M)) = \exp(\mathrm{Trace}(M))$.

Proof. This is easily seen in terms of eigenvalues of M. The Jordan decomposition of M [Lang 04] assure that $\mathrm{Trace}(M)$ is the sum of its eigenvalues. But the exponential of a triangular matrix transforms the diagonal values of this matrix into their scalar exponential. The determinant of $\exp(M)$ is simply the product of its eigenvalues, which is precisely the exponential of the trace of M. □

3.2.2 Algebraic Properties of SPD Matrices

SPD Matrices have remarkable algebraic properties. First, there always exists a unique real and *symmetric* logarithm for any SPD matrix, which is its principal logarithm. Second, if the space of SPD matrices is not a subgroup of $GL(n)$, it is stable with respect to inversion. Moreover, its spectral decomposition is particularly simple[1].

Theorem 3.2. *For any* $S \in Sym(n)$, *there exists an orthonormal coordinate system in which* S *is diagonal. This is in particular the case for SPD matrices.* $Sym^+_\star(n)$ *is not a subgroup of* $GL(n)$, *but it is stable by inversion. Moreover, the matrix exponential* exp : $Sym(n) \to Sym^+_\star(n)$ *is one-to-one and onto.*

Proof. For a proof of the first assertion, see elementary linear algebra manuals, or [Lang 04]. For the second assertion, we see from Section 2.3.4 that SPD matrices have a unique real logarithm whose eigenvalues have an imaginary part between $-\pi$ and $+\pi$, since the eigenvalues of SPD matrices are real and always positive. It is simple to see that this real logarithm, called the principal logarithm, can be obtained simply by replacing its eigenvalues by their natural logarithms, which always yields a symmetric matrix. □

[1] This is due to the fact that SPD matrices are *normal operators*, like rotations and antisymmetric matrices [Lang 04].

Thanks to the existence of an orthonormal basis in which an SPD matrix (resp. a symmetric matrix) is diagonal, the logarithms (resp. the exponential) has a particularly simple expression. In such a basis, taking the log (resp. the exp) of a is simply done by applying their scalar version to eigenvalues:

$$\begin{cases} \log(R.\mathrm{Diag}(\lambda_1, ..., \lambda_N).R^T) = R.\mathrm{Diag}(\log(\lambda_1), ..., \log(\lambda_N)).R^T \\ \exp(R.\mathrm{Diag}(\lambda_1, ..., \lambda_N).R^T) = R.\mathrm{Diag}(\exp(\lambda_1), ..., \exp(\lambda_N)).R^T. \end{cases}$$

These formulae provide a particularly efficient method to calculate the logarithms and exponentials of symmetric matrices, whenever the cost of a diagonalization is less than that of the many matrix multiplications (in the case of the exponential) and inversions (in the case of the logarithm) used in the general matrix case by classical algorithms [Higham 05, Cheng 01]. For small values of n, and in particular $n = 3$, we found such formulae to be extremely useful.

3.2.3 Differential Properties of SPD Matrices

This time from the point of view of topology and differential geometry, the space of SPD matrices has also many particularities. The properties recalled here are elementary and will not be detailed. See [Arsigny 05b] for complete proofs.

Proposition 3.2. *$Sym_\star^+(n)$ is an open convex half-cone of $Sym(n)$ and is therefore a submanifold of $Sym(n)$, whose dimension is $n(n+1)/2$.*

3.2.4 Compatibility Between Algebraic and Differential Properties

We have seen that exp is a smooth bijection. We show here that the logarithm, i.e. its inverse, is also smooth. As a consequence, all the algebraic operations on SPD matrices presented before are also smooth, in particular the inversion. Thus, the two structures are fully compatible.

Theorem 3.3. log : *$Sym_\star^+(n) \to Sym(n)$ is $\mathcal{C}^\infty$. Thus,* exp *and its inverse* log *are both smooth, i.e. they are diffeomorphisms. This is due to the fact that the differential of* exp *is nowhere singular.*

Proof. In fact, we only need to prove the last assertion. If it is true, the Implicit Function Theorem [Schwartz 97] applies and assures that log is also smooth. Since the differential of exp at 0 is simply given by the identity, it is invertible by continuity in a neighborhood of 0. We now show that this propagates to the entire space $Sym(n)$. Indeed, let us then suppose that for a point M, the differential $D_{M/2}\exp$ is invertible. We claim that then $D_M \exp$ is also invertible, which suffices to prove the point. To show this, let us take $dM \in Sym(n)$ such that $D_M \exp .dM = 0$. If $D_M \exp$ is invertible, we should have $dM = 0$. To see this, remark that $\exp(M) = \exp(M/2).\exp(M/2)$. By differentiation and applying to dM, we get:

$$D_M \exp .dM = 1/2((D_{M/2} \exp .dM).\exp(M/2) + \exp(M/2).(D_{M/2} \exp .dM)) = 0.$$

This implies by multiplication by $\exp(-M/2)$:

$$\exp(-M/2)(D_{M/2} \exp .dM).\exp(M/2) + (D_{M/2} \exp .dM) = 0.$$

Since $A^{-1}.\exp(B).A = \exp(A^{-1}.B.A)$ we have also by differentiation: $A^{-1}.D_B \exp(dB).A = D_B \exp(A^{-1}.dB.A)$. Using this simplification and the hypothesis that $D_{M/2} \exp$ is invertible, we obtain:

$$\exp(-M/2).dM.\exp(M/2) + dM = 0.$$

Let us rewrite this equation in an orthonormal basis in which M is diagonal with a rotation matrix R. Let (λ_i) be the eigenvalues of M and let $dN := R.dM.R^T$. Then we have:

$$dN = -\mathrm{Diag}(\exp(-\lambda_1/2), ..., \exp(-\lambda_N/2)).dN.\mathrm{Diag}(\exp(\lambda_1/2), ..., \exp(\lambda_N/2)).$$

Coordinate by coordinate this is writen as:

$$\forall i, j : dN_{i,j}(1 + \exp(-\lambda_i/2 + \lambda_j/2)) = 0.$$

Hence for all i, j : $dN_{i,j} = 0$ which is equivalent to $dM = 0$. And we are done. □

Corollary 3.2. *In the space of SPD matrices, for all $\alpha \in \mathbb{R}$, the power mapping: $S \mapsto S^\alpha$ is smooth. In particular, this is true for the inversion mapping (i.e. when $\alpha = -1$).*

Proof. We have $S^\alpha = \exp(\alpha \log(S))$. The composition of smooth mappings is smooth. □

3.3 Log-Euclidean Means

We focus in this section on the construction of Log-Euclidean means. They are derived from two new structures on SPD matrices.

The first is a Lie group structure [Bourbaki 89] i.e. an algebraic group structure that is *compatible* with the differential structure of the Space of SPD Matrices. The second structure is a *vector space structure*. Indeed, one can define a *logarithmic scalar multiplication* that complements the Lie group structure to form a vector space structure on the space of SPD matrices. In this context, Log-Euclidean metrics are defined as bi-invariant metrics on the Lie group of SPD matrices. The Log-Euclidean mean is the Fréchet mean associated to these metrics. It is particularly simple to compute.

3.3.1 Multiplication of SPD Matrices

It is *a priori* not obvious how one could define a multiplication on the space of SPD matrices compatible with classical algebraic and differential properties. How can one combine smoothly two SPD matrices to make a third one, in such a way that Id is still the identity and the usual inverse remains its inverse? Moreover, if we obtain a new Lie group structure, we would also like the matrix exponential to be the exponential associated to the Lie group structure, which *a priori* can be different.

The first idea that comes to mind is to use directly matrix multiplication. But then the non-commutativity of matrix multiplication between SPD matrices stops the attempt: if $S_1, S_2 \in Sym^+_\star(n)$, $S_1.S_2$ is an SPD matrix (or equivalently, is symmetric) if and only if S_1 and S_2 commute. To overcome the possible asymmetry of the matrix product of two SPD matrices, one can simply take the *symmetric part* (i.e. the closest symmetric matrix in the sense of the Frobenius norm [Higham 89]) of the product and define the new product $\diamond$:

$$S_1 \diamond S_2 := \frac{1}{2}(S_1.S_2 + S_2.S_1).$$

This multiplication is smooth, conserves the identity and the inverse. But $S_1 \diamond S_2$ is not necessarily positive! And since the subset of SPD matrices is not closed in $Sym(n)$, one cannot define in general a closest SPD matrix, but only a closest symmetric *semi-definite* matrix [Higham 89].

In [Pennec 06b], affine-invariant distances between two SPD matrices S_1, S_2 are of the form:

$$d(S_1, S_2) = \| \log(S_1^{-1/2}.S_2.S_1^{-1/2}) \|. \tag{3.3}$$

where $\|.\|$ is a Euclidean norm defined on $Sym(n)$. Let us define the following multiplication $\circledcirc$:

$$S_1 \circledcirc S_2 := S_1^{1/2}.S_2.S_1^{1/2}.$$

With this multiplication, the affine-invariant metric constructed in [Pennec 06b] can be interpreted then as a left-invariant metric. Moreover, this multiplication is smooth, compatible with matrix inversion and matrix exponential, and the product truly defines an SPD matrix. Everything works fine, except that it is not *associative*. This makes everything fail, because associativity is an essential requirement of group structure. Without it, many fundamental properties disappear. For Lie groups, the notion of adjoint representation does not exist anymore without associativity.

Theorem 3.3 points to an important fact: $Sym_\star^+(n)$ is diffeomorphic to its tangent space at the identity, $Sym(n)$. But $Sym(n)$ has an additive group structure, and to obtain a group structure on the space of SPD matrices, one can simply transport the additive structure of $Sym(n)$ to $Sym_\star^+(n)$ with the exponential. More precisely, we let:

Definition 3.1. *Let $S_1, S_2 \in Sym_\star^+(n)$. We define their logarithmic product $S_1 \odot S_2$ by:*

$$S_1 \odot S_2 := \exp(\log(S_1) + \log(S_2)). \tag{3.4}$$

Proposition 3.3. *$(Sym_\star^+(n), \odot)$ is a group. The neutral element is the usual identity matrix, and the group inverse of an SPD matrix is its inverse in the matrix sense. Moreover, whenever two SPD matrices commute in the matrix sense, then the logarithmic product is equal to their matrix product. Furthermore, the multiplication is commutative.*

Proof. The multiplication is defined by addition on logarithms. It is therefore associative and commutative. Since $\log(\mathrm{Id}) = 0$, the neutral element is Id and since $\log(S^{-1}) = -\log(S)$, the new inverse is the matrix inverse. Finally, we have $\exp(\log(S_1) + \log(S_2)) = \exp(\log(S_1)).\exp(\log(S_2)) = S_1.S_2$ when $[S_1, S_2] = 0$. □

Theorem 3.4. *The logarithmic multiplication $\odot$ on $Sym_\star^+(n)$ is compatible with its structure of smooth manifold: $(S_1, S_2) \mapsto S_1 \odot S_2^{-1}$ is $\mathcal{C}^\infty$. Therefore, $Sym_\star^+(n)$ is given a commutative Lie group structure by $\odot$.*

Proof. $(S_1, S_2) \mapsto S_1 \odot S_2^{-1} = \exp(\log(S_1) - \log(S_2))$. But since exp and log and the addition are smooth, their composition is also smooth. By definition (see [Gallot 93], page 29), $Sym_\star^+(n)$ is a Lie group. □

Proposition 3.4. $\exp : (Sym(n), +) \to (Sym_\star^+(n), \odot)$ *is a Lie group isomorphism. In particular, one-parameter subgroups of $Sym_\star^+(n)$ are obtained by taking the matrix exponential of those of $Sym(n)$, which are simply of the form $(t.V)_{t\in\mathbb{R}}$ where $V \in Sym(n)$. As a consequence, the Lie group exponential in $Sym_\star^+(n)$ is given by the classical matrix exponential on the Lie Algebra $Sym(n)$.*

Proof. We have explicitly transported the group structure of $Sym(n)$ into $Sym_\star^+(n)$ so exp is a morphism. It is also a bijection, and thus an isomorphism. The smoothness of exp then assures its compatibility with the differential structure.

Let us recall the definition of one-parameter subgroups. $(S(t))_{t\in\mathbb{R}}$ is such a subgroup if and only if we have $\forall t, s : S(t+s) = S(t) \odot S(s) = S(s) \odot S(t)$. But then $\log(S(t+s) = \log(S(t) \odot S(s)) = \log(S(t)) + \log(S(s))$ by definition of $\odot$. Therefore $\log S(t)$ is also a one-parameter subgroup of $(Sym(n), +)$, which is necessarily of the form $t.V$ where $V \in Sym(n)$. V is the *infinitesimal generator* of $S(t)$. Finally, the exponential is obtained from one-parameter subgroups, which are all of the form $(\exp(t.V))_{t\in\mathbb{R}}$ (see [Sternberg 64], Chap. V). □

Thus, we have given the space of SPD matrices a structure of Lie group that leaves unchanged the classical matrix notions of inverse and exponential. The new multiplication used, i.e. the logarithmic multiplication, generalizes the matrix multiplication when two SPD matrices do not commute in the matrix sense.

The associated Lie Algebra is the space of symmetric matrices, which is diffeomorphic and isomorphic to the group itself. The associated Lie bracket is the null bracket: $[S_1, S_2] = 0$ for all $S_1, S_2 \in Sym(n)$.

To our knowledge, this Lie group structure is new in the literature. For a space as commonly used as SPD matrices, this is quite surprising. Probably, the reason is that the Lie group of SPD Matrices is *not* a multiplicative matrix group, contrary to most Lie groups.

3.3.2 Log-Euclidean Metrics on the Lie Group of SPD Matrices

Now that we have given $Sym_\star^+(n)$ a structure of Lie group, we turn to the task of exploring metrics compatible with this new structure. Among Riemannian metrics in Lie groups, *bi-invariant* metrics are the most convenient, as will be detailled in Chapter 7. In the case of our Lie group structure for tensors, the situation regarding bi-invariant metrics is particularly simple, since our multiplication $\odot$ is *commutative.*

Proposition 3.5. *Any metric* $<,>$ *on* $T_{Id}Sym_\star^+(n) = Sym(n)$ *extended to* $Sym_\star^+(n)$ *by left- or right-multiplication is a bi-invariant metric.*

Proof. The commutativity of the multiplication implies that $Ad(Sym_\star^+(n)) = \{\mathrm{Id}\}$, which is trivially an isometry group. ☐

Definition 3.2. *Any bi-invariant metric on the lie group of SPD matrices is also called a Log-Euclidean metric, because it corresponds to a Euclidean metric in the logarithmic domain as is shown in Corollary 3.3.*

Corollary 3.3. *Let* $<,>$ *be a bi-invariant metric on* $Sym_\star^+(n)$*. Then its geodesics are simply given by the translated versions of one-parameter subgroups, namely:*

$$(\exp(V_1 + t.V_2))_{t\in\mathbb{R}} \text{ where } V_1,\ V_2 \in Sym(n). \tag{3.5}$$

The exponential and logarithmic maps associated to the metric can be expressed in terms of matrix exponential and logarithms in the following way:

$$\begin{cases} \log_{S_1}(S_2) = D_{\log(S_1)}\exp.(\log(S_2) - \log(S_1)) \\ \exp_{S_1}(L) = \exp(\log(S_1) + D_{S_1}\log.L). \end{cases} \tag{3.6}$$

The scalar product between two tangent vectors $V_1,\ V_2$ *at a point* S *is given by:*

$$< V_1, V_2 >_S = < D_S \log.V_1, D_S \log.V_2 >_{Id}. \tag{3.7}$$

From this equation, we get the distance between two SPD matrices:

$$d(S_1, S_2) = \|\log_{S_1}(S_2)\|_{S_1} = \|\log(S_2) - \log(S_1)\|_{Id}. \tag{3.8}$$

where $\|.\|$ *is the norm associated to the metric.*

Proof. Theorem 7.1 in Chapter 7 states that geodesics are obtained by translating one-parameter subgroups and Prop. 3.4 gives the form of these subgroups in terms of matrix exponential. By definition, the metric exponential $\exp_{S_1} : T_{S_1}Sym_\star^+(n) \to Sym_\star^+(n)$ is the mapping that associates to a tangent vector L the value at time 1 of the geodesic starting at time 0 from S_1 with an initial speed vector L. Differentiating the geodesic equation Eq. (3.5) at time 0 yields an initial vector speed equal to $D_{V_1} \exp .V_2$. As a consequence, $\exp_{S_1}(L) = \exp(\log(S_1) + (D_{\log(S_1)} \exp)^{-1}.L)$. The differentiation of the equality $\log \circ \exp = \mathrm{Id}$ yields: $(D_{\log(S_1)} \exp)^{-1} = D_{S_1} \log$. Hence the formula for $\exp_{S_1}(L)$. Solving in L the equation $\exp_{S_1}(L) = S_2$ provides the formula for $\log_{S_1}(S_2)$.

The metric at a point S is obtained by propagating by translation the scalar product on the tangent space at the identity. Let $L_S : Sym_\star^+(n) \to Sym_\star^+(n)$ be the logarithmic multiplication by S. We have: $< V_1, V_2 >_S = < D_S L_{S^{-1}}.V_1, D_S L_{S^{-1}}.V_2 >$. But simple computations show that $D_S L_{S^{-1}} = D_S \log$. Hence Eq. (3.7). Finally, we combine Eq. (3.6) and Eq. (3.7) to obtain the (simple this time!) formula for the distance. □

Corollary 3.4. *Endowed with a bi-invariant metric, the space of SPD matrices is a flat Riemannian space: its sectional curvature (see [Gallot 93], page 107) is null everywhere.*

Proof. This is clear, since it is isometric to the space $Sym(n)$ endowed with the Euclidean distance associated to the metric. □

In [Pennec 06b], the metric defined on the space of SPD matrices is affine-invariant. The action $\mathrm{act}(A)$ of an invertible matrix A on the space of SPD matrices is defined by:

$$\forall S,\ \mathrm{act}(A)(S) = A.S.A^T.$$

Affine-invariance means that for all invertible matrix A, the mapping $\mathrm{act}(A) : Sym_\star^+(n) \to Sym_\star^+(n)$ is an isometry. This group action describes how an SPD matrix, assimilated to a covariance matrix, is affected by a general affine change of coordinates.

Here, the Log-Euclidean Riemannian framework will not yield full affine-invariance. However, it is not far from it, because we can obtain invariance by similarity (isometry plus scaling).

Proposition 3.6. *We can endow $Sym_\star^+(n)$ with a similarity-invariant metric, for instance by choosing $< V_1, V_2 > := Trace(V_1.V_2)$ for $V_1, V_2 \in Sym(n)$.*

Proof. Let $R \in SO(n)$ be a rotation and $s > 0$ be a scaling factor. Let S be an SPD matrix. V is transformed by the action of $s.R$ into $\mathrm{act}(sR)(S) = s^2.R.S.R^T$. From Eq. (3.8), the distance between two SPD matrices S_1 and S_2 transformed by sR is:

$$d(\mathrm{act}(sR)(S_1), \mathrm{act}(sR)(S_2)) = \mathrm{Trace}(\{\log(\mathrm{act}(sR)(S_1)) - \log(\mathrm{act}(sR)(S_2))\}^2).$$

A scaling by a positive factor λ on an SPD matrix corresponds to a translation by $\log(\lambda).\mathrm{Id}$ in the domain of logarithms. Furthermore, we have $\log(R.S.R^T) = R.\log(S).R^T$ for any SPD matrix S and any rotation R. Consequently, the scaling zeros out in the previous formula and we have:

$$\begin{aligned} d(\mathrm{act}(sR)(S_1), \mathrm{act}(sR)(S_2)) &= \mathrm{Trace}(\{R.(\log(S_1) - \log(S_2)).R^T\}^2) \\ &= \mathrm{Trace}(\{\log(S_1) - \log(S_2)\}^2) \\ &= d(S_1, S_2). \end{aligned}$$

Hence the result. □

Thus, we see that the Lie group of SPD Matrices with an appropriate Log-Euclidean metric has many *invariance properties*: Lie group bi-invariance and similarity-invariance. Moreover, Theorem 7.1 in Chapter 7 shows that the the inversion mapping: $S \mapsto S^{-1}$ is an isometry.

3.3.3 A Vector Space Structure on SPD Matrices

We have already seen that the Lie group of SPD Matrices is isomorphic and diffeomorphic to the additive group of symmetric matrices. We have also seen that with a Log-Euclidean metric, the Lie group of SPD Matrices is also isometric to the space of symmetric matrices endowed with the associated Euclidean metric. There is more: the Lie group isomorphism exp from the Lie Algebra of symmetric matrices to the space of SPD matrices can be smoothly extended into an isomorphism of vector spaces. Indeed, let us define the following operation:

Definition 3.3. *The logarithmic scalar multiplication $\circledast$ of an SPD matrix by a scalar $\lambda \in \mathbb{R}$ is:*

$$\lambda \circledast S = \exp(\lambda . \log(S)) = S^{\lambda}. \tag{3.9}$$

When we assimilate the logarithmic multiplication to an addition and the logarithmic scalar multiplication to a usual scalar multiplication, we have all the properties of a vector space. By construction, the mapping $\exp : (Sym(N), +, .) \rightarrow (Sym^+_\star(n), \odot, \circledast)$ is a vector space isomorphism. Since all algebraic operations on this vector space are smooth, this defines what could be called a 'Lie vector space structure' on SPD matrices.

Of course, this result does not imply that the space of SPD matrices is a vector subspace of the vector space of square matrices. But it shows that we can view this space as a vector space when we identify a SPD matrix to its logarithm. The question of whether or not the SPD matrix space is a vector space depends on the vector space structure we are considering, and *not* on the space itself.

From this point of view, bi-invariant metrics on the Lie group of SPD Matrices are simply the classical Euclidean metrics on the vector space $(Sym(n), +, .)$. Thus, we have in fact defined a new Euclidean structure on the space of SPD matrices by transporting that of its Lie Algebra $Sym(n)$ on SPD matrices. But this Euclidean structure does not have the defects mentioned in the introduction of this Chapter: matrices with null eigenvalues are at infinite distance and the symmetry principle is respected. Moreover, with an appropriate metric, similarity-invariance is also guaranteed.

3.3.4 Log-Euclidean Mean

We present here the definition of the Log-Euclidean mean of SPD matrices and its invariance properties.

Theorem 3.5. *Let $(S_i)_1^N$ be a finite number of SPD matrices. Then their Log-Euclidean Fréchet mean exists and is unique. It is given explicitly by:*

$$\mathbb{E}_{LE}(S_1, ..., S_N) = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log(S_i)\right). \tag{3.10}$$

The Log-Euclidean mean is similarity-invariant, invariant by group multiplication and inversion, and is exponential-invariant (i.e. invariant with respect to scaling in the domain of logarithms).

Proof. When one expresses distances in the logarithm domain, one is faced with the classical computation of an Euclidean mean. Hence the formula by mapping back the results with exp in the domain of SPD matrices. Now, this mean does not depend on the chosen Log-Euclidean metric, and since there exist similarity-invariant metrics among Log-Euclidean metrics, this property propagates to the mean. The three last invariance properties are reformulations in the domain of SPD matrices of classical properties of the arithmetic mean in the domain of logarithms. □

Affine-Invariant Metrics	**Log-Euclidean Metrics**
Riemannian Exponential: $\exp_{S_1}(L) =$	
$S_1^{1/2}.\exp(S_1^{-1/2}.L.S_1^{-1/2}).S_1^{1/2}$	$\exp(\log(S_1) + D_{S_1}\log.L)$
Riemannian Logarithm: $\log_{S_1}(S_2) =$	
$S_1^{1/2}.\log(S_1^{-1/2}.S_2.S_1^{-1/2}).S_1^{1/2}$	$D_{\log(S_1)}\exp.(\log(S_2) - \log(S_1))$
Dot product: $< L_1, L_2 >_S =$	
$< S^{-1/2}.L_1.S^{-1/2}, S^{-1/2}.L_2.S^{-1/2} >_{\mathrm{Id}}$	$< D_S\log.L_1, D_S\log.L_2 >_{\mathrm{Id}}$
Distance: $d(S_1, S_2) =$	
$\|\log(S_1^{-1/2}.S_2.S_1^{-1/2})\|$	$\|\log(S_2) - \log(S_1)\|$
Geodesic between S_1 and S_2:	
$S_1^{1/2}.\exp(tW).S_1^{1/2}$ with $W = \log\left(S_1^{-1/2}.L.S_1^{-1/2}\right)$	$\exp((1-t)\log(S_1) + t\log(S_2))$
Invariance properties	
Affine-invariance	Lie group bi-invariance, Similarity-invariance

Table 3.1: Comparison between affine-invariant and Log-Euclidean metrics. Note on the one hand the important simplifications in terms of distance and geodesics in the Log-Euclidean case. On the other hand, this results in the use of the differentials of the matrix exponential and logarithm in the exponential and logarithm maps.

3.4 Comparison with the Affine-Invariant Mean

We compare in this section the Log-Euclidean mean to the recently-introduced affine-invariant mean [Pennec 06b, Moakher 05, Lenglet 06a, Fletcher 04a]. To this end, we first recall the differences between affine-invariant metrics and Log-Euclidean metrics in terms of elementary operators, distance and geodesics. Then we turn to a study of the algebraic properties of Fréchet means in the Log-Euclidean and affine-invariant cases.

3.4.1 Elementary Metric Operations and Invariance

Distances, geodesics and Riemannian means take a much simpler form in the Log-Euclidean than in the affine-invariant case. Invariance properties are comparable: some Log-Euclidean metrics are not only bi-invariant but also similarity-invariant. These properties are summarized in Table 3.1. However, we see in this table that the Riemannian exponential and logarithmic are complicated in the Log-Euclidean case by the use of the differentials of the matrix exponential and logarithm. This is the price to pay to obtain simple distances and geodesics. Interestingly, using spectral properties of symmetric matrices, one can obtain a closed form for the differential of both matrix logarithm and exponential and it is possible compute them very efficiently. See [Fillard 05b] for more details.

3.4.2 Affine-Invariant Means

Let $(S_i)_{i=1}^N$ be a system of SPD matrices. Contrary to the Log-Euclidean case, there is in general no closed form for the affine-invariant Fréchet mean $E_{Aff}(S_1, ..., S_N)$ associated to affine-invariant metrics. The affine-invariant mean is defined *implicitly* by a *barycentric* *equation*, which is the following:

$$\sum_{i=1}^{N} \log(\mathbb{E}_{Aff}(S_1, ..., S_N)^{-1/2}.S_i.\mathbb{E}_{Aff}(S_1, ..., S_N)^{-1/2}) = 0. \tag{3.11}$$

This equation is equivalent to the following other barycentric equation, given in [Moakher 05]

$$\sum_{i=1}^{N} \log(\mathbb{E}_{Aff}(S_1, ..., S_N)^{-1}.S_i) = 0. \tag{3.12}$$

The two equations are equivalent simply because for all i:

$$\mathbb{E}_{Aff}(S_1, ..., S_N)^{-1/2}.S_i.\mathbb{E}_{Aff}(S_1, ..., S_N)^{-1/2} = A.\mathbb{E}_{Aff}(S_1, ..., S_N)^{-1}.S_i.A^{-1}$$

with $A = \mathbb{E}_{Aff}(S_1, ..., S_N)^{-1/2}$. The fact that $\log(A.S.A^{-1}) = A.\log(S).A^{-1}$ for any SPD matrix S and any invertible matrix A suffices to conclude.

To solve Eq. (3.11), the only known strategy is to resort to an iterative numerical procedure, such as the Gauss-Newton gradient descent method described in [Pennec 06b].

3.4.3 Geometric Interpolation of Determinants

The definition of the Log-Euclidean mean given by Eq. (3.10) is extremely similar to that of the classical scalar geometrical mean. We classically have:

Definition 3.4. *The geometrical mean of positive numbers* $d_1, ..., d_N$*,is given by*

$$\mathbb{E}(d_1, ..., d_N) = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log(d_i)\right).$$

The Log-Euclidean and affine-invariant Fréchet means can *both* be considered as generalizations of the geometric mean. Indeed, their determinants are both equal to the scalar geometric mean of the determinants of the original SPD matrices. This fundamental property can be thought of as the *common property that should have all generalizations of the geometric mean* to SPD matrices.

Theorem 3.6. *Let* $(S_i)_{i=1}^{N}$ *be* N *SPD matrices. Then the determinant of their Log-Euclidean and affine-invariant means is the geometric mean of their determinants.*

Proof. From Proposition 3.1 we know that $\det(\exp(M)) = \exp(\mathrm{Trace}(M))$ for any square matrix M. Then for the geometric mean, we get:

$$\begin{aligned}
\det(\mathbb{E}_{LE}(S_1, ..., S_N)) &= \exp(\mathrm{Trace}(\log(\mathbb{E}_{LE}(S_1, ..., S_N)))) \\
&= \exp\left(\mathrm{Trace}(\frac{1}{N}\sum_{i=1}^{N} \log(S_i))\right) \\
&= \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log(\det(S_i))\right) \\
&= \exp\left(\mathbb{E}(\log(\det(S_1, ..., S_N)))\right).
\end{aligned}$$

For affine-invariant means, there is no closed form for the mean. *But* there is the barycentric equation given by Eq. (3.11). By applying the same formula as before after having taken the exponential and using $\det(S.T) = \det(S).\det(T)$ we obtain the result. □

Theorem 3.6 shows that the Log-Euclidean and affine-invariant means of SPD matrices are quite similar. In terms of interpolation, this result is satisfactory, since it implies that the interpolated determinant, i.e. the volume of the associated interpolated ellipsoids, will vary between the values of the determinants of the source SPD matrices. Indeed, we have:

Corollary 3.5. *Let $(S_i)_{i=1}^N$ be N SPD matrices. Then the determinant of their Log-Euclidean and affine-invariant means are within the interval* $[\inf_{i \in 1 \ldots N}(S_i), \sup_{i \in 1 \ldots N}(S_i)]$.

Proof. This is simply a consequence of the monotonicity of the scalar exponential and of the scalar integral. □

Corollary 3.6. *Let S_1 and S_2 be two SPD matrices. The geodesic interpolations provided by the affine-invariant and Log-Euclidean metrics lead to a geometric interpolation of determinants. As a consequence, this interpolation of determinants is monotonic.*

Proof. Indeed, in both cases, the interpolated determinant $\mathrm{Det}(t)$ is the geometric mean of the two determinants, i.e. at $t \in [0, 1]$: $\mathrm{Det}(t) = \exp((1-t)\log(\det(S_1)) + t\log(\det(S_2)))$. This interpolation is monotonic, since the differentiation yields:

$$\frac{d}{dt}\mathrm{Det}(t) = \mathrm{Det}(t)\log(\det(S_2.S_1^{-1})).$$

As a consequence $\mathrm{Det}(t)$ is equal to $\det(S_1).\exp(t.\log(\det(S_2.S_1^{-1})))$ and the sign of $\frac{d}{dt}\mathrm{Det}(t)$ is constant and given by $\log(\det(S_2.S_1^{-1}))$. □

3.4.4 Criterion for the Equality of the Two Means

In general, Log-Euclidean and affine-invariant means are close, yet they are *not* identical. Nonetheless, there are a number of cases where they are identical, for example when the logarithms of averaged SPD matrices all commute with one another. In fact, we have more:

Proposition 3.7. *Let $(S_i)_{i=1}^N$ be N SPD matrices. If the Euclidean mean of the associated logarithms commutes with all* $\log(S_i)$*, then the Log-Euclidean and the affine-invariant means are identical.*

Proof. Let $\bar{L} := \frac{1}{N}\sum_{i=1}^N \log(S_i)$. The hypothesis is that $[\bar{L}, \log(S_i)] = 0,\ \forall i$. This implies that $\log(\exp(-\frac{1}{2}\bar{L}).S_i.\exp(-\frac{1}{2}\bar{L})) = \log(S_i) - \bar{L},\ \forall i$. We see then that $\exp \bar{L}$, i.e. the Log-Euclidean mean, is the solution of Eq. (3.11), i.e. is the affine-invariant mean. □

So far, we have not been able to prove the converse part of this proposition. However, the next subsection provides a partial proof, valid when SPD matrices are isotropic enough, i.e. close to a scaled version of the identity. The intensive numerical experiments we have carried out strongly suggest that the result given in the next section is true in general. The full proof of this assertion will be the subject of future work.

3.4.5 Larger Anisotropy in Log-Euclidean Means

In Section 3.4.6, we will verify experimentally that affine-invariant means tend to be less anisotropic then Log-Euclidean means. The following theorem accounts for this phenomenon when SPD matrices are isotropic enough.

Theorem 3.7. *Let $(S_i)_{i=1}^N$ be a finite number of SPD matrices close enough to the identity, so that we can apply the Baker-Campbell-Hausdorff formula in all cases (see Chapter 2, Section 2.3.6). When the logarithm of the Log-Euclidean mean does not commute with all* $\log(S_i)$*, then we have the following inequality:*

$$Trace(\mathbb{E}_{Aff}(S_1, ..., S_N)) < Trace(\mathbb{E}_{LE}(S_1, ..., S_N)). \tag{3.13}$$

Proof. The idea is to see how the two means differ close to the identity. To this end, we introduce a small scaling factor t and see how the two means vary when t is close to zero. For all i, let $S_{i,t}$ be the version of S_i scaled by t in the logarithmic domain. Around the identity, we can use the Baker-Campbell-Hausdorff (BCH) formula to simplify the barycentric equation (Eq. (3.11)). Let us denote in both Riemannian cases: $\mathbb{E}(S_t) = \mathbb{E}(S_{1,t}, ..., S_{N,t})$ and $\mathbb{E}(S) := \mathbb{E}(S_1, ..., S_N)$. We will also use the following notations: $\log(S_i) := L_i$, $\bar{L}_{t;Aff} := \log(\mathbb{E}_{Aff}(S_t))$ and $\bar{L}_{LE} := \log(\mathbb{E}_{LE}(S))$.

First, we use twice the BCH formula to obtain the following approximation:

$$\begin{aligned}\log(\mathbb{E}_{Aff}(S_t)^{-1/2}.S_{i,t}.\mathbb{E}_{Aff}(S_t)^{-1/2}) = \;& tL_i - \bar{L}_{t;Aff} - t^3\tfrac{1}{12}[L_i, [L_i, \bar{L}_{t;Aff}]] \\ & +t^3\tfrac{1}{24}[\bar{L}_{t;Aff}, [\bar{L}_{t;Aff}, L_i]] + O(t^5).\end{aligned} \tag{3.14}$$

Then we average over i to obtain the following approximation Lemma:

Lemma 3.1. *When t is small enough, we have:*

$$\bar{L}_{t;Aff} = t\bar{L}_{LE} + \frac{t^3}{12.N}\sum_{i=1}^{N}[L_i, [\bar{L}_{LE}, L_i]] + O(t^5). \tag{3.15}$$

Proof. To obtain the approximation, note that the second factor $t^3\frac{1}{24}[\bar{L}_{t;Aff}, [\bar{L}_{t;Aff}, L_i]]$ in Eq. (3.14) becomes a $O(t^5)$. Indeed, when the sum over i is done, L_i becomes $\bar{L}_{LE}$. But we can replace $\bar{L}_{LE}$ by its value in term of affine-invariance mean using Eq. (3.14). Then, using the fact that $[\bar{L}_{t;Aff}, \bar{L}_{t;Aff}] = 0$ we see that we obtain a $O(t^5)$.

Note also that thanks to the symmetry with respect to inversion, $\bar{L}_{t;Aff}$ becomes $-\bar{L}_{t;Aff}$ when t is changed into $-t$, i.e. $t \mapsto \bar{L}_{t;Aff}$ is odd. As a consequence, only odd terms appear in the development in powers of t. □

Next, we take the exponential of Eq. (3.15) and differentiate the exponential to obtain:

$$\mathbb{E}_{Aff}(S_t) = \mathbb{E}_{LE}(S_t) + D_{t\bar{L}_{LE}}\exp.\left(\frac{t^3}{12.N}\sum_{i=1}^{N}[L_i, [\bar{L}_{LE}, L_i]]\right) + O(t^5).$$

Then we use several properties to approximate the trace of affine-invariant means. First, we use Corollary 3.1 to simplify the use of the differential of the exponential. Then we approximate the exponential by the first two terms of its series expansion. We obtain:

$$\mathrm{Trace}(\mathbb{E}_{Aff}(S_t)) = \mathrm{Trace}(\mathbb{E}_{LE}(S_t)) + t^3.F(t, L_i, \bar{L}_{LE}) + O(t^5),$$

with $F(t, L_i, \bar{L}_{LE}) = \mathrm{Trace}\left(\exp(t\bar{L}_{LE}).\frac{1}{12.N}\sum_{i=1}^{N}[L_i, [\bar{L}_{LE}, L_i]]\right)$. This expression can be simplified:

$$\begin{aligned}F(t, L_i, \bar{L}_{LE}) &= \mathrm{Trace}\left((\mathrm{Id} + t\bar{L}_{LE}).\frac{1}{12.N}\sum_{i=1}^{N}[L_i, [\bar{L}_{LE}, L_i]]\right) + O(t^2) \\ &= \frac{t}{12.N}\sum_{i=1}^{N}\mathrm{Trace}\left(\bar{L}_{LE}.[L_i, [\bar{L}_{LE}, L_i]]\right) + O(t^2) \\ &= -\frac{t}{12.N}\sum_{i=1}^{N}\mathrm{Trace}\left(L_i^2.\bar{L}_{LE}^2 - (L_i.\bar{L}_{LE})^2\right) + O(t^2).\end{aligned}$$

As a consequence, the difference between the two traces can be written:

$$\mathrm{Trace}(\mathbb{E}_{Aff}(S_t)) - \mathrm{Trace}(\mathbb{E}_{LE}(S_t)) = -\frac{t^4}{12.N}\sum_{i=1}^{N} \mathrm{Trace}\left(L_i^2.\bar{L}_{LE}^2 - (L_i.\bar{L}_{LE})^2\right) + O(t^5).$$

To conclude, we use the following Lemma:

Lemma 3.2. *Let A, $B \in Sym(n)$. Then: $Trace(A^2.B^2 - (A.B)^2) \geq 0$. The inequality is strict if and only if A and B do not commute.*

Proof. Let (A_i) (resp. (B_i)) be the column vectors of A (resp. B). Let $<,>$ be the usual scalar product. Then we have:

$$\begin{cases} \mathrm{Trace}(A^2.B^2) = \sum_{i,j} < A_i, A_j >< B_i, B_j > \\ \mathrm{Trace}((A.B)^2) = \sum_{i,j} < A_i, B_j >< B_i, A_j > . \end{cases}$$

Let us now chose a rotation matrix R that makes A diagonal: $R.A.R^T = Diag(\lambda_1, ..., \lambda_n) =: D$. Let us define $C := R.B.R^T$ and use the notations (C_i) and (D_i) for the column vectors of C and D. We have:

$$\begin{cases} \mathrm{Trace}(A^2.B^2) = \sum_{i,j} < D_i, D_j >< C_i, C_j >= \sum_i \lambda_i^2 < C_i, C_i > \\ \mathrm{Trace}((A.B)^2) = \sum_{i,j} < D_i, C_j >< C_i, D_j >= \sum_{i,j} \lambda_i.\lambda_j < C_i, C_j > . \end{cases}$$

Then the Cauchy-Schwarz inequality yields:

$$|\sum_{i,j} \lambda_i.\lambda_j < C_i, C_j > | \leq \sum_i \lambda_i^2 < C_i, C_i >,$$

which proves the first point. But the Cauchy-Schwarz inequality is an equality if and only if there is a constant μ such that $D.C = \mu C.D$. But only $\mu = 1$ allows the inequality of the lemma to be an equality. This is equivalent to $C.D = D.C$, which is equivalent in turn to $A.B = B.A$. Hence the result. □

End of Proof of Theorem 3.7 When we apply Lemma 3.2 to the obtained estimation for the trace, we see that for a $t \neq 0$ small enough, the trace of the affine-invariant mean is indeed strictly inferior to the trace of the Log-Euclidean mean whenever the mean logarithm does not commute with all logarithms $\log(S_i)$. □

Corollary 3.7. *By invariance of the two means with respect to scaling, the strict inequality given in Theorem 3.7 is valid in a neighborhood of any SPD matrix of the form λId with $\lambda > 0$.*

Corollary 3.8. *When the dimension is equal to 2, the Log-Euclidean mean of SPD matrices which are isotropic enough is strictly more anisotropic than their affine-invariant mean when those means do not coincide.*

Proof. In this case, there are only two eigenvalues for each mean. Their products are equal and we have a strict inequality between their sum. Consequently, the largest eigenvalue of the Log-Euclidean mean is strictly larger than the affine-invariant one, and we have the opposite result for the smallest eigenvalue. □

3.4.6 Linear and Bilinear Interpolation of SPD Matrices

Volume elements (or *voxels*) in clinical DT images are often spatially anisotropic. Yet, in many practical situations where DT images are used, it is recommended (see [Basser 00]) to work with isotropic voxels to avoid spatial biases. A preliminary resampling step with an adequate interpolation method is therefore important in many cases. Proper interpolation methods are also required to generalize to the SPD case usual registration techniques used on scalar or vector images. The framework of Riemannian metrics allows a direct generalization to SPD matrices of classical resampling methods with the use of associated Fréchet means instead of the Euclidean (i.e. arithmetic) mean.

In the Riemannian case, the equivalent of linear interpolation is *geodesic interpolation*. To interpolate between two SPD matrices, intermediate values are taken along the shortest path joining the two matrices. Fig 3.1 presents a typical result of linear interpolation between two SPD matrices. The Euclidean, affine-invariant and Log-Euclidean results are given. The 'swelling effect' is clearly visible in the Euclidean case: the volume of associated ellipsoids is parabolically interpolated and reaches a global maximum between the two extremities! This effect disappears in both Riemannian cases, where volumes are interpolated geometrically. As expected, Log-Euclidean means are a little more anisotropic than their affine-invariant counterpart.

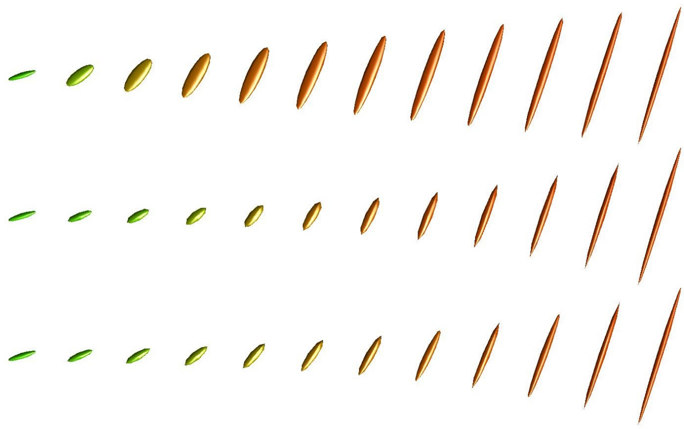

Figure 3.1: **Linear interpolation of two SPD matrices**. **Above**: linear interpolation on coefficients. **Middle**: affine-invariant interpolation. **Below**: Log-Euclidean interpolation. The coloring of ellipsoids is based on the direction of dominant eigenvectors. Note the characteristic swelling effect observed in the Euclidean case, which is not present in both Riemannian frameworks. Note also that Log-Euclidean means are slightly more anisotropic their affine-invariant counterparts.

To resample images, bi-(resp. tri-)linear interpolation generalizes in 2D (resp. in 3D) the linear interpolation and offers an efficient compromise between simplicity and accuracy in the scalar and vector cases. With this technique, the value at any given point is inferred from known values measured at the vertices of a regular grid whose elementary cells are rectangles in 2D (resp. right parallelepipeds in 3D), which is usually the case with MR images. More precisely, the interpolated value at a given point is given by the weighted mean of the values at the vertices of the current cell. The weights are the *barycentric coordinates* of the current point with respect to the vertices of the current cell.

Fig. 3.2 presents the results of the bilinear interpolation of four SPD matrices placed at the extremities of a rectangle. Again, a large 'swelling effect' is present in Euclidean results and not in both Riemannian results, and Log-Euclidean means are slightly more anisotropic than their affine-invariant equivalent. One should note that the computation of the affine-invariant mean here is iterative, since the number of averaged matrices is larger than 2 (we use the Gauss-Newton method described in [Pennec 06b]), whereas the closed form given by Eq. 3.10 is used directly in the Log-Euclidean case. This has a large impact on computation times: $0.003s$ (Euclidean), $0.009s$ (Log-Euclidean) and $1s$ (affine-invariant) for a 5×5 grid on a Pentium M 2 GHz. Computations were carried out with Matlab™, which explains the poor computational performance. Here, Log-Euclidean means were calculated approximately 100 times faster than affine-invariant means because the logarithms of the four interpolated tensors were only computed once, instead of computing them each time a new barycenter is calculated. When only one mean is computed, the typical ratio is closer to 20, since between 15 and 20 iterations are typically needed (for 3×3 SPD matrices) to obtain the affine-invariant mean with a precision of the order of 10^{-12}.

One should note that from a numerical point of view the computation of Log-Euclidean means is not only much faster but also more *stable* than in the affine-invariant case. On synthetic examples, as soon as SPD matrices are quite anisotropic (for instance with the dominant eigenvalue larger than 500 times the smallest), numerical instabilities appear, essentially due to limited numerical precision (even with double precision). This can complicate greatly the computation of affine-invariant means. On the contrary, the computation of Log-Euclidean means is more stable since the logarithm and exponential are taken only once and thus even very large anisotropies can be dealt with. In applications where very high anisotropies are present, such as the generation of adapted meshes [Mohammadi 97], this phenomenon could severely limit the use of affine-invariant means, whereas no such limitation exists in the Log-Euclidean case.

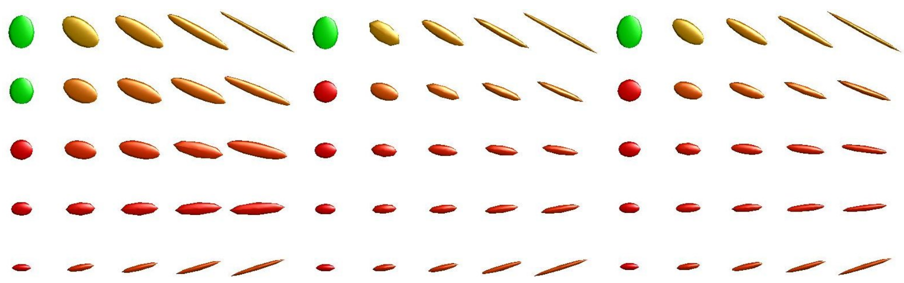

Figure 3.2: **Bilinear interpolation of 4 SPD matrices at the corners of a regular grid**. **Left**: Euclidean interpolation. **Middle**: affine-invariant interpolation. **Right**: Log-Euclidean interpolation. Again, a characteristic swelling effect is observed in the Euclidean case and not in both Riemannian frameworks. As expected, Log-Euclidean means are slightly more anisotropic than their affine-invariant counterparts.

3.5 Probabilities and Statistics with Log-Euclidean Metrics

In this Section, we present the Riemannian statistical framework for the tensor space endowed with Log-Euclidean metric. It is particularly simple, since it is the same as the usual Euclidean framework when one identifies a tensor to its logarithm.

Practically, one simply uses the usual tools of Euclidean statistics on the logarithms and maps the results back to the Tensor Vector Space with the exponential. We recall that this is theoretically fully justified because the tensor space endowed with a Log-Euclidean metric and our novel vector space structure is isomorphic, diffeomorphic and isometric to the vector space of symmetric matrices with the associated[2] Euclidean norm.

Applications of statistics on tensors include for example DT-MRI segmentation [Lenglet 06a], the injection of statistical priors in medical image registration [Commowick 05, Pennec 05], tensor based morphometry [Lepore 06], computing mean diffusion tensor images of organs such as the heart [Peyrat 06b] or the brain [Goodlett 06]. In the four latter references, statistics are computed in our Log-Euclidean framework, which has already begun to be used on a regular based by several teams.

3.5.1 General Riemannian Statistical Framework

The tensor space with a Log-Euclidean metric is a Riemannian space, exactly as the tensor space with an affine-invariant metric. In [Pennec 06a], the statistical framework in Riemannian spaces is fully presented from a geometrical point of view.

In these spaces, one usually generalizes the classical expectation of a random-variable with the notion of Fréchet expectation. It is defined as the set of points which minimize the metric dispersion of the random variable. Let $(G, d(.,.))$ be a Riemannian space with its distance and $S : \Omega \to G$ a G-valued random variable. Let dP be the probability measure associated to S defined on the space of all possible outcomes Ω (see [Billingsley 95] for a complete description of the classical probabilistic framework, particularly for technical requirements such as measurability, which will not be mentioned here).

With these notations, the Fréchet mean $\mathbb{E}(S)$ of S is defined by:

$$\mathbb{E}(S) = \arg\min_{T} \int_{\Omega} d^2(T, S(\omega)) dP(\omega). \tag{3.16}$$

A priori, existence and uniqueness are only guaranteed when the values taken by S are contained in a region of G that is small enough. See [Pennec 06a] page 13 for the statement of Karcher's theorem.

When the Fréchet expectation is uniquely defined, one can also compute centered moments of superior order like the covariance. This is done this time using vectors, namely the logarithms centered on the mean. More precisely, the covariance matrix (see [Pennec 06a] page 17) is defined by:

$$Cov_{\mathbb{E}(S)}(S) = \mathbb{E}(\log_{\mathbb{E}(S)}(S). \log_{\mathbb{E}(S)}(S)^T) = \int_{\Omega} \log_{\mathbb{E}(S)}(S(\omega)). \log_{\mathbb{E}(S)}(S(\omega))^T dP(\omega). \tag{3.17}$$

Within this framework, many usual statistical tools can be used, like Mahalanobis distance, generalizations of the normal law, etc. See [Pennec 06a] for more details.

3.5.2 Random Tensors

Thanks to the isometric isomorphism between the tensor space with a Log-Euclidean metric and the Euclidean vector space of symmetric matrices, the theory of tensor-valued random variables is greatly simplified. Every notion of probabilities and statistics on vectors is readily generalized in the tensor case.

Indeed, one can define the classical vector spaces of random tensors, e.g., the classical L^α Banach spaces of measurable tensor-valued functions:

[2] By *associated*, we mean that the metric on the space of symmetric is the same as that used on the space of symmetric matrices viewed as the Lie algebra of our Lie group structure for tensors.

Definition 3.5. *We can define for $\beta \geq 1$ the Banach vector space $(L^\beta(\Omega, Sym_\star^+(n)), \odot, \circledast, \|.\|_\beta)$ of L^β-integrable tensor-valued random variables by identification with the vector space of symmetric matrices-valued random variables with the same integrability requirement. $S \in L^\beta(\Omega, Sym_\star^+(n))$ if and only if:*

$$\int_\Omega \| \log(S(\omega))\|^\beta dP(\omega) < \infty. \tag{3.18}$$

The associated norm $\|S\|_\beta$ is simply:

$$\|S\|_\beta = \left(\int_\Omega \| \log(S(\omega))\|^\beta dP(\omega) \right)^{\frac{1}{\beta}}. \tag{3.19}$$

One can also compute characteristic functions in the same manner, etc.

3.5.3 Fréchet Means and Covariances with Log-Euclidean Metrics

Computing means and expectations is particularly simple with Log-Euclidean metrics, much more than in the general Riemannian case. Indeed, we have:

For any L^2 tensor-valued random variable S, its Log-Euclidean Fréchet mean $\mathbb{E}_{\mathrm{LE}}(S)$, also called *Log-Euclidean mean*, is defined and uniquely so. It is given as in the Euclidean case by:

Theorem 3.8. *Let $<,>$ be a Log-Euclidean metric on the tensor space. Let S be a L^2 tensor-valued random variable. Then its Fréchet mean is well defined and we have:*

$$\mathbb{E}_{LE}(S) = \exp\left(\int_\Omega \log(S(\omega)) dP(\omega) \right). \tag{3.20}$$

In particular, The Log-Euclidean mean of N tensors is given by:

$$\mathbb{E}_{LE}(S_1, ..., S_N) = \exp\left(\frac{1}{N} \sum_i \log(S_i) \right). \tag{3.21}$$

In the case of L^1 random tensors, one can no longer define the Fréchet mean as usual, since the L^2 metric dispersion used to this effect is not well-defined (it is only so for L^2 random tensors!). But in this very general case, like in the vector case, one can generalize the Fréchet expectation by defining the expectation directly from Eq. (3.20).

Proof. When one expresses everything in the logarithm domain, one is faced with the classical computation of expectations and means in a Euclidean vector space. Hence the result by mapping back the results with exp in the tensor domain. □

This theorem shows in a simple way that it is not necessary to suppose that the random tensor hits almost surely a small enough region of the tensor space. The usual integrability condition on vectors apply.

Exponential-Invariance. As announced in Section 3.3, there is a last invariance property associated to Log-Euclidean metrics:

Proposition 3.8. *The Log-Euclidean mean in the tensor space is exponential-invariant. By this, we mean that if a scaling is applied in the logarithmic domain to a random tensor, then the resulting mean is scaled identically with respect to the Log-Euclidean mean of the same tensor-valued random variable. For example, the mean of the square root of a random tensor is the square root of its mean.*

Proof. This simply results from the factorization of a scalar factor for in the expectation of random vector. □

Actually, this last invariance property is quite remarkable. In [Alauzet 03], in the context of the generation of adapted meshes via metric tensors, it was proposed to compute mean tensors with the following strategy:

1. Transform the data (S_i) into (S_i^β), with $\beta > 0$, for example $\beta = \frac{1}{2}$
2. Compute the arithmetic mean $\sum_i w_i S_i^\beta$ of the (S_i^β)
3. Obtain a mean tensor via $\left(\sum_i w_i S_i^\beta\right)^{\frac{1}{\beta}}$

This points a very interesting fact: we can in fact process tensors via their square roots or their squared versions, or any of their powers. The results can be obtained from processed data by taking the inverse power in a final step.

In DT-MRI (see Chapter 4 for more details on this imaging technique), is it really more justified to process tensors than their square roots? We will see in Chapter 4 than the square roots of diffusion tensors have a physical meaning: they give directly the geometry of the statistical *confidence regions* associated to the local Brownian diffusion process of water molecules. Therefore, it would be very interesting to have processing algorithms which are *invariant* with respect to the taking of square roots.

The exponential-invariance of the Log-Euclidean mean precisely guarantees that the Log-Euclidean mean is invariant with respect to the taking of square roots and more generally of any power. Neither the Euclidean mean nor of the affine-invariant mean are exponential-invariant or invariant to the taking of square roots.

Log-Euclidean Covariance. Like for the mean, there are considerable simplifications in the computations of other moments. Their form is exactly the usual one. For covariances, we get:

Proposition 3.9. *Let us endow the tensor space with a Log-Euclidean metric. Let $S : \Omega \to Sym_\star^+(n)$ be a tensor-valued random variable. Then the associated covariance matrix $Cov(S)$ is:*

$$\begin{aligned} Cov(S) &= \mathbb{E}(\log_{\mathbb{E}_{LE}(S)}(S). \log_{\mathbb{E}_{LE}(S)}(S)^T) \\ &= \int_\Omega (\log(S(\omega)) - \log(\mathbb{E}_{LE}(S))).(\log(S(\omega)) - \log(\mathbb{E}_{LE}(S))^T) dP(\omega). \end{aligned} \tag{3.22}$$

Proof. This is done by replacing $\log_{\mathbb{E}_{\mathrm{LE}}(S)}(S(\omega))$ by its value and taking into account the scalar product at the Fréchet mean from Eqs. (3.6) and (3.7). The differential of the logarithm is canceled out by the differential of the exponential like in the computation of distances. □

3.5.4 General Log-Euclidean Statistical Framework

All probabilistic and statistical notions of the Euclidean framework are directly transposed by the logarithm in the Log-Euclidean framework. Hence, Kolmogorov's Strong Law of Large Number applies for Log-Euclidean means, and so does the Central Limit theorem and all the others vectorial theorems in statistics. One can for example use Principal Component Analysis to analyze data, use Gaussian distributions, etc. This yields a much simpler framework than in [Fletcher 04a,Lenglet 06a] where part of the affine-invariant framework statistics for tensors is presented. One can proceed in this domain exactly like for vector-valued random variables. As examples, we formulate the Strong Law of Large Numbers for tensors and generalize the normal law to tensors. Other generalizations are straightforward and left to the reader.

Kolomogorov's Strong Law of Large Numbers for Tensors. *Let (S_n) be a sequence of independent identically distributed L^1 tensor-valued random variables. Then in the L^1 sense and almost surely for $\omega \in \Omega$:*

$$\frac{1}{N}\sum_{k=1}^{N} \log(S_i)(\omega) \underset{N\to\infty}{\longrightarrow} \mathbb{E}(\log(S)). \tag{3.23}$$

which by continuity of exp *yields with the Log-Euclidean means:*

$$\mathbb{E}_{LE}\left((S_i)_{i=1..N}(\omega)\right) \underset{N\to\infty}{\longrightarrow} \mathbb{E}_{LE}(S) \textit{ almost surely and in the } L^1 \textit{ sense.} \tag{3.24}$$

Proof. As mentioned in the theorem, only the continuity of the exponential is necessary to extend the vector case to the tensor case. For a vectorial proof, see [Billingsley 95]. □

Definition 3.6. *A random tensor T is said to follow a normal law of mean S_0 and covariance Σ if and only if* $\log(S)$ *follows a classical normal law of mean* $\log(S_0)$ *and covariance* Σ.

3.6 Conclusions

In this work, we have presented a particularly simple and efficient Riemannian processing framework for tensors, called *Log-Euclidean*. The associated Fréchet mean, called the *Log-Euclidean mean*, is a generalization of the geometric mean to SPD matrices. It is simply an arithmetic mean in the domain of matrix logarithms. This mean corresponds to a bi-invariant mean in a novel structure of Lie group on SPD matrices, or equivalently to a Euclidean mean when this structure is smoothly extended into a vector space by a novel scalar multiplication.

The Log-Euclidean mean is similar to the recently introduced affine-invariant mean, which is another generalization of the geometric mean to SPD matrices. Indeed, the Log-Euclidean mean is similarity-invariant and the two means have the same determinant, which is the geometric mean of the determinants of averaged SPD matrices. However, they are not equal: the Log-Euclidean trace is larger when the two means differ. The most striking difference between the two means resides in their computational cost: the Log-Euclidean mean can be calculated approximately 20 times faster than the affine-invariant mean. This property can be crucial in applications where large amounts of data are processed. This is especially the case in medical imaging with diffusion tensor imaging and in numerical analysis with the generation of adapted meshes.

Furthermore, we have presented in this Chapter the general Log-Euclidean statistical framework for tensors. It is particularly simple to use, since Log-Euclidean Riemannian statistics are simply Euclidean statistics on the logarithms of tensors. Our framework is for example used in [Peyrat 06b, Peyrat 06a] to study the statistical variability of the diffusion tensors of canine hearts and in [Lepore 06] to perform statistical tests on deformation tensors steming from a registration procedure to compare morphologically two groups of individuals (HIV/AIDS patients vs. normals).

Chapter 4

Log-Euclidean Processing of Diffusion Tensors

In this Chapter, we focus on the application of the Log-Euclidean framework for tensors to the processing of a specific type of tensors: *diffusion tensors*. Indeed, this specific type of tensor is particularly relevant in the medical imaging community, since diffusion tensor imaging is an emerging imaging modality whose importance has been growing considerably. In particular, most attempts to reconstruct non-invasively the connectivity of the brain are based on DTI.

This Chapter was entirely published in the international journal Magnetic Resonance in Medicine [Arsigny 06f]. In this publication, we wrote a condensed and intuitive presentation of the theory of Log-Euclidean metrics on tensors, detailed in the previous Chapter. One can still find it in Section 4.2. This short presentation should help the reader assimilate the results obtained from (quite) a mathematical perspective in Chapter 3.

Contents

4.1 Introduction **54**
4.1.1 The Defects of Euclidean Calculus 55
4.1.2 Riemannian Metrics 55
4.2 Theory **56**
4.2.1 Matrix Exponential, Logarithm and Powers 56
4.2.2 Definition of Log-Euclidean Metrics 57
4.2.3 Invariance Properties of Log-Euclidean Metrics 58
4.2.4 Log-Euclidean Computations on Tensors 58
4.2.5 Comparison of the Affine-Invariant and Log-Euclidean Frameworks . 59
4.3 Methods **60**
4.3.1 Interpolation 60
4.3.2 Regularization 60
4.3.3 Absolute Value of a Symmetric Matrix 61
4.3.4 Materials 61
4.4 Results **62**
4.4.1 Interpolation 62
4.4.2 Regularization 64

4.5 **Discussion and Conclusions** **69**
4.5.1 The Defects of Euclidean Calculus 69
4.5.2 Conclusions and Perspectives 70

Abstract. Diffusion tensor imaging is an emerging imaging modality whose importance has been growing considerably. However, the processing of this type of data (i.e. symmetric positive-definite matrices), called 'tensors' here, has proved difficult in recent years. Usual Euclidean operations on matrices suffer from many defects on tensors, which have led to the use of many *ad hoc* methods. Recently, affine-invariant Riemannian metrics have been proposed as a rigorous and general framework in which these defects are corrected. These metrics have excellent theoretical properties and provide powerful processing tools, but also lead in practice to complex and slow algorithms. To remedy this limitation, a new family of Riemannian metrics called Log-Euclidean is proposed in this Chapter. They also have excellent theoretical properties and yield similar results in practice, but with much simpler and faster computations. This new approach is based on a novel vector space structure for tensors. In this framework, Riemannian computations can be converted into Euclidean ones once tensors have been transformed into their matrix logarithms. Theoretical aspects are presented and the Euclidean, affine-invariant and Log-Euclidean frameworks are compared experimentally. The comparison is carried out on interpolation and regularization tasks on synthetic and clinical 3D DTI data.

4.1 Introduction

Diffusion tensor imaging (DT-MRI or DTI or equivalently DT imaging) [Basser 94] is an emerging imaging modality whose importance has been growing considerably. In particular, most attempts to reconstruct non-invasively the connectivity of the brain are based on DTI (see [Mori 02, Lenglet 04, Fillard 03, Vemuri 01, Basser 00, Poupon 00] and references within for classical fiber tracking algorithms). Other applications of DT-MRI also include the study of diseases such as stroke, multiple sclerosis, dyslexia and schizophrenia [Le Bihan 01].

The diffusion tensor is a simple and powerful model used to analyze the content of Diffusion-Weighted images (DW-MRIs). It is based on the assumption that the motion of water molecules can be well approximated by a Brownian motion in each voxel of the image. This Brownian motion is entirely characterized by a symmetric and positive-definite matrix, called the 'diffusion tensor' [Basser 94]. In this Chapter, we restrict the term 'tensor' to mean a symmetric and positive-definite matrix.

As we mentionned in the previous Chapter, with the increasing use of DT-MRI and other types of tensors, there has been a growing need to generalize to the tensor case many usual vector processing tools. In particular, regularization techniques are required to denoise them. Furthermore, classical tasks like interpolation also need to be generalized to resample DT images, for example to work with isotropic voxels, as recommended in [Basser 00]. It would also be very valuable to generalize to tensors classical vector statistical tools, in order to analyze the variability of tensors or model the noise that corrupts them. Previous attempts to do so are only partially satisfactory: for example, it was proposed in [Basser 03] to define a Gaussian distribution on tensors as a Gaussian distribution on symmetric matrices, without taking into account the positive-definiteness constraint. This becomes problematic with Gaussians whose covariance is large: in this case, non-positive eigenvalues do appear with a significant probability.

Many *ad hoc* approaches have already been proposed in the literature to process tensors (see [Westin 02, Chefd'hotel 04] and references within). But in order to fully generalize to tensors the usual PDEs or statistical tools used on scalars or vectors, one needs to define a consistent operational framework. The framework of Riemannian metrics [Pennec 99,Gallot 93] has recently emerged as particularly adapted to this task [Pennec 06b,Batchelor 05, Lenglet 06a, Fletcher 04a].

4.1.1 The Defects of Euclidean Calculus

The simplest Riemannian structures are the Euclidean ones, and we detail here the defects of Euclidean calculus in the specific context of DT-MRI.

Let S_1 and S_2 be two tensors. An example of Euclidean structure is given by the so-called 'Frobenius distance': $\mathrm{dist}^2(S_1, S_2) = (\mathrm{Trace}((S_1 - S_2)^2))$. This straightforward metric leads *a priori* to simple computations. Unfortunately, though Euclidean distances are well-adapted to general square matrices, they are unsatisfactory for tensors, which are very specific matrices. Typically, symmetric matrices with null or negative eigenvalues appear on clinical data as soon as we perform on tensors Euclidean operations which are non-convex. Example of such situations are the estimation of tensors from diffusion-weighted images, the regularization of tensors fields, etc. The noise in the data is at the source of this problem. To avoid obtaining non-positive eigenvalues, which are difficult to interpret physically, it has been proposed to regularize only *features* extracted from tensors, like first eigenvectors [Coulon 04] or orientations [Chefd'hotel 04]. This is only partly satisfactory, since such approaches do not take into account *all* the information carried by tensors.

After a diffusion time Δ, we know with a confidence say of 95% that a water molecule is located within a region called a *confidence region*, which is the multidimensional equivalent of a confidence interval. The larger the volume of these regions, the larger is the dispersion of the random displacement of water molecules. In the case of Brownian motion, the random displacement is Gaussian, and confidence regions are therefore ellipsoids. The volumes of these ellipsoids are proportional to the square root of the *determinant* of the covariance matrix of the displacement. In DT-MRI, this covariance matrix is equal to the diffusion tensor multiplied by 2Δ [Basser 94]. The value of the determinant of the diffusion tensor is therefore a direct measure of the dispersion of the local diffusion process. But the Euclidean averaging of tensors generally leads to a *tensor swelling effect* [Feddern 04, Chefd'hotel 04, Tschumperlé 01]: the determinant (and thus the dispersion) of the Euclidean mean of tensors can be larger than the determinants of the original tensors! Introducing more dispersion in computations amounts to introducing more diffusion, which is physically unrealistic.

4.1.2 Riemannian Metrics

To fully circumvent these difficulties, affine-invariant Riemannian metrics have been recently proposed for tensors by several teams, as pointed out in the previous Chapter. The application of these metrics to the averaging of tensors and the definition of a Riemannian anisotropy measure were presented [Moakher 05, Batchelor 05]. The generalization of principal component analysis (PCA) to tensors was given in [Fletcher 04a]. The affine-invariant statistical framework and its application to the segmentation of DT-MRI was presented in [Lenglet 06a]. PDEs within the affine-invariant framework were studied in [Pennec 06b] with applications to the interpolation, extrapolation and regularization of tensor fields.

With affine-invariant metrics, symmetric matrices with negative and null eigenvalues are at an infinite distance from any tensor and the swelling effect disappears. Practically, this prevents the appearance of non-positive eigenvalues, which is particularly difficult to avoid in Euclidean algorithms. But the price paid for this success is a high computational burden, essentially due to the curvature induced on the tensor space. This substantial computational cost can be seen directly from the formula giving the distance between two tensors S_1 and S_2 [Pennec 06b]:

$$\mathrm{dist}(S_1, S_2) = \left\| \log\left(S_1^{-\frac{1}{2}}.S_2.S_1^{-\frac{1}{2}} \right) \right\|, \tag{4.1}$$

where $\|.\|$ is a Euclidean norm on symmetric matrices. In general, affine-invariant computations involve an intensive use of matrix inverses, square roots, logarithms and exponentials.

We present in this Chapter a new Riemannian framework to fully overcome these computational limitations while preserving excellent theoretical properties. Moreover, we obtain this result without any unnecessary complexity, since all computations on tensors are converted into computations on vectors. This framework is based on a new family of metrics named *Log-Euclidean*, which are particularly simple to use. They result in classical Euclidean computations in the domain of matrix logarithms. In the next section, we present the theory of Log-Euclidean metrics (more details on this theory can be found in a research report, see [Arsigny 05b]). In the Methods section, we describe the adaptation of classical processing tools to the Log-Euclidean framework for interpolation and regularization tasks. We also present a highly useful tool for the visualization of difference between tensors: the *absolute value* of a symmetric matrix. Then, we show that the affine-invariant and Log-Euclidean frameworks perform better than the Euclidean one for the interpolation and regularization of our synthetic and clinical 3D DT-MRI data. Affine-invariant and Log-Euclidean results are very similar, but computations are simpler and experimentally much faster in the Log-Euclidean than in the affine-invariant framework.

4.2 Theory

In this Section, we present briefly the theoretical results obtained in Chapter 3, this time from an intuitive and practical point of view.

4.2.1 Matrix Exponential, Logarithm and Powers

The notions of matrix logarithm and exponential are central in the theoretical framework presented here. For any matrix M, its exponential is given by: $\exp(M) = \sum_{k=0}^{\infty} M^k/k!$. As in the scalar case, the *matrix logarithm* is defined as the inverse of the exponential. One should note that for general matrices, neither the uniqueness nor the existence of a logarithm is guaranteed for a given invertible matrix [Culver 66, Bourbaki 89]. However, the important point here is that the logarithm of a tensor is well-defined and is a symmetric matrix. Conversely, the exponential of any symmetric matrix yields a tensor. This means that under the matrix exponentiation operation, there is a one-to-one correspondence between symmetric matrices and tensors.

This one-to-one correspondence can be seen quite intuitively thanks to the simple spectral decomposition of these matrices. Indeed, the matrix logarithm L of a tensor S can be calculated in three steps:

1. perform a diagonalization of S, which provides a rotation matrix R and a diagonal matrix D with the eigenvalues of S in its diagonal, with the equality: $S = R^T.D.R$.

2. transform each diagonal element of D (which is necessarily positive, since it is an eigenvalue of S) into its natural logarithm in order to obtain a new diagonal matrix $\tilde{D}$.

3. recompose $\tilde{D}$ and R to obtain the logarithm with the formula $L = \log(S) = R^T.\tilde{D}.R$.

Conversely, the matrix exponential S is obtained by replacing the natural logarithm with the scalar exponential. One also generalizes of the notion of powers (and in particular square roots) to tensors by replacing their eigenvalues by the corresponding scalar power (for example by their square roots).

4.2.2 Definition of Log-Euclidean Metrics

Based on the specific properties of the matrix exponential and logarithm on tensors that we presented above, we can now define a novel *vector space* structure on tensors. This is quite a surprising result: in the sense of this new algebraic structure, *tensors* can be also looked upon as *vectors*! As will be shown in the rest of this Chapter, this novel viewpoint provides a particularly powerful and simple-to-use framework to process tensors.

Since there is a one-to-one mapping between the tensor space and the *vector* space of symmetric matrices, one can transfer to tensors the standard algebraic operations (addition "+" and scalar multiplication ".") with the matrix exponential. This defines on tensors a *logarithmic multiplication* $\odot$ and a *logarithmic scalar multiplication* $\circledast$, given by:

$$\begin{cases} S_1 \odot S_2 \stackrel{def}{=} \exp\left(\log(S_1) + \log(S_2)\right) \\ \lambda \circledast S \stackrel{def}{=} \exp\left(\lambda . \log(S)\right) = S^{\lambda}. \end{cases}$$

The logarithmic multiplication is *commutative* and coincides with matrix multiplication whenever the two tensors S_1 and S_2 commute in the matrix sense. With $\odot$ and $\circledast$, the tensor space has by construction a *vector space structure*, which is *not* the usual structure directly derived from addition and scalar multiplication on matrices.

When one considers only the multiplication $\odot$ on the tensor space, one has a *Lie group* structure [Gallot 93], i.e. a space which is both a smooth manifold and a group in which multiplication and inversion are *smooth mappings*. This type of mathematical tool is for example particularly useful in theoretical physics [Tarantola 06]. Here, the smoothness of $\odot$ comes from the fact that both the exponential and the logarithm mappings are smooth [Arsigny 05b]. Among Riemannian metrics in Lie groups, the most convenient in practice are *bi-invariant metrics*, i.e. metrics that are invariant by multiplication and inversion. When they exist, these metrics are used in differential geometry to generalize to Lie groups a notion of mean which is completely consistent with multiplication and inversion. This approach applies particularly well in the case of the group of rotations [Pennec 96, Pennec 98a, Moakher 02]. However, such metrics do not always exist, as in the case of the groups of Euclidean motions [Woods 03, Pennec 06a] and affine transformations. It is remarkable that bi-invariant metrics exist in our tensor Lie group. Moreover, they are particularly simple. Their existence simply results from the *commutativity* of logarithmic multiplication between tensors. We have named such metrics *Log-Euclidean* metrics, since they correspond to Euclidean metrics in the domain of logarithms. From a Euclidean norm $\|.\|$ on symmetric matrices, they can be written:

$$\mathrm{dist}(S_1, S_2) = \|\log(S_1) - \log(S_2)\|. \tag{4.2}$$

From Eq. (4.2), it is clear that Log-Euclidean metrics are also Euclidean distances for the vector space structure we defined earlier. We did not define them directly from the latter algebraic structure to emphasize the fact that they are also *Riemannian metrics*, like affine-invariant metrics.

As one can see, the Log-Euclidean distance is much simpler than the equivalent affine-invariant distance given by Eq. (4.1), where matrix multiplications, square roots and inverses are used before taking the norm of the logarithm. The greater simplicity of Log-Euclidean metrics can also be seen from Log-Euclidean geodesics in the tensor space. In the Log-Euclidean case, the shortest path $\gamma_{\mathrm{LE}}(t)$ going from the tensor S_1 at time 0 to the tensor S_2 at time 1 is a straight line in the domain of logarithms. This geodesic is given by:

$$\gamma_{\mathrm{LE}}(t) = \exp\left((1-t)\log(S_1) + t\log(S_2)\right).$$

Its affine-invariant equivalent $\gamma_{\mathrm{Aff}}(t)$ involves the use of square roots and inverses and takes the following form:

$$\gamma_{\mathrm{Aff}}(t) = {S_1}^{\frac{1}{2}}.\exp\left(t\log\left({S_1}^{-\frac{1}{2}}.S_2.{S_1}^{-\frac{1}{2}}\right)\right).{S_1}^{\frac{1}{2}}.$$

Contrary to the classical Euclidean framework on tensors, one can see from Eq. (4.2) that symmetric matrices with null or negative eigenvalues are at an infinite distance from any tensor and therefore will not appear in practical computations. The same property holds for affine-invariant metrics [Pennec 06b].

4.2.3 Invariance Properties of Log-Euclidean Metrics

Log-Euclidean metrics satisfy a number of *invariance properties*, i.e. are left unchanged by several operations on tensors. First, distances are not changed by inversion, since taking the inverse of a system of matrices only results in the multiplication by -1 of their logarithms, which does not change the value of the distance given by Eq. (4.2). Also, Log-Euclidean metrics are by construction invariant with respect to any logarithmic multiplication, i.e. are invariant by any translation in the domain of logarithms. However, there is more. Although Log-Euclidean metrics do not yield full affine-invariance as the affine-invariant metrics defined in [Pennec 06b], a number of them are invariant by *similarity* (orthogonal transformation and scaling) [Arsigny 05b]. This means that computations on tensors using these metrics will be invariant with respect to a change of coordinates obtained by a similarity. In this work, we use the simplest similarity-invariant Log-Euclidean metric, which is given by:

$$\mathrm{dist}(S_1, S_2) = \left(\mathrm{Trace}\left(\{\log(S_1) - \log(S_2)\}^2\right)\right)^{\frac{1}{2}}.$$

4.2.4 Log-Euclidean Computations on Tensors

From a practical point of view, one would like operations such as averaging, filtering, etc. to be as simple as possible. In the affine-invariant case, such operations rely on an intensive use of matrix exponentials, logarithms, inverses and square roots. In our case, the space of tensors with a Log-Euclidean metric is in fact *isomorphic* (the algebraic structure of vector space is conserved) and *isometric* (distances are conserved) with the corresponding Euclidean space of symmetric matrices. As a consequence, the Riemannian framework for statistics and analysis is extremely simplified. To illustrate this, let us recall the notion of *Fréchet mean* [Pennec 99, Jones 02], which is the Riemannian equivalent of the Euclidean (or arithmetic) mean. Given a Riemannian metric, the associated Fréchet mean of N tensors S_1,..., S_N with arbitrary positive weights w_1, ..., w_N is defined as the point $\mathbb{E}(S_1, ..., S_N)$ minimizing the following *metric dispersion*:

$$\mathbb{E}(S_1, ..., S_N) = \arg\min_S \sum_{i=1}^{N} w_i \, \mathrm{dist}^2(S, S_i),$$

where dist(.,.) is the distance associated to the metric. The Log-Euclidean Fréchet mean is a direct generalization of the geometric mean of positive numbers and is given *explicitly* by:

$$\mathbb{E}_{\text{LE}}(S_1, ..., S_N) = \exp\left(\sum_{i=1}^{N} w_i \log(S_i)\right). \tag{4.3}$$

The closed form given by Eq. (4.3) makes the computation of Log-Euclidean means straightforward. On the contrary, there is no closed form for affine-invariant means $\mathbb{E}_{\text{Aff}}(S_1, ..., S_N)$ as soon as $N > 2$ [Moakher 05]. The affine-invariant is only *implicitly* defined through the following barycentric equation:

$$\sum_{i=1}^{N} w_i \log\left(\mathbb{E}_{\text{Aff}}(S_1, ..., S_N)^{-1/2}.S_i.\mathbb{E}_{\text{Aff}}(S_1, ..., S_N)^{-1/2}\right) = 0. \tag{4.4}$$

In the literature, this equation is solved iteratively, for instance using a Gauss-Newton method as detailed in [Pennec 06b, Lenglet 06a, Fletcher 04a]. This optimization method has the advantage of having quite a fast convergence speed, like all Newton methods.

Contrary to the affine-invariant case, the processing of tensors in the Log-Euclidean framework is simply Euclidean in the logarithmic domain. Tensors can be transformed first into symmetric matrices (i.e. vectors) using the matrix logarithm. Then, to simplify even more computations, these matrices with 6 degrees of freedom can be represented by 6D vectors in the following way:

$$\log(S) \simeq \vec{S} = \left(\log(S)_{1,1}, \log(S)_{2,2}, \log(S)_{3,3}, \sqrt{2}.\log(S)_{1,2}, \sqrt{2}.\log(S)_{1,3}, \sqrt{2}.\log(S)_{2,3}\right)^T,$$

where $\log(S)_{i,j}$ is the coefficient of $\log(S)$ placed in the (i, j) position. With this representation, the classical Euclidean norm between such 6D vectors is equal to a Log-Euclidean similarity-invariant distance between the tensors they represent. Note that this holds only for the particular similarity-invariant distance used in this work. To deal with another Log-Euclidean distance, one should adapt the 6D vector representation to the metric by changing adequately the relative weights of the matrix coefficients.

Once tensors have been transformed into symmetric matrices or 6D vectors, classical vector processing tools can be used *directly* on these 6D representations. Finally, results obtained on logarithms are mapped back to the tensor domain with the exponential. Hence, vector statistical tools or PDEs are readily generalized to tensors in this framework.

4.2.5 Comparison of the Affine-Invariant and Log-Euclidean Frameworks

As will be shown experimentally in the Results section, Log-Euclidean computations provide results very similar to their affine-invariant equivalent, presented in [Pennec 06b]. The reason behind this is the following: the two families of metrics provide two different generalizations to tensors of the geometric mean of positive numbers. By this we mean that the determinants of both Log-Euclidean and affine-invariant means of tensors are exactly equal to the *scalar* geometric mean of the determinants of the data [Arsigny 05b]. This explains the absence of swelling effect in both cases, since the interpolation of tensors along geodesics yields in both cases the same monotonic interpolation of determinants.

The two Riemannian means are even identical in a number of cases, in particular when averaged tensors commute in the sense of matrix multiplication. Yet, the two means are different in general, as shown theoretically in [Arsigny 05b] (the trace of the Log-Euclidean mean is always larger (or equal) than the trace of the affine-invariant mean) and experimentally in the Results section. More precisely, Log-Euclidean means are generally more anisotropic than their affine-invariant equivalent. We observed that this resemblance between the two means extends to general computations which involve averaging, such as regularization procedures, as is shown in the Results section.

4.3 Methods

4.3.1 Interpolation

Voxels in clinical DT images are often quite anisotropic. Algorithms tracking white matter tracts can be biased by this anisotropy, and it is therefore recommended (e.g. see [Basser 00]) to use isotropic voxels. A preliminary resampling step with an adequate interpolation method is therefore important for such algorithms. Adequate interpolation methods are also required to generalize to the tensor case usual registration techniques used on scalar or vector images. The framework of Riemannian metrics allows a direct generalization of classical resampling methods, by re-interpreting them as computing weighted means of the original data. Then the idea is to replace the Euclidean mean by its Riemannian counterpart, i.e. the Fréchet mean. See [Pennec 06b] for a more detailed discussion of this topic. This way one can generalize the classical linear, bilinear and trilinear interpolations to tensors with a Riemannian metric. For both metrics mentioned in this work, this entails in one case using directly Eq. (4.3) and in the other case iteratively solving Eq. (4.4).

4.3.2 Regularization

DT images are corrupted by noise, and regularizing them can be a crucial preliminary step for DTI-based algorithms that reconstruct the white matter connectivity. As shown in [Pennec 06b], Riemannian metrics provide a general framework to regularize to tensors usual vector regularization tools.

Practically, an anisotropic regularization is very valuable, since it allows a substantial reduction of the noise level while sharp contours and structures are mostly preserved. We focus here on a simple and typical Riemannian criterion for the anisotropic regularization of tensor fields, which is based on Φ-functions [Tschumperlé 05, Chefd'hotel 04]. In this context, the regularization is obtained by the minimization of a Φ-functional $Reg(S)$ given by:

$$Reg(S) = \int_\Omega \Phi\left(\|\nabla S\|_{S(x)}(x)\right) dx,$$

where Ω is the spatial domain of the image and $\Phi(s)$ a function penalizing large values of the norm of the *spatial gradient* ∇S of the tensor field $S(x)$. The spatial gradient is defined here as $\nabla S = (\frac{\partial S}{\partial x_1}, \frac{\partial S}{\partial x_2}, \frac{\partial S}{\partial x_3})$, where x_1, x_2 and x_3 are the three spatial coordinates, and where $\frac{\partial S}{\partial x_i}$ is the matrix describing how $S(x)$ linearly varies near x in the i^{th} spatial direction. Note that $\frac{\partial S}{\partial x_i}$ is only symmetric and not necessarily positive definite because it is given by an infinitesimal *difference* between two tensors, which is a non-convex operation. For more details on how spatial gradients can be practically computed, see [Pennec 06b] Section 5.

Here, we use the classical function $\Phi(s) = 2\sqrt{1 + s^2/\kappa^2} - 2$ [Chefd'hotel 04]. We would like to emphasize that contrary to the Euclidean case, the norm of ∇S *depends explicitly on the current point* $S(x)$ (see [Pennec 06b, Arsigny 05b] for more details) and is given by:

$$\|\nabla S\|^2_{S(x)} = \sum_{i=1}^{3} \left\| \frac{\partial S}{\partial x_i}(x) \right\|^2_{S(x)}.$$

In general, this dependence on the current point leads to complex resolution methods. Thus, in the affine-invariant case, these methods rely on an intensive use of matrix inverses, square roots, exponentials and logarithms [Pennec 06b]. However, in the Log-Euclidean framework the general Riemannian formulation is extremely simplified. The reason is that the dependence on the current tensor disappears on the logarithms of tensors [Arsigny 05b], so that the norm of the gradient is given by:

$$\|\nabla S(x)\|_{S(x)} = (< \nabla S(x), \nabla S(x) >_{S(x)})^{\frac{1}{2}} = \|\nabla \log(S(x))\|_{Id},$$

where Id is the identity matrix. This means that only the scalar product at the identity needs to be used. The transformation of tensors into their matrix logarithms transforms Riemannian computations at $S(x)$ into Euclidean computations at Id. As a consequence, *the energy functional can be minimized directly on the vector field of logarithms.* The regularized tensor field is given in a final step by the matrix exponential of regularized logarithms.

In the regularization experiments of this Chapter, the minimization method used is a first-order gradient descent with a fixed time step dt. We use an explicit finite difference scheme on logarithms in the Log-Euclidean case (see [Fillard 05b] for details about numerical schemes and others aspects of the implementation) and the geodesic marching scheme described in [Pennec 06b] in the affine-invariant case. In the Euclidean framework, we also use affine-invariant geodesic marching rather than a classical explicit scheme to limit the appearance of non-positive eigenvalues, proceeding similarly as in [Chefd'hotel 04]. Homogeneous Neumann boundary conditions are used, parameters were empirically chosen to be $\kappa = 0.05$, $dt = 0.1$, and 100 iterations are performed in the results shown in Fig. 4.5 and 50 iterations for those shown in Fig. 4.6.

4.3.3 Absolute Value of a Symmetric Matrix

When several variants of an algorithm are used to process tensors images, visualization tools are quite valuable to inspect the results. A simple solution is to visualize an image of the norm of the (Euclidean) difference between tensors. Regrettably, all information about *orientation* is lost in this case.

To visualize simultaneously the magnitude *and* the orientation of differences, one can use the *absolute value* of a symmetric matrix. Similarly to the exponential or square root, it is defined as the symmetric positive semi-definite matrix obtained by replacing the eigenvalues of the original matrix by their absolute values. Thus, this absolute value retains all the information about the magnitude and the orientation of any symmetric matrix, and can still be visualized directly with the usual ellipsoid representation. As a consequence, this mathematical tool is very useful to visualize the difference between two tensors, as can be seen in the Results section. We first introduced this tool in [Fillard 05c].

4.3.4 Materials

The experiments in this study are carried out partly on synthetic tensor images, and partly on a clinical DTI volume. The clinical scan of the brain was acquired with a 1.5-T MR imaging system (Siemens Sonata) with actively shielded magnetic field gradients (G maximum, 40 mT/m). A sagittal spin-echo single shot echo-planar parallel Grappa diffusion-weighted imaging sequence with acceleration factor two and six non collinear gradient directions was applied with two b values (b=0 and $1000 s.mm^{-2}$. Field of view: 24.0×24.0 cm; image matrix: 128×128 voxels; 30 sections with a thickness of 4mm; nominal voxel size: $1.875 \times 1.875 \times 4mm^3$. TR/TE= 4600/73 ms. The gradient directions used were as follows: $[(1/\sqrt{2}, 0, 1/\sqrt{2})$;

$(-1/\sqrt{2}, 0, 1/\sqrt{2})$; $(0, 1/\sqrt{2}, 1/\sqrt{2})$; $(0, 1/\sqrt{2}, -1/\sqrt{2})$; $(1/\sqrt{2}, 1/\sqrt{2}, 0)$; $(-1/\sqrt{2}, 1/\sqrt{2}, 0)]$ providing the best accuracy in tensor components when six directions are used [Basser 98]. The acquisition time of diffusion-weighted imaging was 5 minutes and 35 seconds. Image analysis was performed on a voxel-by-voxel basis by using dedicated software (DPTools, `http://fmritools.hd.free.fr`). Before performing the tensor estimation, an unwarping algorithm was applied to the DTI data set to reduce distortions related to eddy currents induced by the large diffusion-sensitizing gradients. This algorithm relies on a three-parameter distortion model including scale, shear, and linear translation in the phase-encoding direction [Haselgrove 96]. The optimal parameters were assessed independently for each section relative to the T2-weighted corresponding image by the maximization of the mutual information. However, due to the low signal-to-noise ratio in these images, part of the distortions remained. The tensors were estimated using the method described in [Fillard 05b], with a small regularization. The parameters of this estimation were set to $\lambda = 0.25$ and $\kappa = 0.1$. 50 iterations were used.

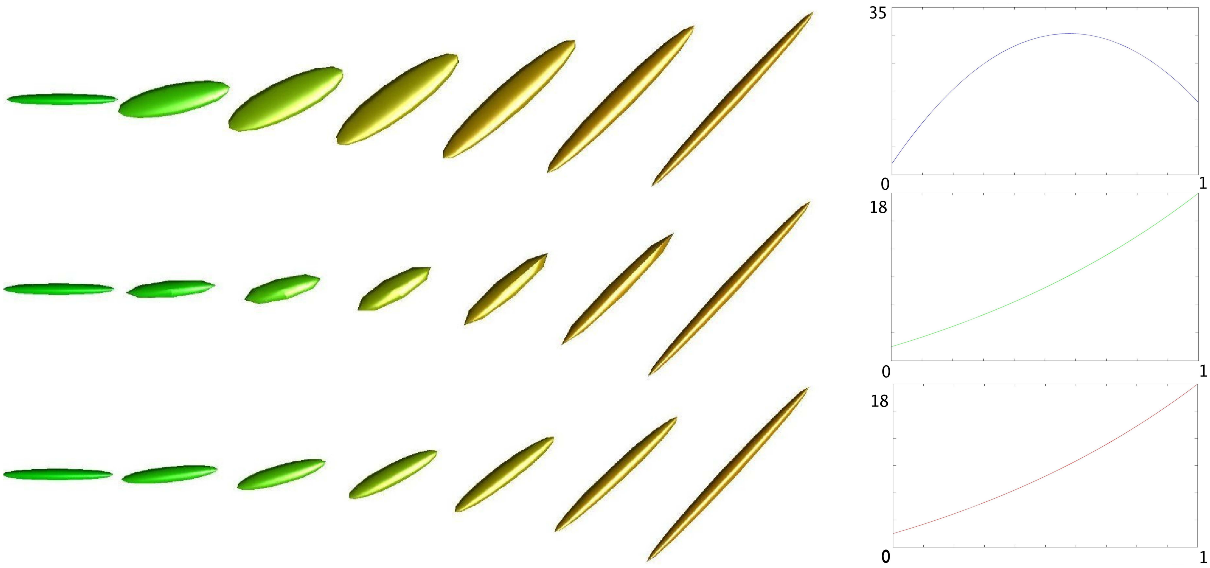

Figure 4.1: **Geodesic interpolation of two tensors**. **Left:** interpolated tensors. **Right:** graphs of the determinants of the interpolated tensors. **Top**: linear interpolation on coefficients. **Middle**: affine-invariant interpolation. **Bottom**: Log-Euclidean interpolation. The coloring of ellipsoids is based on the direction of dominant eigenvectors, and was only added to enhance the contrast of tensor images. Note the characteristic swelling effect observed in the Euclidean case due to a parabolic interpolation of determinants. This effect is not present in both Riemannian frameworks since determinants are monotonically interpolated. Note also that Log-Euclidean means are more anisotropic their affine-invariant counterparts.

4.4 Results

4.4.1 Interpolation

Results of the (geodesic) linear interpolation of two synthetic tensors are presented in Fig. 4.1. One can clearly see the swelling effect characteristic of the Euclidean interpolation, which has no physical interpretation. On the contrary, a monotonic (and identical) interpolation of determinants is obtained in both Riemannian frameworks. The larger anisotropy in Log-Euclidean means is also clearly visible in this figure.

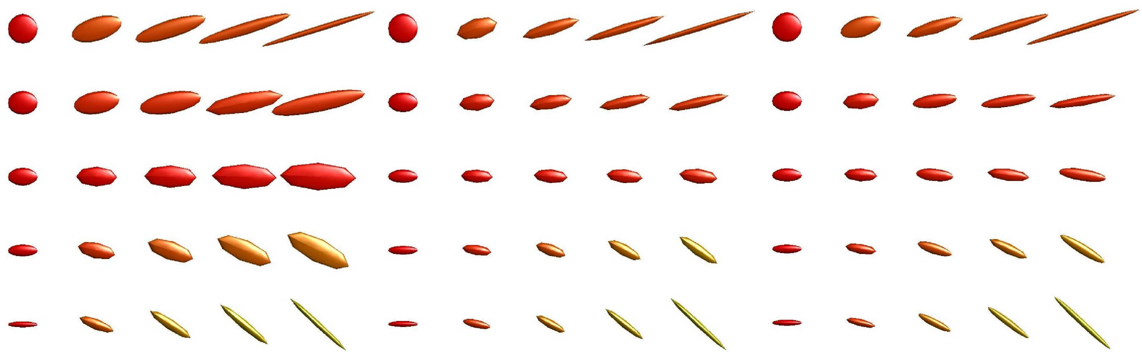

Figure 4.2: **Bilinear interpolation of 4 tensors at the corners of a grid**. **Left**: Euclidean reconstruction. **Middle**: affine-invariant reconstruction. **Right**: Log-Euclidean interpolation. Note the characteristic swelling effect observed in the Euclidean case, which is not present in both Riemannian frameworks. Note also that Log-Euclidean means are slightly more anisotropic than their affine-invariant counterparts.

Fig. 4.2 shows the results obtained for the bilinear interpolation of four synthetic tensors with three methods: Euclidean (linear interpolation of coefficients), affine-invariant and Log-Euclidean. Again, there is a pronounced swelling effect in the Euclidean case, which does not appear in both Riemannian cases. Also, there is a slightly larger anisotropy in Log-Euclidean means. One should note that the computation of the affine-invariant mean here is iterative, since the number of averaged tensor is greater than 2 (we use the Gauss-Newton method described in [Pennec 06b]), whereas the closed form given by Eq. (4.3) is used directly in the Log-Euclidean case. This has a large impact on computation times: $0.003s$ (Euclidean), $0.009s$ (Log-Euclidean) and $1s$ (affine-invariant) for a 5×5 grid on a Pentium M 2 GHz. Computations were carried out in the Matlab™framework, which explains the poor computational performance. A C++ implementation would yield much lower computation times, but the ratio would be comparable. This clearly demonstrate that Log-Euclidean metrics combine greater simplicity and performance, as compared to affine-invariant metrics, at least in terms of interpolation tasks.

From a numerical point of view, one should note that the computation of Log-Euclidean means is more *stable* than in the affine-invariant case. On synthetic examples, we noticed that for large anisotropies (for instance with the dominant eigenvalue larger than 500 times the smallest), large numerical instabilities appear, essentially due to limited numerical accuracy of the logarithm computations (even with double precision). This can complicate greatly the computation of affine-invariant means. In the case of our clinical DTI data, this type of phenomenon also occurs, although to a lesser degree. We observed that the computation of the affine-invariant mean can in this case be 5 to 10 times longer than usual at times, when the averaged data presents a substantial inhomogeneity. On the contrary, the computation of Log-Euclidean means is much more stable since the logarithm and exponential are taken only once and thus even very large anisotropies can be dealt with. Of course, on clinical DT images anisotropies are not so pronounced and drastic instabilities will not appear. But for the processing of other types of tensors with much higher anisotropies, this could be crucial.

To compare the Euclidean and Riemannian bilinear interpolations on clinical data, we

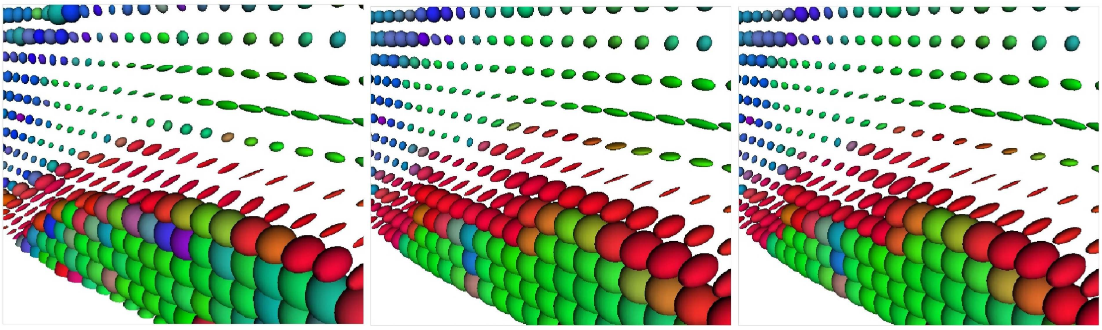

Figure 4.3: **Bilinear interpolation in a real DTI slice**. **Left**: Original DTI slice, before down-sampling. **Middle**: Euclidean interpolation. **Right**: Log-Euclidean interpolation. Half the columns and lines of the original DTI slice were removed before reconstruction with a bilinear interpolation. The slice is taken in the mid-sagittal plane and displayed in perspective. Again, the coloring of ellipsoids is based on the direction of dominant eigenvectors, and was only added to enhance the contrast of tensor images. Note how the tensors corresponding to the Corpus Callosum (in red, above the large and round tensors corresponding to a part of the ventricles) are better reconstructed (more anisotropic) in the Log-Euclidean case.

Similarity Measure	Eucl. interpol.	Aff.-inv. interpol.	Log-Eucl. interpol.
Mean Euclidean Error	0.2659	0.2614	0.2611
Mean Affine-invariant Error	0.2703	0.2586	0.2584
Log-Euclidean Error	0.2694	0.2577	0.2575

Table 4.1: **Mean reconstruction errors for the clinical slice reconstruction experiment.** The three interpolation results are quite close. However, both Riemannian frameworks perform slightly better than the Euclidean one, independently of the similarity measure considered. This is essentially due to the better Riemannian reconstruction of the Corpus Callosum.

have reconstructed by bilinear interpolation a down-sampled DTI slice. One column out of two and one line out of two were removed. The slice was chosen in the mid-sagittal plane where strong variations are present in the DT image. The results in Fig. 4.3 show that the tensors corresponding to the corpus callosum are better reconstructed in the Log-Euclidean case. Affine-invariant results are very similar to Log-Euclidean ones and not shown here. In other regions, the differences between the interpolations are much smaller. The mean reconstruction errors for all three frameworks are shown in Tab. 4.1. We assessed the reconstruction errors with three similarity measures: with our Euclidean, Log-Euclidean and affine-invariant metrics, we computed the mean distance between original and reconstructed tensors. As can be seen in this table, Log-Euclidean and affine-invariant results are quantitatively slightly better than Euclidean results, independently of the similarity measure considered. This is essentially due to the better reconstruction of the Corpus Callosum in both Riemannian cases.

4.4.2 Regularization

To compare the Euclidean, affine-invariant and Log-Euclidean frameworks, let us begin with a simple example where we restored a noisy synthetic image of tensors. The eigenvalues of the original tensors were set to $(2, 1, 1)$. We added some isotropic Gaussian white noise of variance 0.5 on the b_0 image and each of the 6 synthetic diffusion-weighted images, and tensors were estimated with the method presented in [Fillard 05b] with parameters $\lambda = 0.25$ and

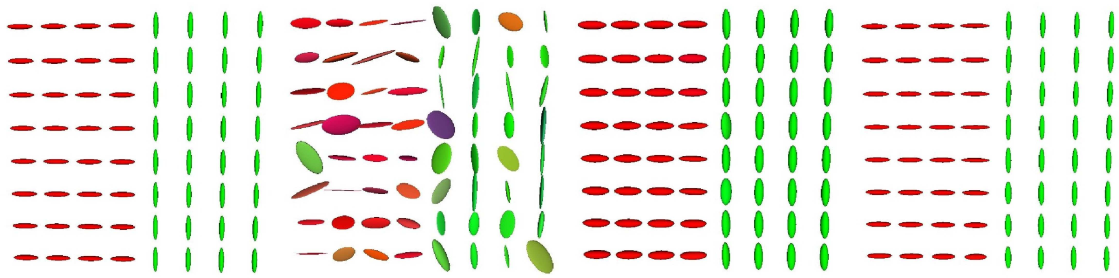

Figure 4.4: **Regularization of a synthetic DTI slice. Left**: original synthetic data. **Middle Left**: noisy data. **Middle Right**: Euclidean regularization. **Right**: Log-Euclidean regularization. The original data is correctly reconstructed in the Log-Euclidean case, as opposed to the Euclidean case where the result is spoiled by the swelling effect.

$\kappa = 0.1$ (the regularization was small). Results are shown in Fig. 4.4: surprisingly, although no anisotropic filtering other than the one described in the Methods Section was used, the boundaries between the two regions are kept perfectly distinct, thanks to the strong gradients in this area. Furthermore, the impact of the Euclidean swelling effect is clearly visible. On the contrary, both Riemannian frameworks yield very good results, the only extremely small difference being as predicted slightly more anisotropy for Log-Euclidean results. Affine-invariant results are not shown here because they are very close to the Log-Euclidean ones. Like in the interpolation reconstruction experiment, we assessed the reconstruction errors with the Euclidean, Log-Euclidean and affine-invariant metrics. For each metric, we computed the mean distance between original and reconstructed tensors. The quantitative results are shown in Tab. 4.2: as expected affine-invariant and Log-Euclidean results are close and yield much better results than in the Euclidean case, regardless of the similarity measure used.

Similarity Measure	Eucl. regul.	Aff.-inv. regul.	Log-Eucl. regul.
Mean Euclidean Error	0.228	0.080	0.051
Mean Affine-invariant Error	0.533	0.142	0.119
Log-Euclidean Error	0.532	0.135	0.111

Table 4.2: **Mean reconstruction errors for the synthetic regularization experiment.** Both Riemannian results are much better than the Euclidean one, independently of the similarity measure considered. This is due to the absence of swelling effect in both Riemannian cases.

Let us now turn to a clinical DTI volume, which presents a substantial level of noise. A quantitative evaluation or validation of the restoration results presented here remains to be done, and this general problem will be the subject of future work. However, as shown in Fig. 4.5, both Riemannian results are qualitatively satisfactory: the smoothing is done without blurring the edges in both Riemannian cases, contrary to the Euclidean results which are spoiled by a pronounced swelling effect, especially in the regions of high anisotropy. Also note that to a lesser degree, this swelling effect is present in regions with much less anisotropy, in fact almost everywhere except in the ventricles. The affine-invariant and Log-Euclidean results are very similar to each other, with only slightly more anisotropy in the Log-Euclidean case.

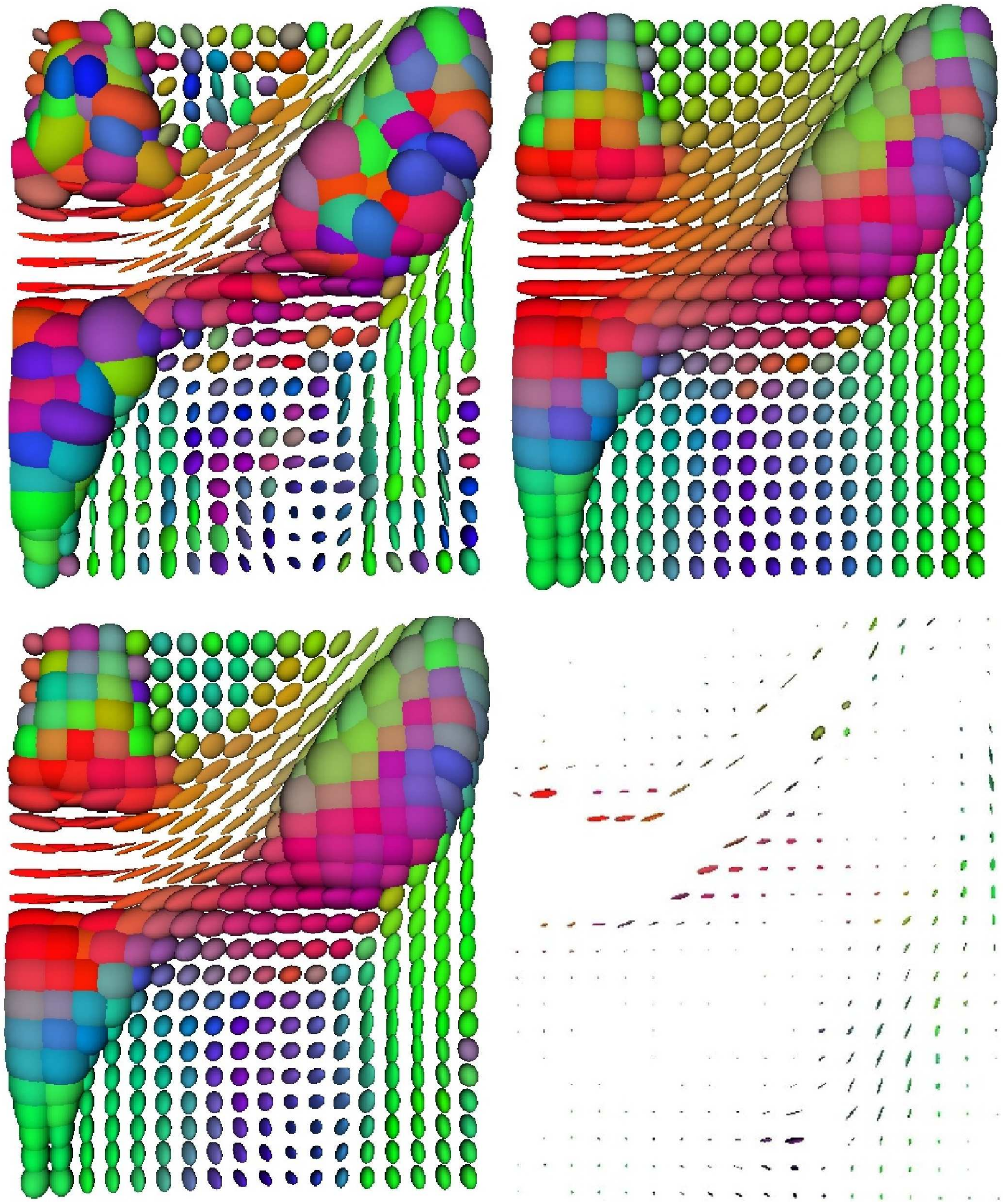

Figure 4.5: **Regularization of a clinical DTI volume (3D). Top Left**: close-up on a slice containing part of the left ventricle and nearby. **Top Right**: Euclidean regularization. **Bottom Left**: Log-Euclidean regularization. **Bottom Right**: highly magnified view ($\times 100$) of the absolute value of the difference between Log-Euclidean and affine-invariant results. The absolute value of tensors is taken to allow the simultaneous visualization of the amplitude *and* orientation of the differences. See the Methods section for a definition of the absolute value. Note that there is *no* tensor swelling in the Riemannian cases. On the contrary, in the Euclidean case, a swelling effect occurs almost everywhere (except maybe in the ventricles), in particular in regions of high anisotropy. Last but not least, the difference between Log-Euclidean and affine-invariant results is very small. Log-Euclidean results are only slightly more anisotropic than their affine-invariant counterparts.

To highlight this similarity, we display in Fig. 4.5 the *absolute values* of the (Euclidean) differences between affine-invariant and Log-Euclidean results. The definition of the absolute value of a symmetric matrix is given in the Methods section, and this mathematical tool is much useful to visualize the difference between two tensors. We can see in Fig. 4.5 that the differences are mainly concentrated along the dominant directions of diffusion, which is explained by the larger anisotropy in Log-Euclidean means. However, this relative difference is very small, of the order of less than 1%.

A regularization of DT images should not only correctly regularize the determinants of tensors, but also adequately regularize other scalar measures associated to tensors. In Fig. 4.6, the effect of the Log-Euclidean regularization on the fractional anisotropy (FA) and on the norm of the gradient are shown. In this experiment, only half of the regularization used to obtain the results of Fig. 4.5 is kept. As one can see, the regularization, which is performed directly on the tensors, induces a regularization of the FA and gradient norm. Qualitatively, major anisotropic structures have been preserved, including for example the internal capsule, while the noise has been substantially reduced.

As in the case of interpolation, the simpler Log-Euclidean computations are also significantly faster: our current implementation in C++ requires for 100 iterations 30 minutes in the Log-Euclidean case instead of 122 minutes for affine-invariant results on a Pentium Xeon 2.8 GHz with 1 Go of RAM. Our implementation has not been optimized yet and will be improved in the near future. Consequently, the values given here are only upper bounds of what can be achieved. However, the difference in computation times is typical and Log-Euclidean computations can even be 6 or 7 times faster than their affine-invariant equivalent [Arsigny 05b].

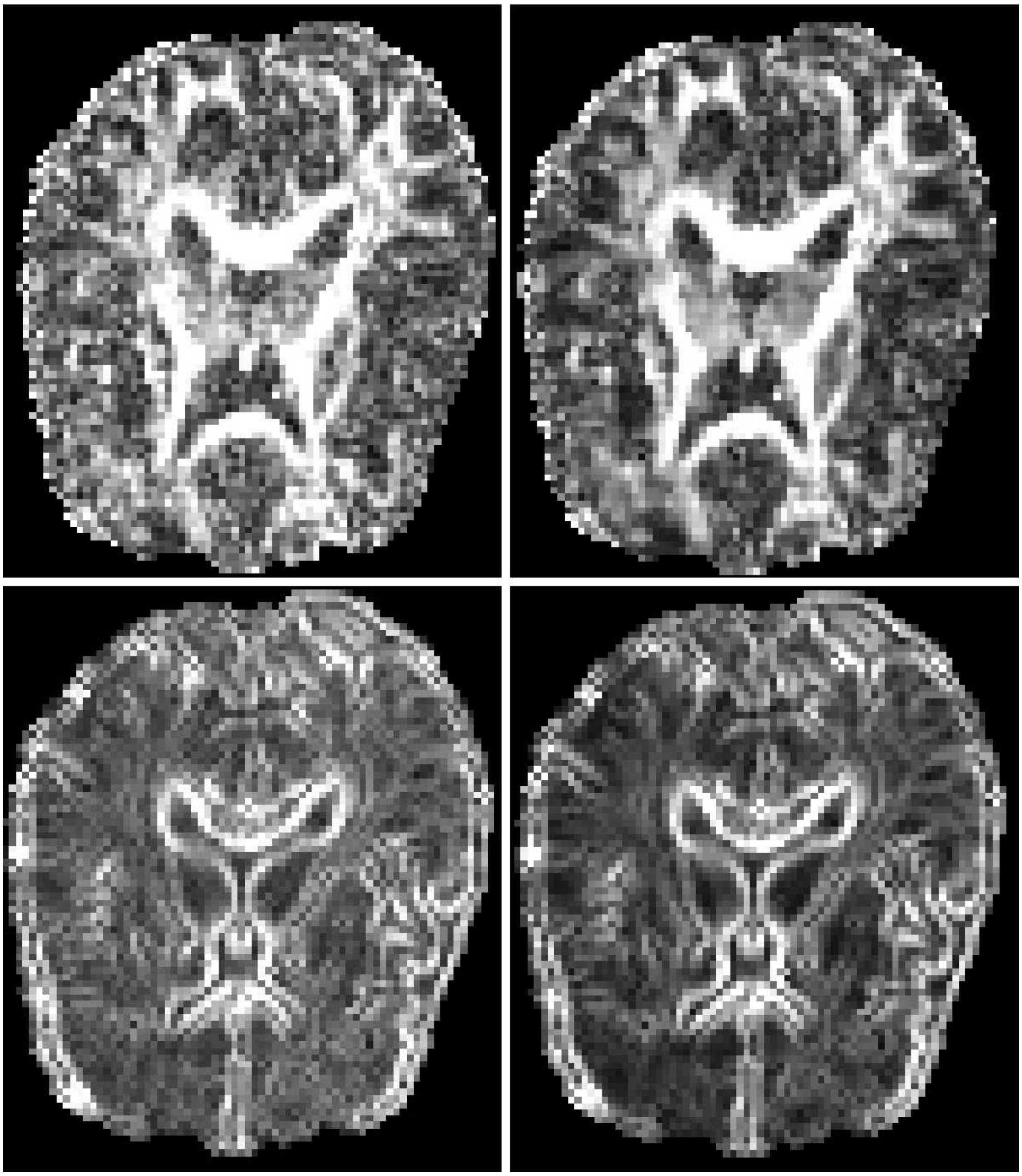

Figure 4.6: **Log-Euclidean regularization of a clinical DTI volume (3D): typical effect on FA and gradient. Top left:** FA before Log-Euclidean regularization. **Top right:** FA after regularization. **Bottom left:** Log-Euclidean norm of the gradient before regularization. **Bottom right:** Log-Euclidean norm of the gradient after regularization. The effect of the Log-Euclidean regularization on scalar measures like FA and the norm of the gradient is qualitatively satisfactory: the noise has been reduced while most structures are preserved.

4.5 Discussion and Conclusions

4.5.1 The Defects of Euclidean Calculus

As shown in the Results section, Log-Euclidean metrics correct the defects of the classical Euclidean framework [Tschumperlé 01]: the positive-definiteness is preserved and determinants are monotonically (geometrically, in fact) interpolated along geodesics. Log-Euclidean results are very similar to those obtained in the affine-invariant framework, only recently introduced for diffusion tensor calculus [Pennec 06b, Batchelor 05, Lenglet 06a, Fletcher 04a]. This is not surprising: we have shown that the two families of metrics are very close, since their respective Fréchet means are *both* generalizations to tensors of the geometric mean of positive numbers. Yet, these two metrics are different, and it is striking that this similarity in results is obtained with much simpler *and* faster algorithms in the Log-Euclidean case. This comes from the fact that all Log-Euclidean computations on tensors are equivalent to Euclidean computations on the logarithms of tensors, which are simple vectors.

Of course, this large simplification is obtained at the cost of affine-invariance, which is replaced by similarity-invariance for a number of Log-Euclidean metrics, like the one used in this study. This means that affine-invariant results cannot be biased by the coordinate system chosen, whereas Log-Euclidean results potentially can. However, invariance by similarity is already a strong property, since it guarantees that computations are not biased neither by the spatial orientation nor by the spatial scale chosen. Moreover, the very large similarity between the Log-Euclidean and affine-invariant results on typical clinical DT images show that this loss of invariance does not result in any significant loss of quality. One would have to change the system of coordinates very anisotropically, for instance rescaling one coordinate by a factor of 20 and leaving the other two unchanged, to substantially bias Log-Euclidean results. But such situations do not occur in medical imaging, where the usual changes of coordinates (e.g. changing current coordinates to Talairach coordinates) are not anisotropic enough to induce such a bias.

In terms of regularization, the Log-Euclidean framework also has the advantage of taking into account simultaneously *all* the information carried by tensors, like the affine-invariant one. This is not the case in methods based on the regularization of *features* extracted from tensors, like their dominant direction of diffusion [Coulon 04] or their orientation [Chefd'hotel 04]. An alternative representation of tensors are Cholesky factors, which are used in [Wang 04]. However, with this representation, tensors can leave the set of positive definite matrices during iterated computations, and the positive-definiteness is not easily maintained, as mentioned in [Wang 04]. Also, it is unclear how the smoothing of Cholesky factors affect tensors, whereas the smoothing of tensor logarithms can be interpreted as a *geometric* regularization of tensors which geometrically smoothes determinants.

In this Chapter, we have presented results obtained only with one particular Log-Euclidean metric, inspired from the classical Frobenius norm on matrices. The relevance of this particular choice will be investigated in future work. This is necessary, because it has been shown [Zhang 04] that the choice of Euclidean metric on tensors can substantially influence the registration of DT images. This should also be the case in the Log-Euclidean framework.

Moreover, in this work, we have assumed that diffusion tensors are positive-definite. This assumption is consistent with the choice of Brownian motion to model the motion of water molecules. It could be argued that our framework does not apply to diffusion tensors which have been estimated without taking into account this constraint, and can therefore have non-positive eigenvalues. But these non-positive eigenvalues are difficult to interpret from a physical point of view, and are essentially due to the noise corrupting DW-MRIs! The problem lies therefore in the estimation method and not in our framework. Non-positive eigenvalues can be avoided for example by using a simultaneous estimation and smoothing of tensors, which relies on spatial correlations between tensors to reduce the amount of noise. In this work, we have used the method described in [Fillard 05b], which was inspired by the approach developed in [Wang 04].

4.5.2 Conclusions and Perspectives

In this work, we have presented a particularly simple and efficient Riemannian framework for diffusion tensor calculus. Based on Log-Euclidean metrics on the tensor space, this framework transforms Riemannian computations on tensors into Euclidean computations on vectors in the domain of matrix logarithms. As a consequence, classical statistical tools and PDEs usually reserved to vectors are simply and efficiently generalized to tensors in the Log-Euclidean framework.

In this Chapter, we have only focused on two important tasks: the interpolation and the regularization of tensors. But this metric approach can be effectively used in all situations where diffusion tensors are processed. Indeed, we have presented efficient Log-Euclidean extrapolation techniques in [Arsigny 05b], and Log-Euclidean statistics presented in Chapter 3. Another important task is the estimation of tensors from DW-MRIs. Adapting ideas from [Wang 04] to the Log-Euclidean framework, we have completed a joint estimation and regularization of diffusion tensors directly from the Stejskal-Tanner equations [Fillard 05b] and we have recently adapted this estimation approach to the Rician nature of the noise corrupting MR images [Fillard 06a, Fillard 06b]. This type of joint estimation and smoothing is largely facilitated by the Log-Euclidean framework because all computations are carried out in a vector space.

In future work, we will study in further detail the restoration of noisy DT images. In particular, we plan to quantify the impact of the regularization on the tracking of fibers in the white matter of the human nervous system. We also intend to use these new tools to model and reconstruct better the anatomical variability of the human brain with tensors as we began to do in [Fillard 05c]. Also, the generalization of our approach to more sophisticated models of diffusion like generalized diffusion tensors [Özarslan 03] or Q-balls [Tuch 04] is a challenging task we plan to investigate.

Acknowledgements

The authors thank Denis Ducreux, MD, Kremlin-Bicêtre Hospital (France), for the DT-MRI data he kindly provided for this study.

A patent is pending for the general Log-Euclidean processing framework on tensors (French filing number 0503483, 7th of April, 2005).

Chapter 5

Original Polyrigid and Polyaffine Transformations

After having focused on tensor processing in the previous two Chapters, we put the stress on geometrical deformations and their processing in the sequel of this thesis.

In this Chapter, we present two novel classes of transformations, called *polyrigid* and *polyaffine*, which parameterize locally rigid or affine *diffeomorphic* deformations with a small number of flexible degrees of freedom.

Contents

5.1 Introduction 72
5.2 Theory of Polyrigid and Polyaffine Transformations 73
5.2.1 Regions of Influence and Interpolation of Sparse Data 73
5.2.2 A Framework with ODEs 75
5.2.3 Theoretical Properties of Polyrigid Transformations 76
5.2.4 Extension to Polyaffine Transformations 76
5.2.5 Summary of the properties of Polyrigid Transformations 77
5.3 Implementation of Polyrigid Transformations 78
5.3.1 Discretization Schemes 78
5.3.2 Implementation with the Insight Toolkit 79
5.4 Registration of Histological Slices 81
5.4.1 Object of the Study and Experimental Setup 81
5.4.2 Limitations of the First-Order Gradient Descent. 84
5.4.3 Registration Results using a Levenberg-Marquardt Algorithm 88
5.4.4 Alternating Optimization 88
5.5 Results with more Complex Regions 90
5.5.1 The Shape of the Regions of Influence. 90
5.5.2 Results Obtained with Three Anchor Points 90
5.6 Conclusions and Perspectives 93
5.7 Appendix: First Derivatives of Original Polyrigid Transformations 94

Abstract. We describe in this Chapter a novel kind of geometrical transformations, named polyrigid and polyaffine. These transformations efficiently code for locally rigid or affine deformations with a small number of intuitive parameters. They can describe compactly large rigid or affine movements, unlike most free-form deformation classes. Very flexible, this tool can be readily adapted to a large variety of situations, simply by tuning the number of rigid or affine components and the number of parameters describing their regions of influence.

The displacement of each spatial position is defined by a continuous trajectory that follows a differential equation which averages the influence of each rigid or affine component. We show that the resulting transformations are diffeomorphisms, smooth with respect to their parameters. We devise a new and flexible numerical scheme to allow a trade-off between computational efficiency and closeness to the ideal diffeomorphism. Our algorithms are implemented within the Insight Toolkit, whose generic programming style offers rich facilities for prototyping. In this context, we derive an effective optimization strategy of the transformations which demonstrates that this new tool is highly suitable for inference.

The whole framework is exemplified successfully with the registration of histological slices. This choice is challenging, because these data often present locally rigid deformations added during their acquisition, and can also present a loss a matter, which makes their registration even more difficult.

Related Publications. Part of this Chapter was first published in an INRIA research report [Arsigny 03a] and during the international conference MICCAI'03 [Arsigny 03b], where this work was awarded the 'Best Student Presentation in Medical Image Processing and Visualization'. This work was extended to be published in the international journal Medical Image Analysis in 2005 [Arsigny 05c].

5.1 Introduction

As we have seen in the Introduction of the thesis in Section 1.1.2, locally rigid or affine transformations with a small or moderate number of degrees of freedom are particularly relevant for the registration of medical images in a number of situations. This includes the the case of anatomical structures incorporating rigid elements (such as bone articulations, or structures which are subject to simple local deformations, like histological slices), as well as the case of structures which are subject to simple local deformations, like histological slices.

Our goal in this Chapter is be to define new parametric transformations that exhibit a locally rigid or affine behavior, and that can be efficiently implemented. Also, a very desirable property is *invertibility*, which is not guaranteed in the approaches based on splines or other interpolation techniques, except in the case of the Geodesic Interpolating Splines [Camion 01], which are limited to the interpolation of a sparse set of displacements.

An approach was proposed in [Little 96] to smoothly interpolate a deformation outside independent rigidly moving regions. This computationally efficient approach is unfortunately "parameterized" by the motion *and the arbitrarily complex shape* of each rigidly-moving region. As a consequence, it is not straightforward to use this model for inference (i.e. non-rigid registration). Moreover, the invertibility of the interpolated transformation is not always ensured. This interpolation method is used in [Pitiot 03], which deals with the registration of histological slices. This is a pivotal issue for the fusion of MR images and histological slices, which is a promising technique for building precise atlases of brain structures [Ourselin 01a], [Bardinet 02].

Our idea is to use simple fuzzy regions defined by very few parameters: mainly the position of the center, a typical radius of influence and the associated rigid or affine transformation.

We show in Section 5.2 that a simple average of the displacement induced by each region leads to invertibility problems. Thus, we develop an infinitesimal approach where the displacement is obtained by the integration of the average speed. To address the implementation efficiency, we investigate in Section 5.3 several numerical schemes. The result is a new family of invertible and fully parametric transformations that we called *polyrigid and polyaffine transformations*. The Insight Toolkit (ITK) is a very attractive framework for the implementation of these new transformations, since it provides a base class for all parametric transformations and powerfull registration tools that greatly facilitate the rapid development of new algorithms. We describe also in Section 5.3 how Polyrigid Transformations are implemented within this framework. This code is freely available on the Internet [1]. We show in Section 5.4 that this new general tool is well-suited for the non-rigid registration of articulated-like object. This is exemplified on 2D histological slices. In Section 5.5, we also present preliminary results that show how polyrigid transformations can be refined to describe precisely regions of influence of a complex shape.

5.2 Theory of Polyrigid and Polyaffine Transformations

5.2.1 Regions of Influence and Interpolation of Sparse Data

Simple Parameterization of Regions of Influence

In order to model transformations having several distinct rigid behaviors in different regions, it is necessary to define how each component of the global transformation is anchored geometrically. One could of course choose to have regions of influence of arbitrary shape, like in [Little 96], but this is not convenient for inference. Having a reduced number of parameters describing the shape and extent of each region of influence allows for simple optimization of these parameters, which is a highly desirable feature for registration purposes.

We propose here a Gaussian model for regions of influence: to each region we have an anchor point $a \in \mathbb{R}^n$, and in addition we also have two other parameters, a typical distance σ and a parameter p such that the influence of the i-th component is described by a "weight" $w_i(x) = p_i.G_{(a_i,\sigma_i)}(x)$ where $G_{(a_i,\sigma_i)}$ is the Gaussian of mean a_i and of standard deviation σ_i. Thus, instead of using regions in which the transformation is purely rigid like in [Little 96], we propose "fuzzy" regions, which makes the transitions or interpolations between the regions straightforward to handle.

In order to obtain a global transformation from several weighted components, the classical way of mixing each local behavior is to average the displacements according to the weights [Sheppard 68]:

$$T(x) = \frac{\sum_i w_i(x) T_i(x)}{\sum_i w_i(x)}. \tag{5.1}$$

Here, the transformations $(T_i)_{i\in 1\cdots N}$ are rigid transformations. They are parameterized by the rotation matrixes (R_i) and the translations (t_i). Their action on a point is given by:

$$\forall x \in \mathbb{R}^n,\ T_i(x) = R_i.x + t_i.$$

Weaknesses of the Classical Averaging

The transformation obtained via (5.1) is smooth, both with respect to spatial coordinates and its parameters. Nonetheless, it has several major drawbacks:

[1] The FTP address is the following:
ftp://ftp-sop.inria.fr/epidaure/Softs/Arsigny/MediaReview/PolyTransfosSrcMediaReview

- Its invertibility is not guaranteed, and indeed will not be assured in many cases, for example if the displacements are large.

- In the favorable case where the inverse exists, it has in general no simple form and has to be estimated by an *ad hoc* technique, for instance using a general deformation field, which is iteratively optimized to obtain the inverse

- Is the behavior of this direct averaging procedure really qualitatively satisfactory? In Fig. 5.1, an example shows that in the case of a mixture of rotations, points do not in general turn around the centers of the rotations. On the contrary the approach proposed here has this property.

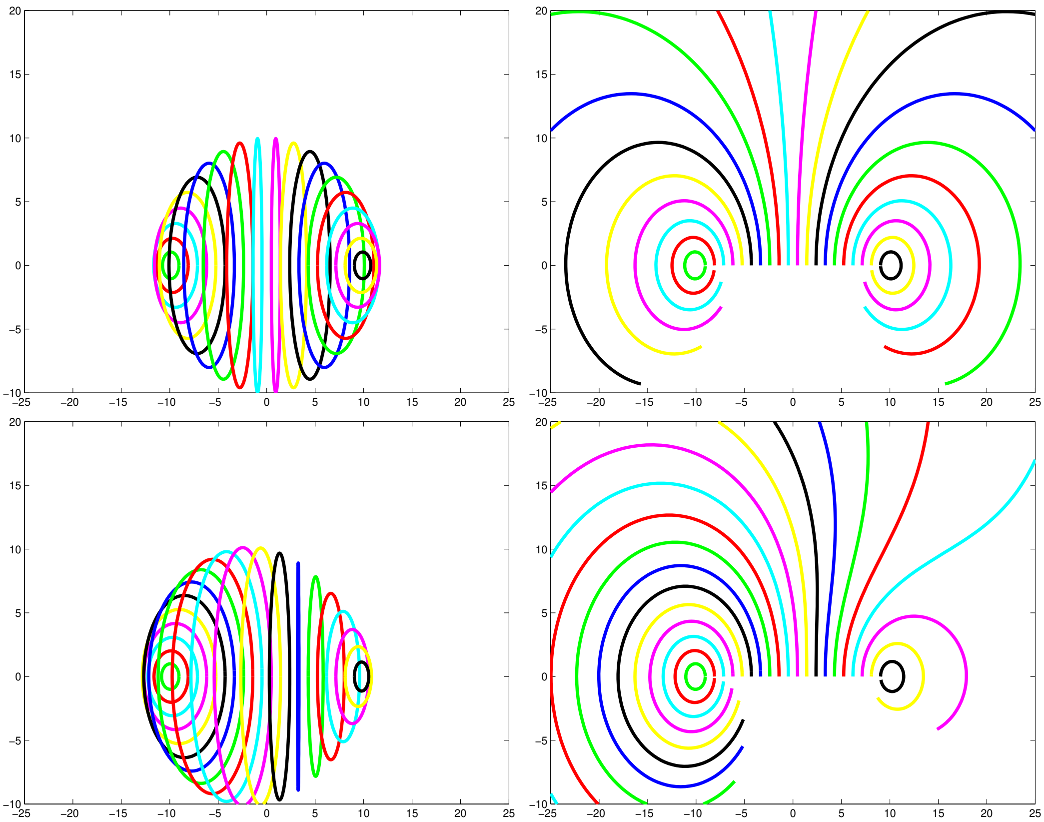

Figure 5.1: **Simple averaging (left) versus proposed approach (right) .** Here, the polyrigid transformation has two rotation components, which have exactly opposite angles. We consider in this figure the various trajectories of points originally in the segment joining the two centers. These trajectories are constituted by all the final positions of the initial points as we progressively increase the angle of rotation from 0 to 2π radians. On top, the two relative weights p_1 and p_2 are equal whereas on the bottom that of the left component is substantially higher than the other, hence the greater influence of the transformation anchored in the left. The form of trajectories show that points moving under the action of a polyrigid transformation do turn around the centers of rotations of the transformation. This property is not verified in the case of the classical averaging.

These reasons have led us to develop a new kind of averaging procedure tackling the above-mentioned problems.

5.2.2 A Framework with ODEs

Invertibility and ODEs

The challenge facing us at this point is the following: how to mix several transformations according to some weight functions in an invertible way? As explained in more details Section 2.2, a classical way of obtaining invertible and smooth transformations is to use ordinary differential equations (ODEs) [Tenenbaum 85]. A particle governed by an ODE follows an equation of the form:

$$\dot{x}(s) = V(x, s).$$

If V is smooth (for instance $\mathcal{C}^1$) with respect to x (spatial coordinates) and s (time), *and if the solution $x(s)$ is defined for all time*, then the *flow* $\Phi(x, s)$ associated to the ODE defines a family of diffeomorphisms. This operator associates to a given starting position the position reached at time s by the particle following the evolution prescribed by the ODE.

More precisely, for each $s \in \mathbb{R}$, we have that $x \mapsto \Phi(x, s)$ is a diffeomorphism from $\mathbb{R}^n$ to $\mathbb{R}^n$. Our approach is based on this key result.

The Case of Rigid Transformations

From the classical results of linear algebra, it is obvious that a rigid transformation is invertible, and its inverse is simply obtained by inverting the rotation part and adapting the translation component in the appropriate way. But another viewpoint can be used to prove the invertibility, using ODEs. More precisely, we can associate to a rigid transformation the following ODE, where the nature of the A_i matrix is explained just below:

$$\dot{x}(s) = V_i(x, s) = t_i + A_i(x - s\, t_i) \text{ for } s \in [0, 1]. \tag{5.2}$$

This is obtained by differentiating the trajectory equation $x(s) = st_i + \exp(sA_i)$. At time 0, we start with the initial position and the image for the rigid transformation is obtained at time 1. Since V_i is smooth and trajectories are defined for all time, the above-mentioned result applies.

In Eq. (5.2), we denote by A_i one of the logarithms of the rotation matrix R_i, which verifies the equality: $\exp(A_i) = R_i$ where exp is the matrix exponential. Since R_i is a rotation, it always has a real logarithm, which is a skew symmetric matrix. For example, in 3D, let $r = (r_x, r_y, r_z)^T$ be a rotation vector associated to a rotation R. We can then define a skew symmetric matrix A associated to r that is a logarithm of R with the relation:

$$A = \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix}.$$

A Continuous Averaging Procedure with ODEs

In order to insure the invertibility of our averaged transformation, let us define a new ODE. The idea is simply to average according to weights the *velocity vectors* associated to each component, instead of averaging the final results:

$$\dot{x}(s) = V(x, s) = \frac{\sum_i w_i(x) V_i(x, s)}{\sum_i w_i(x)} \tag{5.3}$$

Ideally, we would like to define our averaged transformation as $T(x) = \Phi(x, 1)$, where Φ is the flow associated to the ODE (5.3).

This means that each component will influence the motion of a particle accordingly with the weights modeling its influence in space. The result obtained at time 1 is the image of initial position x under the action of the average transformation.

5.2.3 Theoretical Properties of Polyrigid Transformations

Existence and Invertibility of Polyrigid Transformations

Since in (5.3) $V(x,s)$ is $\mathcal{C}^\infty$ with respect to spatial position and time, it only remains to be proved that the evolution does not lead to explosion towards infinity before time 1.

Theorem 5.1. *All solutions of Eq. (5.3) have an infinite life-span, i.e. they are defined for all time, whatever the rigid transformations may be. The polyrigid transformations defined via $\Phi(.,1)$ are thus well-defined and diffeormorphic.*

Proof. There exists three positive constants C_1, C_2 and C_3 such that $\|V(x,s)\|_2 \leq C_1 + C_2|s| + C_3\|x\|$. For instance, take $C_1 = \max_i \|t_i\|_2$ and $C_2 = \max_i \|A_i t_i\|_2$ and $C_3 = \max_i \|A_i\|_2$ where $\|A_i\|_2$ refers to the Frobenius norm of matrix A_i, equal here to the L^2 norm of the associated rotation vector. This yields via a classical bounding that $\forall s,\ \|\Phi(x,s)\| \leq e^{C_3|s|}(\|\Phi(x,0) + (1-e^{-C_3|s|})(C_1/C_3 + |s|C_2/C_3))$, which suffices to prove the result because it shows that the position of the particle evolving with Eq. (5.3) is contained within a sphere whose radius grows exponentially this time. □

A Simple Inverse. The inverse of the transformation is obtained here in a simple fashion: it suffices to go back in time! The skew matrix is changed into its opposite, the translation also and s becomes $1-s$. The inverse transformation thus takes here a simple form.

Differentiability with Respect to the Parameters

We have just seen that any given system of rigid transformations can be averaged so as to yield a diffeomorphism. But, what smoothness can be guaranteed with respect to the parameters? Differentiability is crucial so as to enable simple optimization of the transformation in a registration framework. We have the following result:

Theorem 5.2. *Polyrigid transformation are $\mathcal{C}^\infty$ with respect to all parameters.*

Proof. This comes from the differentiability of the flow of an ODE. Indeed, let us define the new ODE $\dot{z}(s) = W(z,s)$ where $z = (x,p)$, x being the spatial coordinates of a particle and p the parameters of the polyrigid transformation written in a vectorial fashion, and where the velocity vector $W(z,s) = (V(x,s),0)$. Thus, x evolves according to (5.3) and that p does not change as time goes by. W is $\mathcal{C}^\infty$ and the solutions are defined for all time since those of (5.3) are. This implies the differentiability of the flow associated to this ODE, which is exactly the differentiability of the polyrigid transformation with respect to its parameters. □

5.2.4 Extension to Polyaffine Transformations

A Simple Extension via the Real Logarithm

One may wonder to what extent it is possible to use the framework presented above to work with locally affine transformations. This can be done in a direct way if each affine transformation (M_i, t_i) has a linear part M_i that admits a *real logarithm*, i.e., if there exists $A_i \in M(n)$ such that $\exp(A_i) = M_i$. Then, we can adopt all coordinates of A_i as new scalar parameters to work with, and all the results of this section hold for this other type of transformation, that we could call polyaffine.

Unfortunately, not all real invertible matrices $M_i \in GL(n)$ have a real logarithm. Even among real matrices with a positive determinant, this is not true. This is unsatisfactory, because compositions of a dilatation and a rotation are deformations that are essential to the affine generalization of polyrigid transformations.

A General Extension

In order to define polyaffine transformations, what we basically need is simply a smooth trajectory from the identity to any given affine transformation. By differentiating with respect to time we want to obtain a simple ODE. To define such an evolution, we have the following general result: any element of a real connected Lie Groups is equal to the product of *two* exponentials [Wüstner 03]. Indeed, in the linear part of affine transformations, the singular value decomposition yields that $M_i = e^{A_i}.e^{S_i}$ with A_i and S_i respectively skew and symmetric matrices. An equivalent of Eq. (5.2) for polyaffine transformations is thus:

$$\dot{x}(s) = t_i + (A_i + e^{sA_i} S_i e^{-sA_i}).(x - s\, t_i). \tag{5.4}$$

All results mentioned above still hold in this case and hence we can define general polyaffine transformation, smooth both w.r.t. spatial coordinates and parameters. But other parameterization could be chosen: we can also write $M_i = e^{\widetilde{S}_i}.e^{A_i}$ by regrouping the factors of the SVD differently, where $\widetilde{S}_i \neq S_i$ in general. Several extensions are possible and will be investigated in future work.

5.2.5 Summary of the properties of Polyrigid Transformations

In this section, we defined a new class of transformations, modeling a mixture of rigid transformations, whose influence is geometrically anchored in a simple way. These transformations are diffeomorphisms and smooth with respect to all of their parameters. The following tables summarize the various parameters of the transformations (Table 5.1), and the number of scalar parameters obtained in 2 dimensions or 3 dimensions (Table 5.2), where a comparison is be made with B-Splines.

Region parameters	Deformation parameters
Anchor points: (a_i)	Rotation vectors: (r_i)
Standard deviations: (σ_i)	Translation vectors: (t_i)
Relative weights: (p_i)	

Table 5.1: The two types of parameters of polyrigid transformations.

Number of components	2D	B-Spline equivalent	3D	B-Spline equivalent
2	13	3 control points	21	3 c.p.
3	20	5 c.p.	32	5 c.p.
4	27	6 c.p.	43	7 c.p.
N	$7N-1$	$\frac{7N-1}{4}$ c.p.	$11N-1$	$\frac{11N-1}{6}$ c.p.

Table 5.2: Summary of the various types of parameters for polyrigid transformations, and a comparison between their number of parameters and that of the B-Splines.

5.3 Implementation of Polyrigid Transformations

5.3.1 Discretization Schemes

Since in the general case there does not exist a closed form for the position of a point moving under the action of a polyrigid transformation, it is a necessity to resort to a numerical scheme to integrate the ODE defining the transformation. In other words, the trajectory of a point moving via (5.3) has to be sampled: a number of intermediate points N and a rule for obtaining the successive positions $(x_i)_{i\in 0\cdots N}$ have to be chosen, so that the curve defined by the points converges toward the real continuous curve given by the ODE.

In our domain of application, i.e. medical imaging, we have an additional constraint, due to the volume of data much must be processed in common applications. Thus, the numerical scheme should be as computationally inexpensive as possible. This is all the more true here that we make use in Section 5.3 of the first and second derivatives of the numerical scheme, which forbids the use of classical schemes such as Runge-Kutta's.

The Consistent First Order Scheme

The consistence of a numerical scheme is a crutial notion. It is a condition that must be verified to insure the convergence towards the continuous solution when the time step goes to zero. It simply means that when we take the Taylor expansion of the solution of the ODE with respect to time around zero, a numerical scheme must have the same expansion up to a certain order. We say also that a scheme is of a certain order when the coefficients of its Taylor series vanish after that order.

The consistent first order scheme is simply given in the following way: we define the operators $T_1^{1/N}$ and $T_1^{k/N}$ by:

$$\begin{cases} T_1^{1/N}(x,s) = x + \frac{1}{N}V(x,s). \\ T_1^{k/N}(x) = \underbrace{T_1^{1/N}(.,(k-1)/N) \circ \cdots \circ T_1^{1/N}}_{k \text{ compositions}}(x,0). \end{cases} \tag{5.5}$$

The points (x_i) are obtained recursively using:

$$\begin{cases} x_0 = x. \\ \text{for } 1 \leq n \leq N: \ x_n = T_1^{1/N}(x_{n-1},(k-1)/N) = T_1^{k/N}(x_0). \end{cases} \tag{5.6}$$

This simply means that starting at x_0, we jump from x_{n-1} to x_n by adding $\frac{1}{N}$ times the velocity vector $V(x_{n-1}, \frac{n-1}{N})$.

An Efficient Second Order Scheme

The scheme described above is not really satisfactory. In the case of a single rigid component, the approximation makes points move along a diverging spiral instead of a circle (if the transformation is a rotation). This is regrettable, and a simple way of suppressing this approximation is to use the following second-order scheme using new operators $T_2^{1/N}$ and $T_2^{k/N}$:

$$\begin{cases} T_2^{1/N}(x,s) = x + \frac{\sum_i w_i(x)(\frac{1}{N}t_i + (e^{\frac{A_i}{N}} - Id)(x - st_i))}{\sum_i w_i(x)}. \\ T_2^{k/N}(x) = \underbrace{T_2^{1/N}(.,(k-1)/N) \circ \cdots \circ T_2^{1/N}}_{k \text{ compositions}}(x,0). \end{cases} \tag{5.7}$$

Instead of averaging the velocity vectors of each component, we average instead the displacements that would be observed if each component was acting alone during a small interval of time of length $\frac{1}{N}$. This scheme is first-order consistent, but not second-order consistent (it captures only part of the second-order terms). But it is *exact* in the case of a single component, and its convergence is much faster than the first one as shown in Fig. 5.2. Furthermore, the diverging spiral phenomenon observed for the first scheme disappears.

Derivatives of the Transformation

Computing the derivatives of the transformation with respect to its parameters is necessary to use a gradient descent approach. Let us consider for instance a simple registration strategy, where we want to register two images I and J with the sum of square differences (SSD) criterion. This does not imply that our approach is restricted to that particular case: one could obviously compute the derivatives for other criteria. We take two images, J and I, and we want to register J onto I using the inverse of a polyrigid transformation T_p, where p are the parameters of the transformation. In this case, the criterion to be minimized is:

$$S(I, J \circ T_p) = \int_\Omega \|I(x) - J \circ T_p(x)\|^2 \, dx.$$

The gradient of S with respect to p is the following:

$$\frac{\partial S}{\partial p}(I, J \circ T_p) = 2 \int_\Omega (J \circ T_p(x) - I(x)).(\nabla J \circ T_p)(x).\frac{\partial T_p}{\partial p}(x) \, dx.$$

In the last equation, the symbol "." denotes the matrix product. In order to compute the derivatives of the transformation with respect to the parameters, we simply computed the derivatives of each of the schemes. This is done again with a recursive formulation:

$$\frac{\partial T_p^{\frac{k}{N}}(x)}{\partial p} = \frac{\partial T_p^{\frac{1}{N}}\left(\cdot, \frac{k-1}{N}\right)}{\partial p}\left(T_p^{\frac{k-1}{N}}(x)\right) + \frac{\partial T_p^{\frac{1}{N}}\left(\cdot, \frac{k-1}{N}\right)}{\partial x}\left(T_p^{\frac{k-1}{N}}(x)\right).\frac{\partial T_p^{\frac{k-1}{N}}(x)}{\partial p}.$$

For a first-order gradient descent, only the above gradient is necessary. But for a second-order gradient descent, we will also need the second-order derivative:

$$\begin{aligned} \tfrac{\partial^2 S}{\partial p^2}(I, J \circ T_p) \quad = \quad & 2\Big\{ \tfrac{\partial T_p}{\partial p}(x)^T.(\nabla J \circ T_p)(x)^T.(\nabla J \circ T_p)(x).\tfrac{\partial T_p}{\partial p}(x) \\ & + (J \circ T_p(x) - I(x))\tfrac{\partial T_p}{\partial p}(x)^T.(\tfrac{\partial^2 J}{\partial x^2} \circ T_p(x)).\tfrac{\partial T_p}{\partial p}(x) \\ & + (J \circ T_p(x) - I(x))\tfrac{\partial T_p}{\partial p}(x).\tfrac{\partial^2 T_p}{\partial p^2(x)}\Big\}. \end{aligned} \tag{5.8}$$

A useful approximation is obtained by keeping only the first term of this equation. It has the nice property of being symmetric positive, and is a good approximation of the Hessian as long as that the difference of intensities $(J \circ T_p(x) - I(x))$ is small. Therefore, the more we will be close to a "good" solution, the more valid this approximation is. For detailed formulas, we refer the reader to Appendix 5.7.

5.3.2 Implementation with the Insight Toolkit

In order implement these new transformations, we chose to use the framework of the Insight Toolkit[2], which is a rich and rapidly developing set of tools dedicated to the segmentation

[2]http://www.itk.org

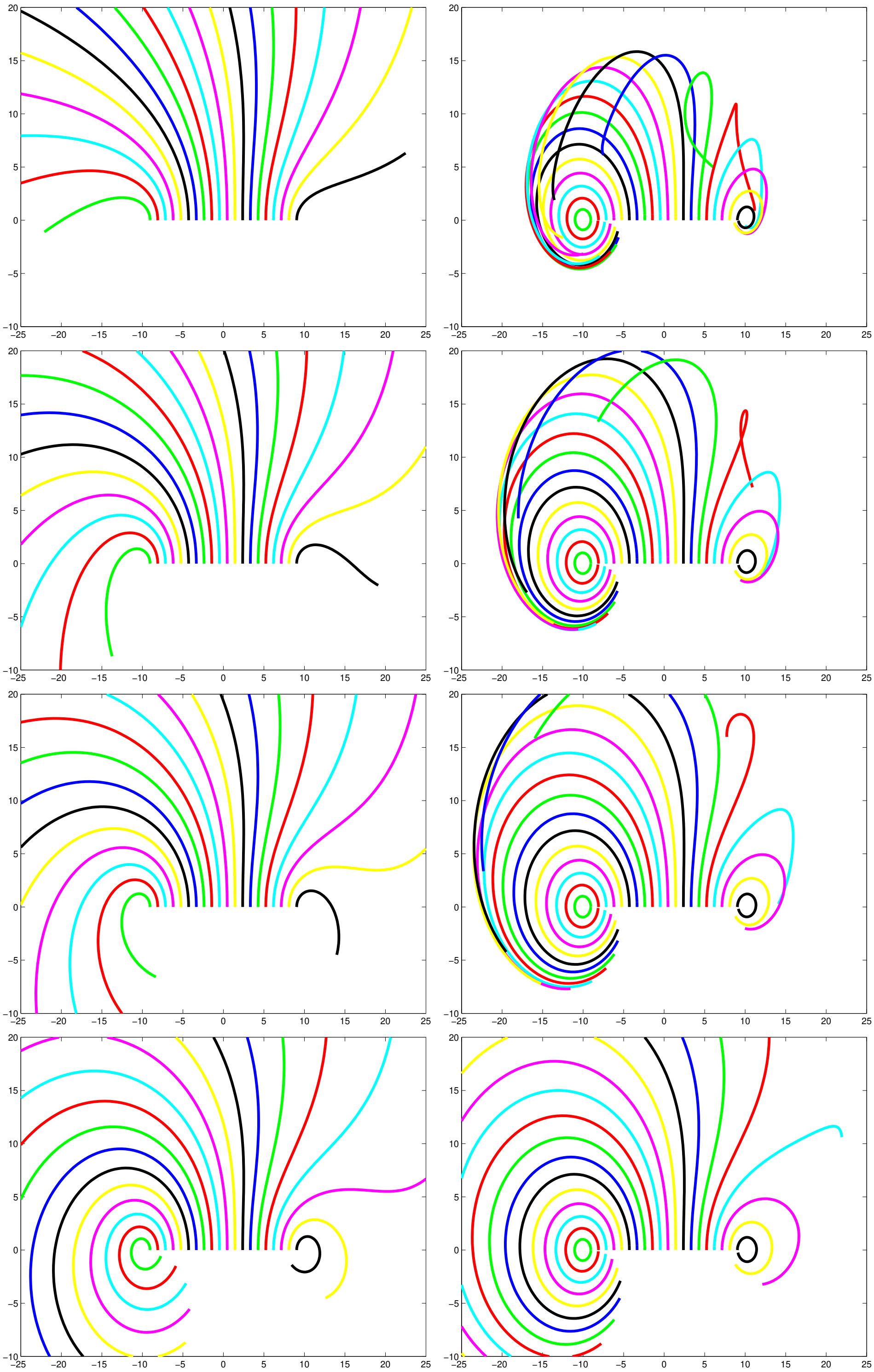

Figure 5.2: **First scheme (on the left) versus second scheme (on the right).** From top to bottom: discretization levels of 3, 5, 7 and 20. As in Fig. 5.1, various trajectories are displayed, these trajectories being obtained when the two opposite rotations see their angle increase progressively between 0 and 2π. Here, the rotation on the left has a larger relative weight than that on the right, which lessens the influence of the latter.

and registration of medical images. Thanks to the generic nature of its programming style, it was the ideal choice to develop our new approach quickly.

The polyrigid transformations were implemented as a new transformation class: `PolyRigidTransform<TScalarType,Dimension,Dimension>`. It is templated by the dimension of the space, and thus can be used in both 2D and 3D applications. Its testing and the development of related registration algorithms were greatly facilitated by ITK, since it provides many tools that can be applied to *any* ITK transformation.

Fig. 5.3 shows the registration framework chosen by ITK. The experiments presented in next section are carried out using our new class of transformation, the SSD similarity criterion (called here a "metric"), with a bilinear interpolation.

For the first-order gradient descent, we used the already implemented ITK optimizer `itk::RegularStepGradientDescentOptimizer`, in which the step of the gradient is reduced if the change of direction is too abrupt. This prevents the algorithm from going systematically too far in the direction of the gradient.

For the second-order gradient descent, we have implemented our own optimizer. This enabled us to adapt completely the optimization to the registration strategy studied in Section 5.4. Fig. 5.4 presents the new ITK classes designed to this effect. In order to take into account the information given by an approximation of the Hessian, we chose to modify the `itk::ImageToImageMetric` class, which provides access only to the first derivative of the similarity measure. Other classes were also modified, so as to perform a registration procedure making use of the Hessian, which is handled by the new class `ImageRegistrationMethodWithHessian`.

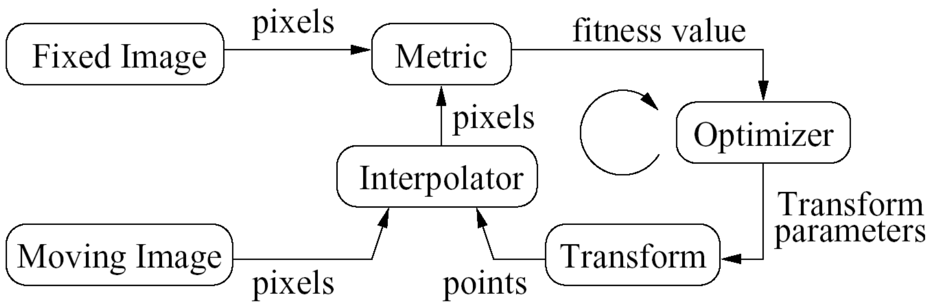

Figure 5.3: ITK's registration framework.

5.4 Registration of Histological Slices

5.4.1 Object of the Study and Experimental Setup

In order to demonstrate the feasibility and power of polyrigid transformations for registration purposes, we present in this section some preliminary results on the registration of histological slices (Fig. 5.5). These images are acquired in such a way that locally rigid or even affine deformations are frequently introduced locally during the acquisition process. E.g., a gyrus has been rotated in the top left corner in Fig. 5.5. The aim of this study is to show that simple polyrigid transformations can substantially and naturally reduce the impact of such non-linear deformations, while preserving the anatomical differences, i.e. whithout introducing unrealistic deformations.

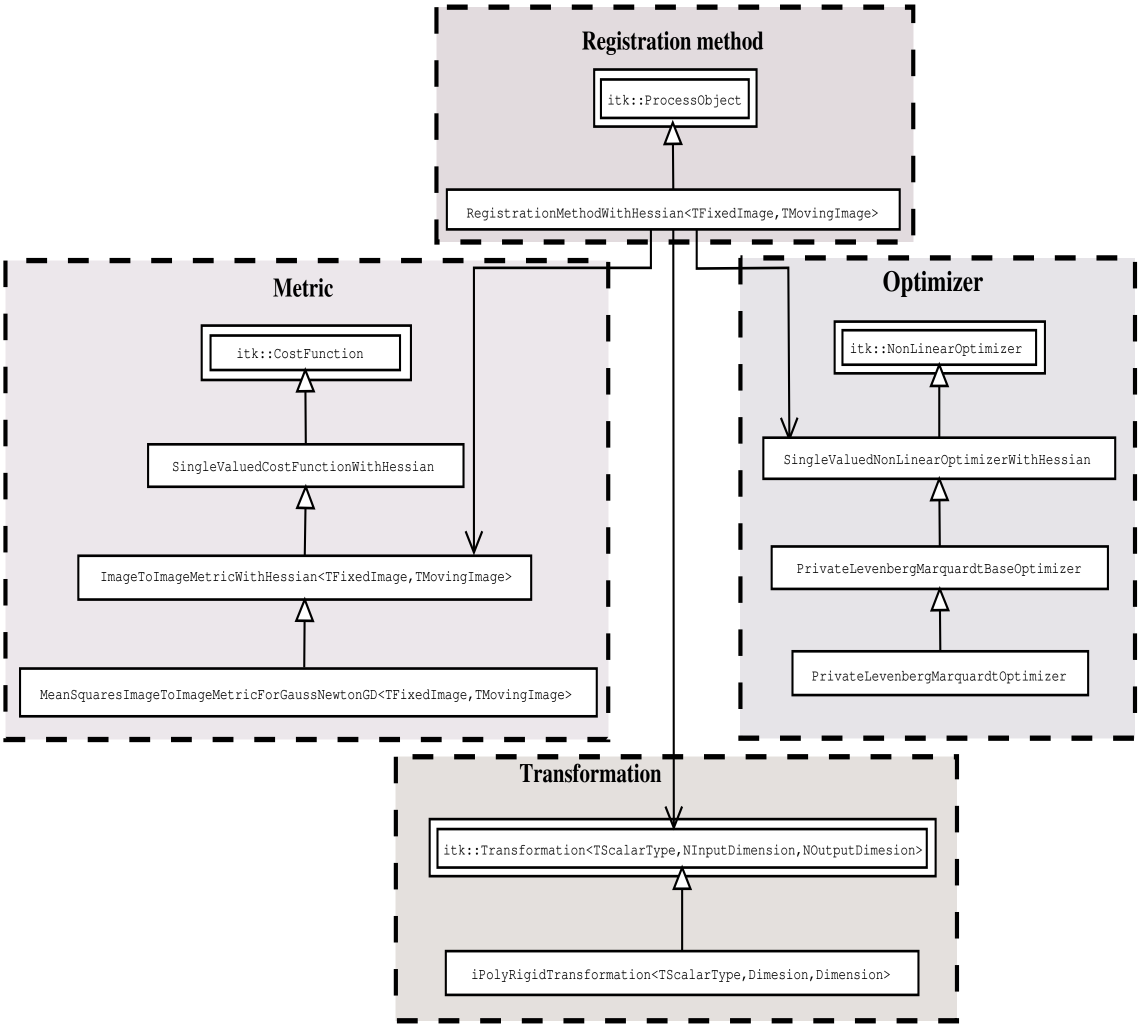

Figure 5.4: **Implemented classes (boxes with a single rectangle) and their relations to each other and existing ITK classes (boxes with two nested rectangles)**. A hollow triangle at the end of an arrow stands for inheritance and and simple lines for dependence (conventions of the UML 1.3 standard). The classes belong to four different "families": that of the registration method, metric, optimizer and transformation. For more clarity, these groups have been put into dashed boxes.

The slices studied in this paper have been kindly provided by P. Thompson and A. Pitiot from the LONI (UCLA), and are consecutive myelin-stained histological sections (or autoradiographs). Stemming from a human brain, it is during three steps that the artificial deformations of a locally rigid nature are introduced. These steps are the cutting process, the successive chemical treatments, and the glass mounting step. The dimensions of the slices are 226 by 384 pixels. The registration of these slices has an additional difficulty: the absence of matter in the lower-left-hand corner of the second slice. Most non-rigid registration algorithms are misled by such a defect because they will try to correct it, and in so doing they introduce irrelevant artificial deformations. During the acquisition process, the calibration of the optical setup remained unchanged. Therefore, the assumption that the various structures present in the images have the same grey level is valid, and we can safely use the sum of square differences criterion.

In figure 5.5, we see the results obtained with classical robust rigid and affine registration procedures [Ourselin 01b]. These methods are not able to register properly the rotated gyrus and at the same time all other gyri. This defect is due to the lack of degrees of freedom in these linear transformations. In the affine case, it is also due to the fact that the extra degrees of freedom, modeling dilatations and shearing, are not used to model the actual deformations appearing in the image. This suggests to use transformations with more degrees of freedom, and if possible, degrees of freedom that are adapted to the real deformations observed. This is precisely what polyrigid transformations are aiming at for this application.

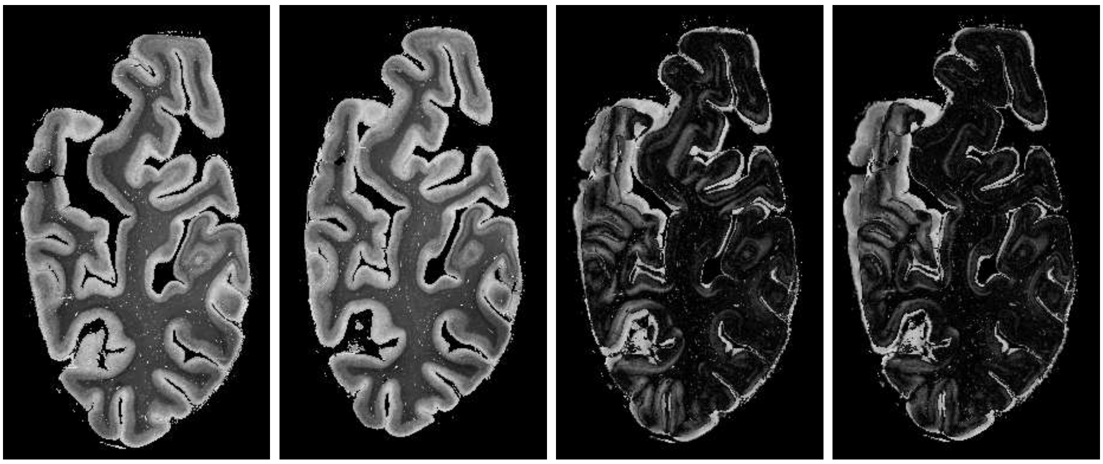

Figure 5.5: **Histological slices (on the left) and Images of absolute differences for affine (third starting from the left) and rigid (on the right) registrations**. The whiter the grey level, the worse the registration is locally. We see in both cases that in many places, the edges of the gyri have not been registered precisely, because of the influence of the rotated gyrus in the top left corner and also because some other (smaller) non-linear deformations have taken place. We see also on the left that in order to register better the rotated gyrus, the affine registration gives poorer results for many edges than the rigid registration. Indeed, this better registration of the gyrus has been obtained at the cost of a dilatation of the slice, which in this situation is not appropriate.

During the experiments, the initialization used is the following:

- All rigid components are initially set to the identity.
- Anchor points are sampled on a regular grid, except in the first experience, where a manual initialization is done.

- Initial relative weights are all equal.
- (σ_i) are initialized at a high value, here 40, so that the influence of all regions extends in a good half of the images.

Four rigid components are used in the experiments. This number is a good compromise between the necessity of having enough degrees of freedom to register correctly the slices and the obvious risk of introducing too many degrees of freedom, which results in large unrealistic deformations. This is precisely what occurs when more components are used. All in all, these four components are parameterized by 27 scalar parameters, which is the equivalent of only 6 control points for the B-splines (24 scalar parameters).

The second numerical scheme is used here, since it outperforms the other. The level of discretization is chosen very low, i.e. almost all results are obtained using no intermediate point between the starting position and the final position of a point. The deformed grids shown in the figures of this section show that the obtained transformations are invertible. Since inverting them was not necessary in this study, it is not necessary here to use more points. In fact, increasing the number of discretization points used leads to very similar results, which shows that such a precision is not necessary here. But for other applications, if discontinuities appear or if it is necessary to use also the inverse transformation, then a finer discretization is of course essential.

5.4.2 Limitations of the First-Order Gradient Descent.

A simple way to minimize the similarity criterion between the images is to use a first-order gradient descent, i.e. to make the parameters evolve in the steepest direction of descent, which is given by the gradient. Unfortunately, this approach cannot be directly used for our model. The partial derivatives in the gradient show differences of several orders of magnitudes! Qualitatively we have $\|\frac{\partial}{\partial r_i} T_p\| >> \|\frac{\partial}{\partial t_i} T_p\| >>$ other derivatives. This implies that the classical gradient descent will make rotations evolve enormously, the translations a little, and the other parameters almost not.

For rotations and translations, the difference of magnitude of their respective partial derivative can be intuitively understood in the following way: for a small variation of the angle of rotation, points far away from the center of rotation will move proportionally to their distance to the center of rotation. In other words, the further away from the center, the higher will be the variation in position, and a small variation can result in a large one at a distance. On the other hand, a variation in translation will affect all the points uniformly, and a small variation always yields a small modification in position. Therefore, we tend to have large partial derivative with respect to rotations as compared to partial derivatives with respect to translations. This difficulty is often encountered in situations where parameters of different natures are to be optimized simultaneously.

A simple remedy is to renormalize the amplitudes of the partial derivatives. Typically, dividing the amplitude of the rotation partial derivative by a factor 1000 is needed to obtain the optimization at least of both rotations and translations. Fig. 5.6 shows the behavior of the registration as the scaling evolves.

As we see in the deformed grid of Fig. 5.7, the final transformation is notably non-linear. But the anchor points have not moved from their initial position, which does not allow for an accurate registration in the upper left-hand corner. We can see in Fig. 5.8 that the edges were much better registered than with using a robust rigid transformation. But the incapacity to optimize the regions of influence thwarted the better registration of the upper-left-hand corner.

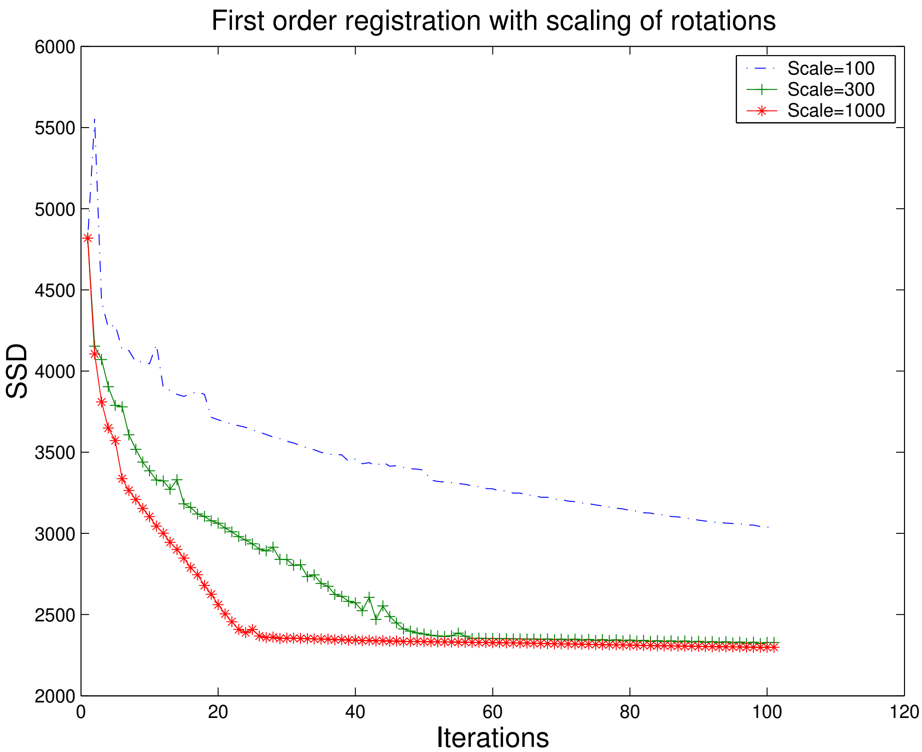

Figure 5.6: **SSD criterion evolution for a polyrigid registration with a simple first order gradient descent.** The only modification to the gradient was the rescaling of the rotation partial derivative, which is much larger in magnitude than the others. The figure shows the SSD evolution during registration for three values of the rotation scaling: 100, 300 and 1000. Thus, we see that an important rescaling (at least of a factor 300) is necessary to improve the registration process, which is otherwise inefficient. The registration results only in the optimization of the rotations, the other parameters hardly evolving during the registration procedure.

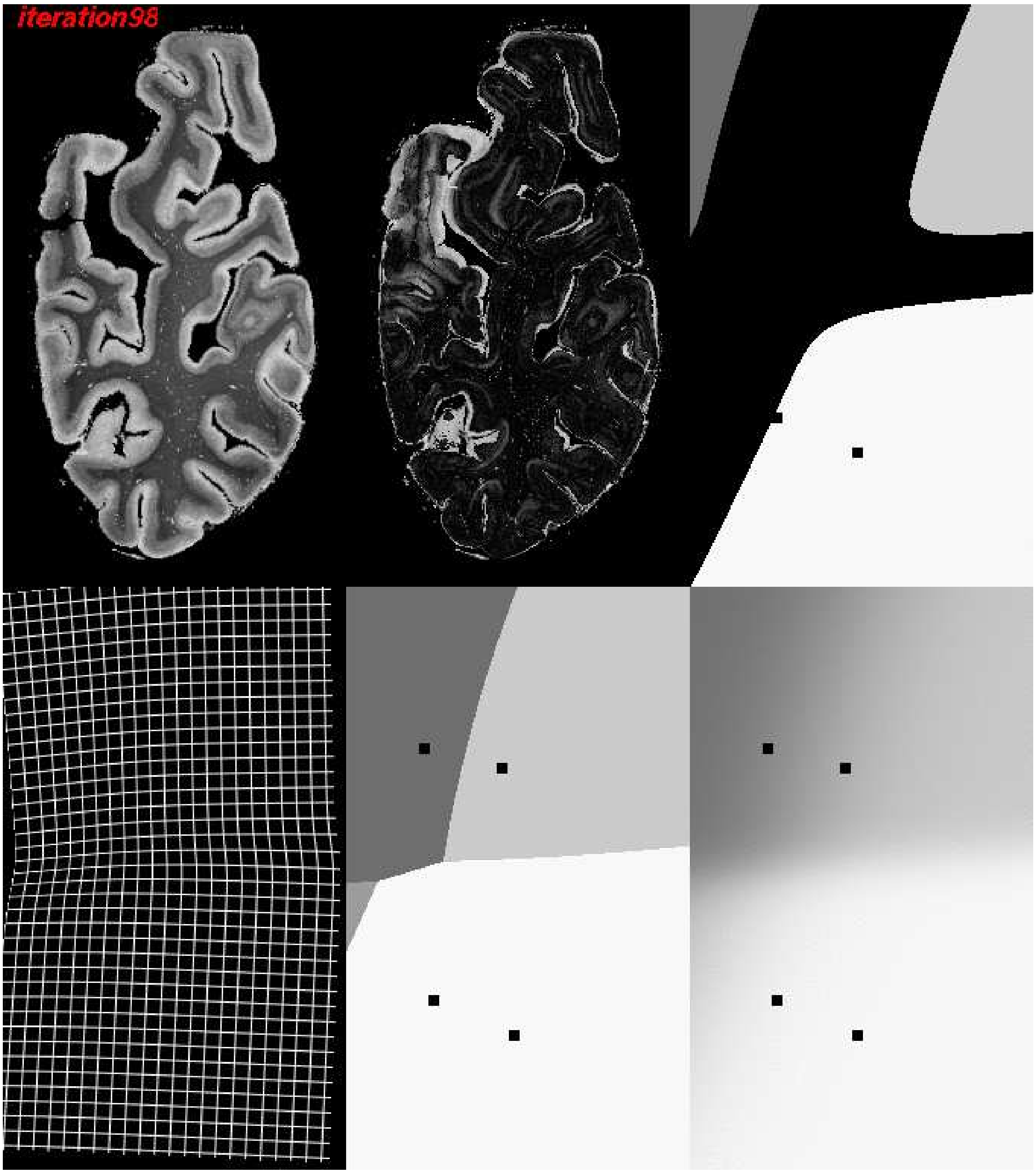

Figure 5.7: **Polyrigid registration result with a simple first-order gradient descent.** From left to right and from top to bottom: (1) The deformed image. (2) The image of absolute difference between the deformed image and the fixed image. (3) A representation of the regions of influence: a grey level is attributed to each region, and this color is displayed if and only if the local weight of the region represents more than 90 percents of the total weight. The anchor points are represented here by small squares. (4) A regular grid deformed like the deformed image. (5) An image of the regions of influence, a grey level being displayed if and only if its associated weight is the largest one. (6) An image of the regions of influence displaying at each point the weighted average of the grey levels according to the local weights. Thus, we see that a non-linear deformation has been obtained, as show the curved lines of the initially regular grid. The defect of this registration is that anchor points have not moved from their initial positions.

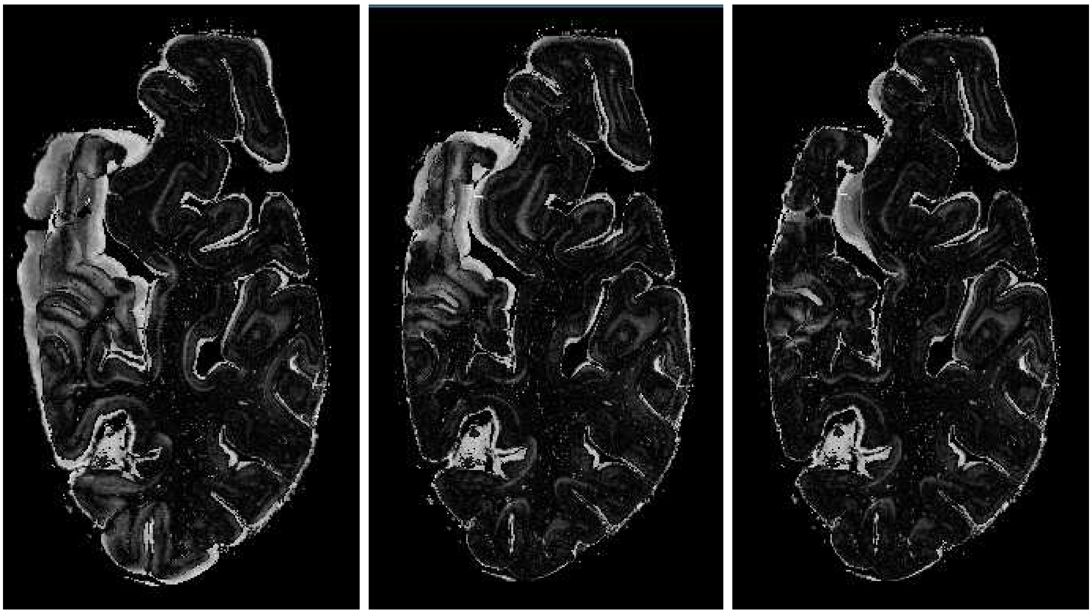

Figure 5.8: **Rigid (on the left) versus polyrigid registration with a simple first-order gradient descent and a rescaling for rotations (on the middle) and Levenberg-Marquardt second-order descent(on the right).** This demonstrates that the absence of matter in the lower left-hand corner has thwarted the polyrigid registration algorithm. In the case of the simple first-order gradient descent, most edges have been very finely registered, much better than in the rigid case. Nonetheless, we see that the gyrus lying in the upper left-hand corner still has not been completely correctly registered, because of the incapacity of the algorithm to optimize the region parameters, which have small derivatives in magnitude. On the contrary, on the right, the Levenberg-Marquardt method has allowed the algorithm to register the previously rotated gyrus. The result is much better qualitatively than for the first order descent, and edges are much more finely registered than in the rigid case. However, some amount of unnatural deformations has been added at the vertical frontier between the gyrus and the rest of the slice. This phenomenon is due to the simple forms of the respective rigid regions

Of course, one could think of estimating a relevant renormalization for each type of parameter. This could be done by computing some kind of average amplitude for each partial derivative, and them dividing the derivatives by that value so as to obtain values of approximately the same amplitude. But this renormalization would have to be carried out for each pair of images to be registered. It would surely not be efficient for all iterations, and it is not so clear why all partial derivatives should have approximately the same amplitude. In the case of a pure translation, forcing the rotation vectors to evolve would not be convenient! This calls for some type of adaptive renormalizing method.

5.4.3 Registration Results using a Levenberg-Marquardt Algorithm

To renormalize the partial derivatives in an adaptive way, a simple idea is to perform a second-order gradient descent scheme. The renormalization is handled by multiplying the gradient by the inverse of a matrix reflecting the second variations of the criterion. Here, this matrix is the approximation of the Hessian described in Section 5.3. The computation of this positive matrix term (in the sense of quadratic forms) can be done only at the expense of a very little cost, since it only requires the knowledge of the transformation's gradient and of the images intensities.

In order to perform an efficient 2nd-order gradient descent, the Levenberg-Marquardt algorithm (LM) was used (see [Bazaraa 93], pages 312-314). At each iteration, a trust indicator is updated, which tunes the gradient descent between a simple first-order gradient descent and a quasi Gauss-Newton descent based on the truncated Hessian. This way, we obtain naturally a renormalization of the various parameters and also a faster convergence, especially when we are close to the minimum.

Figures 5.8 and 5.9 show that the Levenberg-Marquardt performs much better than a first-order descent, both quantitatively and qualitatively. Three major local rigid transformations have been correctly identified. The edges have been very finely registered as compared to rigid registration, as we see in Figure 5.8. This good result is obtained in spite of a very crude initialization which proves the robustness of the proposed registration algorithm. The only remaining problem is the large deformations occurring at the vertical frontier between the originally rotated gyrus and the other gyri. This is partly due to the simple spherical form chosen for the regions of influence, and partly to the discontinuity that originally made the gyrus rotate. The polyrigid deformations are smooth transformations and therefore they cannot properly model discontinuous deformations.

5.4.4 Alternating Optimization

The renormalizing process via a second-order approach can be avoided by simply optimizing the parameters alternatively. Moreover, with more than 4 rigid components, the renormalization introduced in the second-order descent is no longer sufficient: the same defects as in the first-order gradient descent appear again.

As a consequence, we introduce here a strategy optimizing alternatively the various parameters. There is no single way of optimizing alternatively the parameters, and it is theoretically difficult to decide which parameters to group, and how many iterations of optimization are to be used for each group at each iteration of the global optimization. Our tests led us to optimize on the one hand the deformation parameters and on the other the region parameters, one iteration at a time for each. We also use here a Levenberg-Marquardt strategy for each group, to speed up the convergence. This yields a stable and efficient optimization algorithm.

Fig. 5.10 shows the result of the registration. We can clearly see that the registration process has identified and satisfactorily estimated at least three independent rigid behavior. At

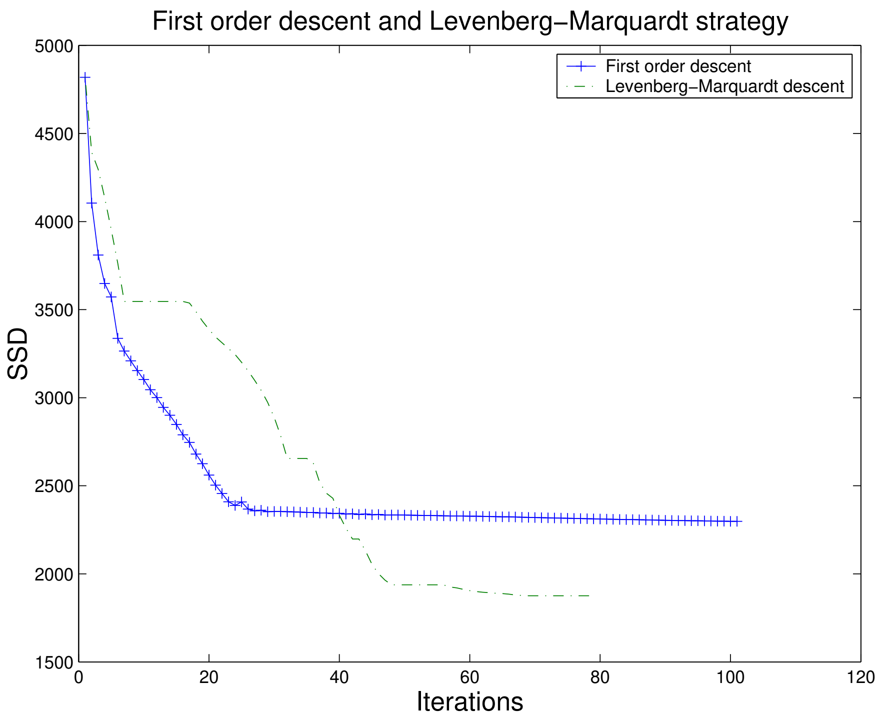

Figure 5.9: **SSD criterion evolution for a polyrigid registration with a Levenberg-Marquardt (LM) versus a simple first-order optimization scheme.** This shows that using a second-order descent has greatly enhanced the final results quantitatively, and also qualitatively as is shown is the next figures.

the same time, the deliberate simplicity of the regions of influence forbids a precise description of the frontiers between the regions. At this point of the registration process, we could resort to a classical non-rigid registration algorithm to make the registration more precise in this sector. But more simply, we can make use of the flexibility of polyrigid transformations by refining the parameters describing the regions of influence, as is shown in the next section of this report.

5.5 Results with more Complex Regions

5.5.1 The Shape of the Regions of Influence.

The assumption that each fuzzy region can be accurately described by a simple Gaussian weight can be too strong in certain cases. But generally speaking, we can be very flexible with the weights because the only limitation is to keep the weights (strictly) positive and smooth with respect to spatial coordinates and parameters. Therefore very complex regions can be used, the simplest way being to use mixtures of simple probability distributions. But other solutions could be used, such as introducing explicitly an pre-defined shape for a region. More precisely, if R is a region, we can define an associated weight with $w(x) = \mathbf{1}_R \star G_\sigma(x)$. $\mathbf{1}_R$ is simply the function returning 1 if $x \in G$ and 0 elsewhere. G_σ is a Gaussian of standard deviation σ, that smoothes $\mathbf{1}_R$ through a convolution. Thus, combinations of pre-defined regions and simply parameterized regions provides quite a rich framework for modeling an application-specific polyrigid transformation.

We present here preliminary results in which we have simply increased the number of anchor points per region. Therefore, regions are modeled via a mixture of Gaussian. This more general form for the weights $w_i(x)$ can be written as follows:

$$w_i(x) = p_i \sum_{j=1}^{n_i} G_{a_i^j, \sigma_i^j}(x).$$

In other terms, each component i has its own number n_i of anchor points $(a_i^j)_{j \in 1 \cdots n_i}$, which all have their specific standard deviation σ_i^j.

5.5.2 Results Obtained with Three Anchor Points

In order to see whether we can obtain better results than in the previous section, we present here the results with three anchor points per region. One could think of refining progressively the number of points, and this is a issue that will be addressed in future work. The present experiment simply consists in making the whole registration proceed with three anchor points, using the most efficient optimization algorithm presented in this report, i.e. the alternating LM strategy.

The experimental setup is identical, except for anchor points, which are initialized on the vertices of equilateral triangles placed on a regular grid.

We obtain here much better results, as show Fig. 5.11. The frontier that was lacking in precision is substantially refined here, introducing less artificial deformations. However, some amount of unrealistic deformation remains. That was to be expected, since it was because of a rift that the gyrus rotated. To proceed further, it would be necessary to make a distinction between the empty background and matter. A possibility would be to add this knowledge in the weights defining the influence of each region, for instance with a geodesic distance. The weight of a region would then be all the stronger as the current point is close in some geodesic

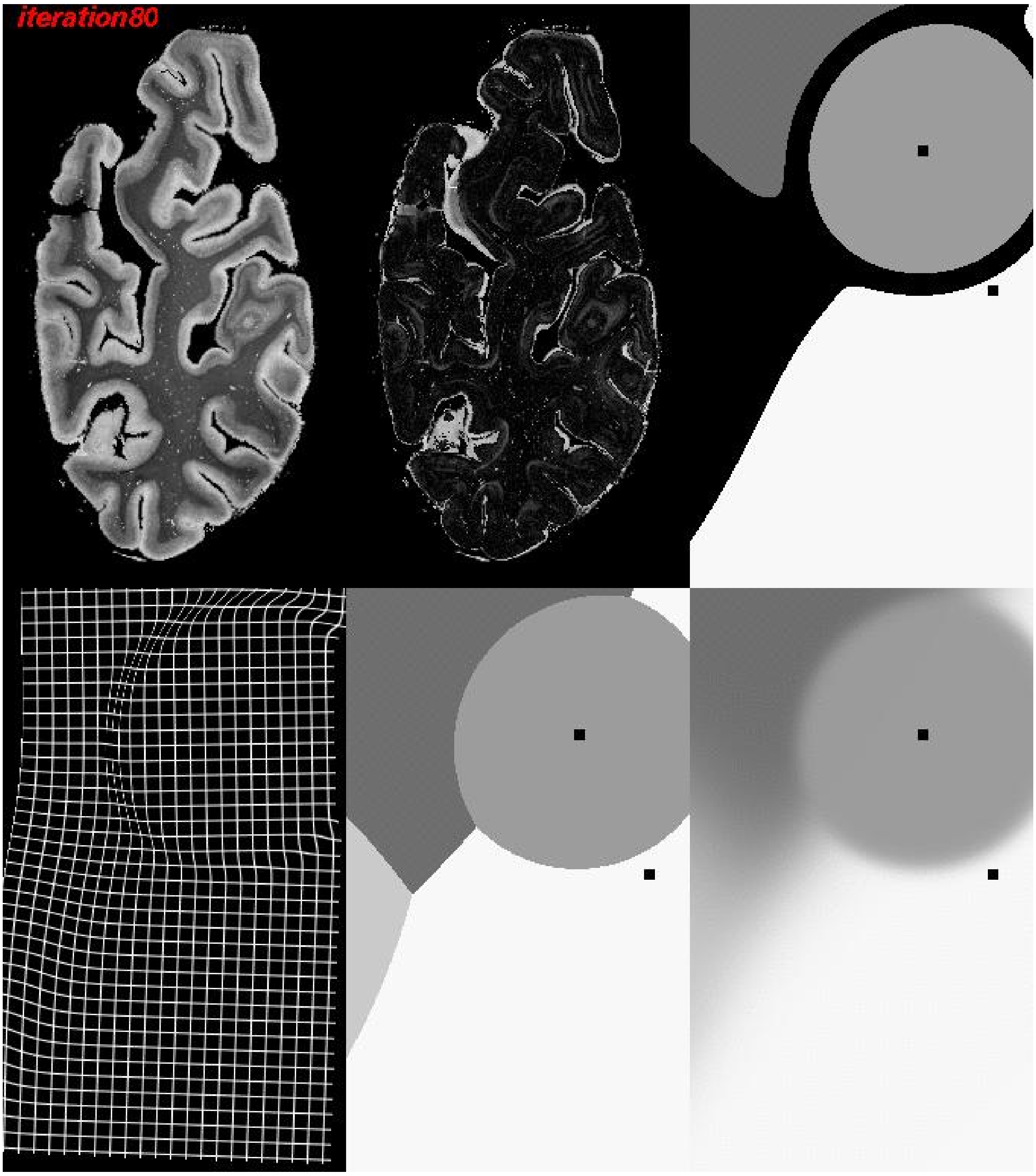

Figure 5.10: **Polyrigid alternating LM registration.** The gyrus has been as correctly registered as can be. Due to the Gaussian model, the vertical frontier on the left of it has a circular form, which results in some unnatural deformations. These deformations are marginal but nonetheless non-negligible. However, only 4 rigid components (i.e. 27 scalar parameters) have been necessary here to register very finely most of the slices, without being disturbed by the lack of matter in the lower-left-hand corner of one the registered slice.

sense. This would result in a different smoothing. The influence of a region close spatially but separated from the current point by a rift of empty background would be seriously lessened. As a consequence, the frontiers between the regions would be more realistic, since they would bear more resemblance with the rifts where they appear.

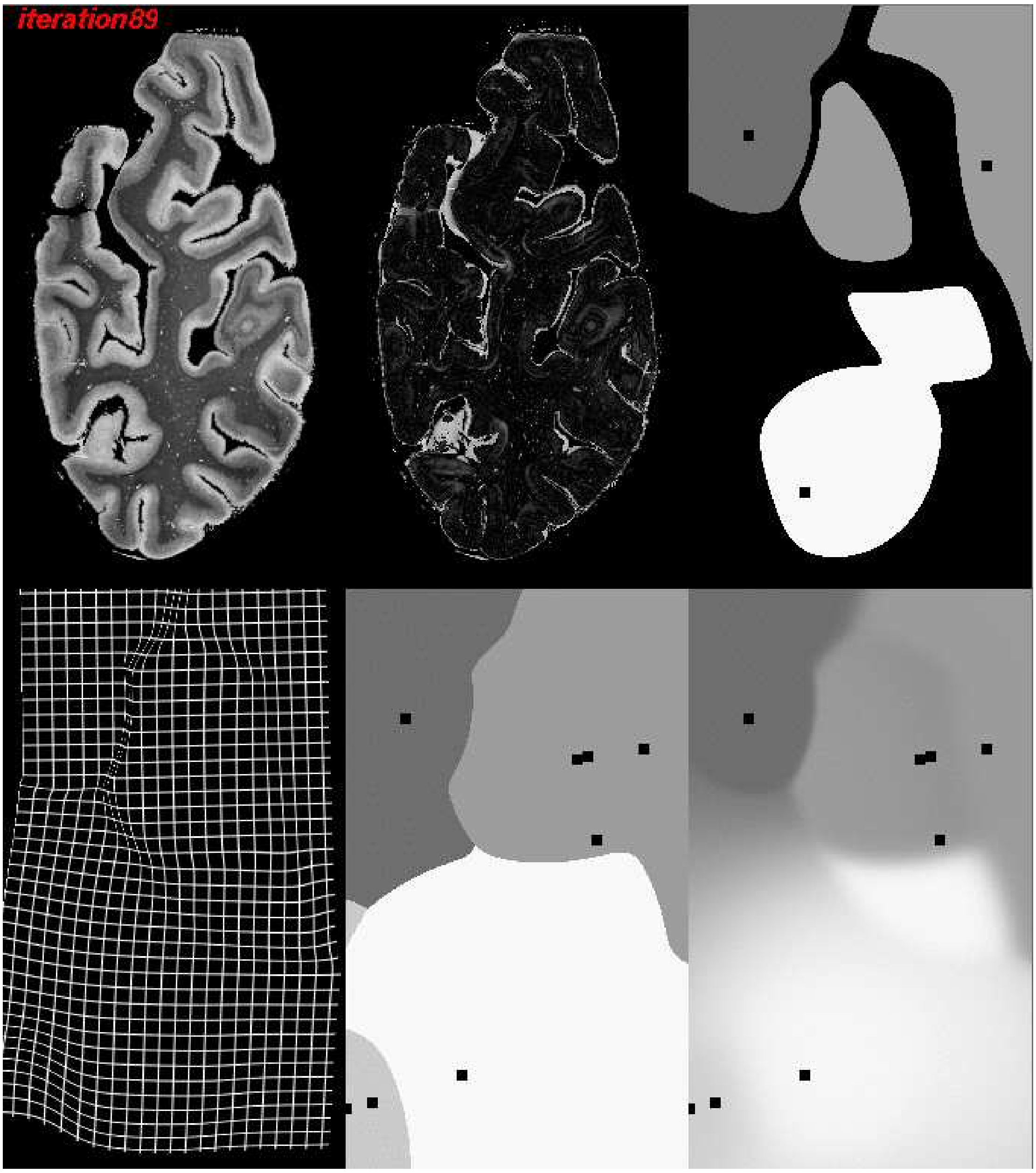

Figure 5.11: **Polyrigid alternating LM registration with 3 anchors points per region.** The result is quite satisfactory: thanks to the Gaussian mixture model, a realistic frontier has been automatically inferred which brings the originally rotated gyrus into a precise registration. All edges have been correctly registered. Few artificial deformations are introduced, thanks to the fact that we have only used four different regions having independent rigid motions. As the deformations of the regular grid show, the transformation is still invertible. It should also be noted that this result has been obtained with a fully automatic and crude initialization, and without resorting to a multi-resolution framework. This demonstrates the robustness of the registration algorithm.

5.6 Conclusions and Perspectives

Conclusions. We have presented in this Chapter a novel and innovative type of geometrical transformations: polyrigid and polyaffine transformations. These transformations have several rigid or affine components, which means that a given number of fuzzy regions are defined, on which the global transformation is mostly rigid or affine. The parameters coding the transformation are simple and intuitive, and provide a compact representation for locally rigid or affine movements.

In a rigorous mathematical framework, we have shown that these transformations are smooth and invertible. We have designed a new and efficient numerical scheme for the practical implementation in any dimension in the polyrigid case, and devise a complete optimization strategy for its application to non-rigid registration (Section 5.4).

Polyrigid transformations are exemplified successfully on the 2D registration of histological slices. Most non-linear artifacts generated during the acquisition process of the slices have been corrected, and it remains only a residual deformation due to the smoothness of polyrigid transformations. For this specific application, further developments would be needed to model the tearing process that has taken place, which is discontinuous by nature.

As shown in Section 5.5, there are many ways of adapting the polyrigid transformations to new applications, by modifying the shape and parameterization of the regions of influence. In order to make the polyrigid transformations more accurate, it should also be possible to define adaptive strategies progressively refine the shape of regions where it is necessary.

Perspectives. We have presented in Section 5.2 the extension of our framework to polyaffine transformations. We believe it is possible to use such an extension in the field of shape statistics. More precisely, one could model the variability of the shape around its mean via the statistical analysis of these variations in a certain space of transformations. By choosing as adequately as possible this space of deformations, a model with a limited number of parameters could be derived. Polyaffine transformations are in our opinion a good candidate for doing so, because they can take into account both local rotations, translations or swellings.

In the same vein, another application would be the building of new anatomical atlases, in the case of dataset presenting rigid by part deformations or simple local deformations. Using adapted transformations to establish correspondences between the various instances would surely lead to more accurate results. In the case of local swellings or shearings, it would be interesting to compare the performances of these new transformations to those obtained for example with B-Splines, for an equal number of degrees of freedom.

Acknowledgments

We are very grateful to P. Thompson, A. Toga, J. Annese and A. Pitiot for providing us with the histological slices through an associated teams collaboration between Epidaure at INRIA and LONI at UCLA.

5.7 Appendix: First Derivatives of Original Polyrigid Transformations

In this Appendix, we focus in 3D on the first derivatives of the second numerical scheme which can be used to discretize our original polyrigid transformations. This scheme was introduced in Section 5.3.1.

Differentiation with Respect to Parameters

Let us denote:

$$M_i^{\frac{1}{N}}(x,s) = \frac{1}{N}t_i + (e^{\frac{A_i}{N}} - Id)(x - st_i).$$

This is the modification "proposed" by the i-th component at a given time s and point x for the second scheme. Conversely, let us write the real modification:

$$M^{\frac{1}{N}}(x,s) = \frac{\sum_i w_i(x)(\frac{1}{N}t_i + (e^{\frac{A_i}{N}} - Id)(x - st_i))}{\sum_i w_i(x)}.$$

Then let p_i be a parameter of a rigid transformation T_p, and more specifically a parameter of the i-th component. When we compute the derivative of $T_2^{1/N}(x,s)$ with respect to p_i, we get the following simplification:

$$\begin{aligned}
\frac{\partial T_2^{1/N}(x,s)}{\partial p_i} &= \frac{\partial}{\partial p_i}\frac{\sum_j w_j(x)\left(\frac{1}{N}t_j + (e^{\frac{A_j}{N}} - Id)(x - st_j)\right)}{\sum_j w_j(x)} \\
&= \frac{\frac{\partial w_i}{\partial p_i}(x)M_i^{\frac{1}{N}}(x,s)}{\sum_j w_j(x)} + \frac{w_i(x)\frac{\partial}{\partial p_i}M_i^{\frac{1}{N}}(x,s)}{\sum_j w_j(x)} \\
&\quad - \left(\frac{\sum_j w_j(x)M_j^{\frac{1}{N}}(x,s)}{\sum_j w_j(x)}\right)\left(\frac{\frac{\partial}{\partial p_i}(\sum_j w_j(x))}{\sum_j w_j(x)}\right) \\
&= \frac{\frac{\partial w_i}{\partial p_i}(x)}{\sum_j w_j(x)}\left(M_i^{\frac{1}{N}}(x,s) - M^{\frac{1}{N}}(x,s)\right) + \frac{w_i(x)}{\sum_i w_i(x)}\frac{\partial}{\partial p_i}M_i^{\frac{1}{N}}(x,s).
\end{aligned}$$

Then, it only remains to see what form take the derivatives of the modifications and of the weights. If we assume that weights have a Gaussian expression as follows:

$$w_i(x) \quad = \quad \frac{p_i}{(2\pi\sigma_i^2)^{n/2}}\exp\left(-\frac{\|x-a_i\|^2}{2\sigma_i^2.}\right). \tag{5.9}$$

Then, we obtain:

$$\begin{cases}
\frac{\partial}{\partial a_i}(w_i(x)) &= -\frac{w_i(x)}{\sigma_i^2}(a_i - x)^T. \\
\frac{\partial}{\partial \sigma_i}(w_i(x)) &= (-n)\frac{p_i}{(2\pi)^{n/2}\sigma_i^{n+1}}\exp\left(-\frac{\|x-c_i\|^2}{2\sigma_i^2}\right). \\
&\quad +\frac{p_i}{(2\pi)^{n/2}\sigma_i^n}\frac{\|x-a_i\|^2}{\sigma_i^3}\exp\left(-\frac{\|x-a_i\|^2}{2\sigma_i^2}\right). \\
&= \left(-\frac{n}{\sigma_i} + \frac{\|x-a_i\|^2}{\sigma_i^3}\right)w_i(x). \\
\frac{\partial}{\partial p_i}(w_i(x)) &= \frac{1}{(2\pi\sigma_i^2)^{n/2}}\exp\left(-\frac{\|x-a_i\|^2}{2\sigma_i^2}\right).
\end{cases} \tag{5.10}$$

For derivatives of the modifications, we have:

$$\frac{\partial}{\partial t_i} M_i^{\frac{1}{N}}(x, s) = \frac{1}{N} Id - s \left(e^{\frac{A_i}{N}} - Id \right). \tag{5.11}$$

It remains to be seen how one can differentiate $(e^{\frac{A_i}{N}} - Id)$ with respect to the rotation vector r_i.

Derivation with Respect to the Rotation vector

The computation of the derivative of a matrix exponential of a matrix function has no simple form as in the case of scalars. Indeed, when we take $M(p) = \exp(A(p))$, we do not have in general $\frac{\partial}{\partial p} M(p) = \{\frac{\partial}{\partial p} A(p)\} M(p)$. This simply comes from the non-commutation of $A(p)$ and $\frac{\partial}{\partial p} A(p)$.

Let us denote B_x, B_y, B_z the following matrices:

$$B_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \ B_y = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \ B_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

We have the following result:

$$\forall a \in \{x,\ y,\ z\},\ \frac{\partial}{\partial r_a} \exp\left(\frac{1}{N} A\right) = \sum_{n>0} \frac{1}{n! N^n} \sum_{i=1}^{n} A^{i-1}.B_a.A^{n-i}.$$

This simply stems from the derivation of each term of the series defining the exponential.

Spatial Derivatives

Finally, let us consider the spatial derivative of our scheme, which one must also compute in order to obtain the derivative of the transformation with respect to its parameters. We have:

$$\begin{aligned} \frac{\partial T_p^{1/N}(x, s)}{\partial x} &= \frac{1}{N} \frac{\sum_i (M_i^{\frac{1}{N}}(x, s) \frac{\partial w_i(x)}{\partial x} + w_i(x) \frac{\partial}{\partial x} M_i^{\frac{1}{N}}(x, s))}{\sum_i w_i(x)} \\ &\quad - \frac{(\sum_i w_i(x) M_i^{\frac{1}{N}}(x, s))(\sum_i \frac{\partial w_i(x)}{\partial x})}{(\sum_i w_i(x))^2} \\ &= \frac{1}{N} \frac{\sum_i (M_i^{\frac{1}{N}}(x, s) \frac{\partial w_i(x)}{\partial x} + w_i(x)(e^{\frac{A_i}{N}} - Id))}{\sum_i w_i(x)} \\ &\quad - M^{\frac{1}{N}}(x, s) \frac{(\sum_i \frac{\partial w_i(x)}{\partial x})}{(\sum_i w_i(x))}. \end{aligned} \tag{5.12}$$

And finally, the spatial derivative of the weights is given by:

$$\begin{aligned} \frac{\partial w_i(x)}{\partial x} &= \frac{\partial}{\partial x} \left(\frac{p_i}{(2\pi\sigma_i^2)^{n/2}} \exp\left(-\frac{\|x - a_i\|^2}{2\sigma_i^2} \right) \right) \\ &= \frac{p_i}{(2\pi\sigma_i^2)^{n/2}} \left(-\frac{1}{\sigma_i^2}(x - a_i)^T \right) \exp\left(-\frac{\|x - a_i\|^2}{2\sigma_i^2} \right) \\ &= -\frac{w_i(x)}{\sigma_i^2}(x - a_i)^T. \end{aligned} \tag{5.13}$$

Chapter 6

Log-Euclidean Polyaffine Transformations

In this Chapter, we propose an alternative to the polyrigid and polyaffine transformations described in the previous Chapter. The novel framework presented here is called *Log-Euclidean polyaffine*, and overcomes the limitations of our original transformations in terms of *invariance properties*. The remarkable properties of Log-Euclidean polyaffine transformations (LEPTs) allow the fast computation of these transformations on regular grids.

Essentially, these nice results are obtained thanks to a much better *compatibility* between LEPTs and the *algebraic* properties of the Lie groups of rigid or affine transformations. Interestingly, this compatibility is obtained via the implicit use of a *Log-Euclidean* framework for linear transformations, which is the generalization to rigid or affine transformations of the framework we describe for tensors in Chapter 3.

Contents

6.1 Introduction **98**
6.2 A Log-Euclidean Polyaffine Framework **99**
6.2.1 Previous Polyaffine Framework 99
6.2.2 Simpler Velocity Vectors for Affine Transformations 103
6.2.3 Log-Euclidean Polyaffine Transformations 104
6.3 Fast Polyaffine Transform **107**
6.3.1 Matrix Exponential and the 'Scaling and Squaring' Method 111
6.3.2 A 'Scaling and Squaring' Method for LEPTs 111
6.3.3 2D Synthetic Experiments 114
6.3.4 3D Registration Example 122
6.4 A Log-Euclidean Framework for Rigid and Affine Transformations 125
6.4.1 Log-Euclidean Metrics 126
6.4.2 Invariance Properties 127
6.4.3 Regularization of Locally Affine Transformations 128
6.4.4 Log-Euclidean Framework for General Lie Groups 129
6.4.5 Numerical Implementation of Matrix Logarithm. 131
6.5 Conclusion and Perspectives **132**

Abstract. In the previous Chapter, we proposed a general framework called *polyaffine* to parameterize deformations with a finite number of rigid or affine components, while guaranteeing the invertibility of global deformations. However, this framework lacks some important properties: the inverse of a polyaffine transformation is not polyaffine in general, and the polyaffine fusion of affine components is not invariant with respect to a change of coordinate system. We present here a novel general framework, called *Log-Euclidean polyaffine*, which overcomes these defects.

We also detail a simple algorithm, the *Fast Polyaffine Transform*, which allows to compute very efficiently Log-Euclidean polyaffine transformations and their inverses on regular grids. The results presented here on real 3D locally affine registration of MR scans of the human brain suggest that our novel framework provides a general and efficient way of fusing local rigid or affine deformations into a global invertible transformation without introducing artifacts, independently of the way local deformations are first estimated.

Finally, we show in this Chapter that the Log-Euclidean polyaffine framework is implicitly based on a Log-Euclidean framework for rigid and affine transformations, which generalizes to linear transformations the Log-Euclidean framework we recently proposed for tensors, which is described in Chapter 3. We detail in this Chapter the properties of this novel framework, which allows a straightforward and efficient generalization to linear transformations of classical vectorial tools, with excellent theoretical properties. In particular, we propose here a simple generalization to locally rigid or affine deformations of a visco-elastic regularization energy used for dense transformations. Also, we briefly present the extension of the Log-Euclidean framework to any finite-dimensional Lie group, as well as its properties in this very general case.

Related Publications. Most of the material of Chapter was published in an INRIA research report [Arsigny 06a], and the results of this work were presented during the Third International Workshop on Biomedical Image Registration (WBIR'06) [Arsigny 06c]. The Log-Euclidean polyaffine framework and the Log-Euclidean framework for rigid and affine transformations were also used in [Commowick 06a, Commowick 06b] to propose an efficient and general 3D registration framework which allows to register local areas in the images using affine transformations having few degrees of freedom. This was exemplified in detail in two cases: bone registration in the lower abdomen area and critical brain structures segmentation.

6.1 Introduction

In this Chapter, we continue to focus on the parameterization of non-rigid geometrical deformations with a small number of flexible degrees of freedom. This type of parameterization is particularly well-adapted for example to the registration of articulated structures [Papademetris 05] and to the registration of histological slices [Pitiot 06, Arsigny 05c]. After a *global* affine (or rigid) alignment, this sort of parameterization also allows a finer *local* registration with *very smooth* transformations [Commowick 06a, Narayanan 05, Cuzol 05, Rueckert 99].

In Chapter 5, we parameterized deformations with a small number of *rigid or affine components*, which can model smoothly a large variety of local deformations. We provided a general framework to fuse these components into a global transformation, called *polyrigid* or *polyaffine*, whose *invertibility* is guaranteed. However, this framework lacks some important properties: the inverse of a polyaffine transformation is not polyaffine in general, and the polyaffine fusion of affine components is not invariant with respect to a change of coordinate system (i.e. is not *affine-invariant*). Here, we present a novel general framework to fuse rigid or affine components, called *Log-Euclidean polyaffine*, which overcomes these defects and yields transformations which can be very efficiently computed.

The sequel of this Chapter is organized as follows. First, we present the Log-Euclidean polyaffine framework and its intuitive properties. Then, we describe the *Fast Polyaffine Transform* (FPT), which is a numerical algorithm that allows to compute very efficiently Log-Euclidean polyaffine transformations (LEPTs) and their inverses on a regular grid. Afterward, we apply the FPT to a real 3D example, where affine components are estimated with the algorithm of [Commowick 06a, Commowick 06b]. *Without introducing artifacts*, our novel fusion ensures the invertibility of the global transformation. Then, we present the properties of the Log-Euclidean framework for rigid and affine transformations on which our polyaffine Log-Euclidean framework is implicitely based. This Log-Euclidean framework is the analogous of the framework we presented in Chapter 3 for tensors. Also, the extension of the Log-Euclidean framework to any finite-dimensional Lie group is presented, as well as its properties in this very general case.

6.2 A Log-Euclidean Polyaffine Framework

6.2.1 Previous Polyaffine Framework

Before presenting our novel polyaffine framework let us briefly recall the original polyaffine framework, described in detail in the previous Chapter. The idea is to define transformations that exhibit a locally affine behavior, with nice invertibility properties. Following the seminal work of [Little 96], we model here such transformations by a finite number N of affine *components*. Precisely, each component i consists of an affine transformation T_i and of a non-negative *weight function* $w_i(x)$ which models its spatial extension: the influence of the i^{th} component at point x is proportional to $w_i(x)$. Furthermore, we assume that for all x, $\sum_{i=1}^{N} w_i(x) = 1$, i.e. the weights are normalized.

Fusion of Displacements. In order to obtain a global transformation from several weighted components, the classical approach to fuse the N components simply consists in averaging the associated displacements according to the weights [Sheppard 68]:

$$T(x) = \sum_{i=1}^{N} w_i(x) T_i(x). \tag{6.1}$$

The transformation obtained using (6.1) is smooth, but this approach has one major drawback: although each component is invertible, the resulting global transformation is *not invertible* in general. To remedy this, we proposed in the previous Chapter to rely on the averaging of some *infinitesimal* displacements associated to each affine component instead. The resulting global transformation is obtained by integrating an Ordinary Differential Equation (ODE), which is computationally more expensive but guarantees its invertibility and also yields a simple form for its inverse. The nice invertibility properties of this approach are illustrated by Fig. 6.1.

Polyaffine Framework. The polyaffine approach can be decomposed into three steps:

- **Step 1: Associating Velocity Vectors to Affine Transformations.** For each component i, one defines a family of *velocity vector fields* $V_i(., s)$ parameterized by s, which is a time parameter varying continuously between 0 and 1. $V_i(., s)$ satisfy a

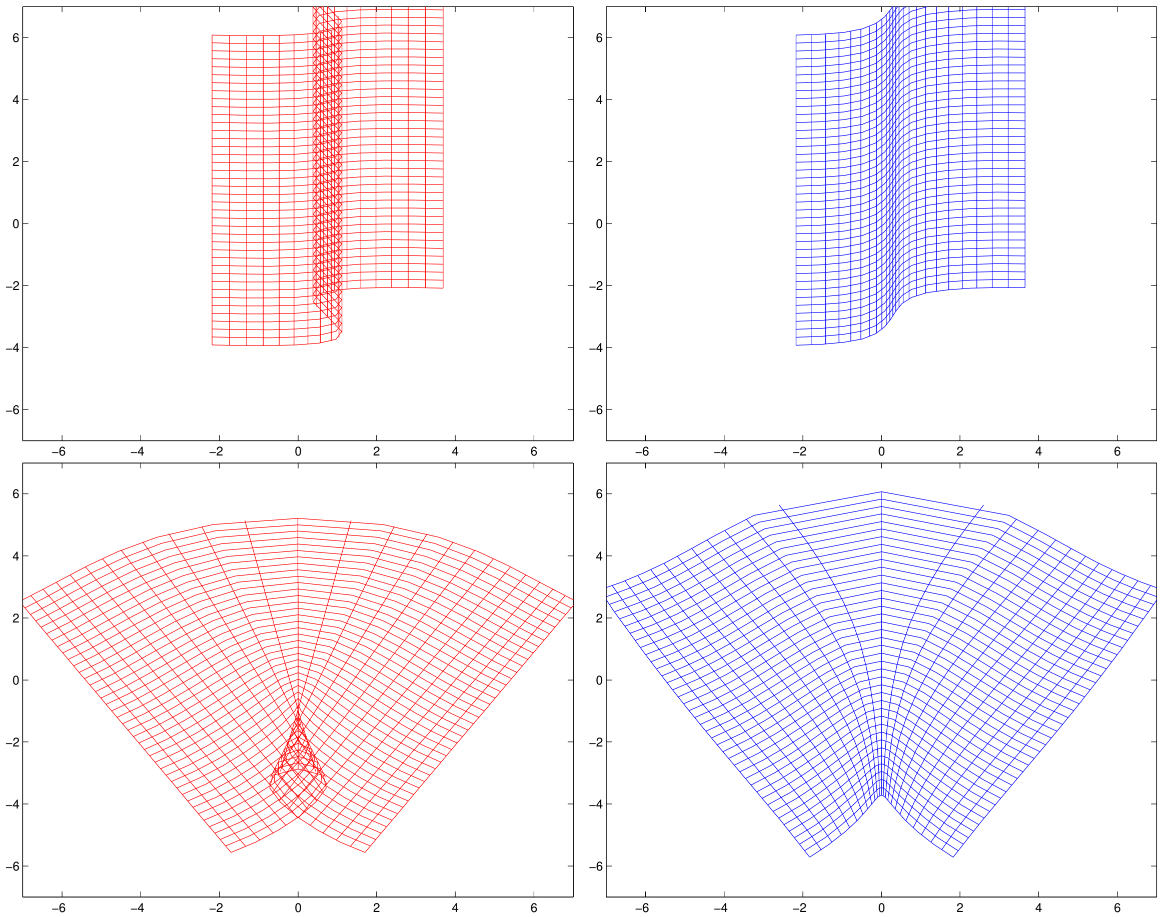

Figure 6.1: **Guaranteeing invertibility with infinitesimal fusion. Right (in blue):** regular grid deformed by the fusion of two affine transformations, using the direct averaging of displacements. **Left (in red):** regular grid deformed by the infinitesimal fusion of the transformations in the polyaffine framework. **Top:** two translations are fused. **Bottom:** two rotations of opposite angles are fused. Note how the regions of overlap disappear when infinitesimal fusion is used. The translations used were the following: $t_1 = (3, 1)^T$ and $t_2 = (-1.5, 3)^T$, and the two rotations of opposite angles of magnitude 0.63 radians where centered on $(-2, 0)$ and $(+2, 0)$. The fusion was carried out with the following weights (given here in unnormalized form): $w_i(x) = 1/(1 + ((x_1 - c_i)/\sigma)^2)$, where $c_1 = -2$, $c_2 = +2$ and $\sigma = 5$ (smooth transition between the two components).

consistency property with T_i: when integrated between time 0 and 1, they should give back the transformation T. Hence the following definition:

Definition 6.1. *The family of vector fields $V(.,s)$, where s belongs to $[0,1]$, is consistent with the transformation T if and only if its integration between time 0 and 1 gives back the transformation T:*

1. *for any initial condition x_0 one can integrate between 0 and 1 the differential equation $\dot{x} = V(x,s)$ so that $x(1)$ exists.*
2. *$x(1)$ is equal to $T(x_0)$.*

Several possible choices exist to associate velocity vector to affine transformations. One of the main contributions of this work is precisely to propose a novel choice for such speed vectors. Interestingly, we do not know at present how many other choices exist and whether they might have even better properties than the ones we have found so far.

- **Step 2: Fusing Velocity Vectors instead of Displacements.** The idea is then to average the vector fields $V_i(.,s)$ according to the weight functions $w_i(x)$ to define an ODE fusing the N components. Weight functions are very important and model the influence in space of each component. They controls in particular the sharpness of transitions between the fused affine transformations. Also, they can take into account the geometry of anatomical regions of interest, as will be the case in the experimental results on 3D MRI data given in the sequel.

  The Polyaffine ODE fusing velocity vectors according to weights functions is the infinitesimal analogous of (6.1) and writes:

$$\dot{x} = V(x,s) \stackrel{def}{=} \sum_i w_i(x) V_i(x,s). \tag{6.2}$$

- **Step3: Integration of the Polyaffine ODE.** In this infinitesimal framework, the value at point x_0 of the global transformation T fusing the N components is obtained via the integration of Eq. (6.2) between 0 and 1, with the initial condition $x(0) = x_0$. This principle was first illustrated in Chapter 2 by Fig. 2.1.

What Velocity Vectors for Affine Transformations at Step 1? Let us take an affine transformations $T = (M,t)$, where M is the linear part and t the translation. To define a family of velocity vector fields consistent with T, it was proposed in [Arsigny 05c] to rely on the *matrix logarithm* of the linear part M of T. More precisely, let L be the principal matrix logarithm of M. The family of speed vector fields $V(.,s)$ we associated to T writes:

$$V(x,s) = t + L(x - s\,t) \text{ for } s \in [0,1]. \tag{6.3}$$

Well-Definedness of the Principal Logarithm. One should note that using principal logarithms of the linear part of affine transformations at the first step of the polyaffine framework is not always possible.

The theoretical limitation implied by this particular choice of velocity vectors is the following: principal logarithms are not always well-defined. More precisely, the principal logarithm of an invertible matrix M is well-defined if and only if the (complex) eigenvalues of M do not lie on the (closed) half-line of negative real numbers [Cheng 01].

For rotations, this means quite intuitively that the amount of (local) rotation present in each of the components should be strictly below π radians in magnitude. This can be clearly seen in the domain of matrix logarithms, where this constraint corresponds to imposing that the imaginary part of eigenvalues be less then π in magnitude. Fig. 6.2 illustrates this general situation, which is not specific to rotations.

For general invertible linear transformations with positive determinant, the interpretation of this constraint on eigenvalues is not so clear, since rotational and non-rotational deformations are intertwined. However, one should note the closed half-line of negative number is a set of null (Lebesgue) measure in the complex plane, which indicates that very few linear transformations with positive determinant (corresponding to extremely large deformations) will not have a principal matrix logarithm. From a practical point of view, one can anyway just check whether the constraint is satisfied by computing numerically the eigenvalues of M, which only amounts to solving a third degree polynomial equation for 3D affine transformations.

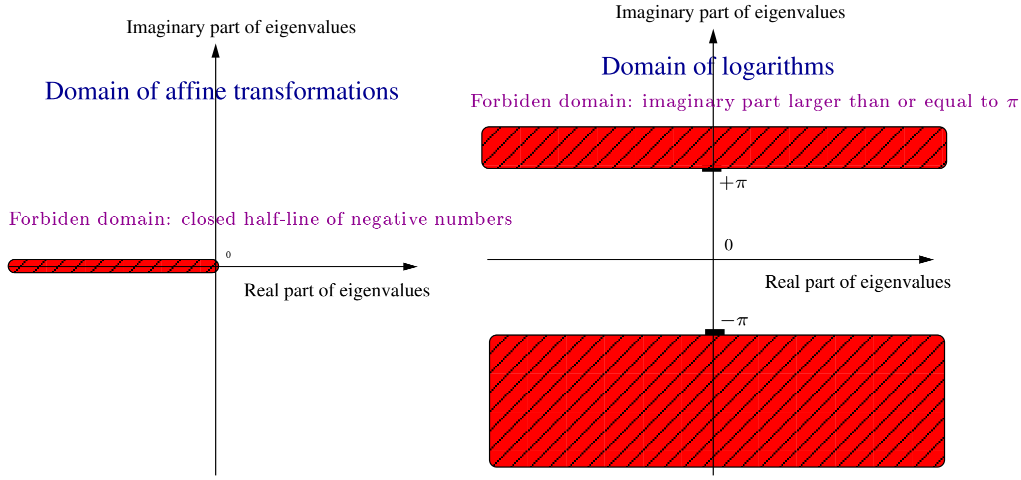

Figure 6.2: **Constraints imposed on affine transformations by the use of the principal matrix logarithm. Left:** only affine transformations whose (complex) eigenvalues do not lie on the (closed) half-line of negative real numbers have a principal logarithm and can be handled by our framework. Simplifying things a bit, this corresponds intuitively to imposing that (local) rotations be smaller in magnitude than π radians. This can be seen more clearly on the principal logarithms of these *admissible* affine transformations: the imaginary part of their eigenvalues must be smaller than π in magnitude. This is illustrated on the **right** on this figure. A more detailed discussion of this constraint is given in Subsection 6.2.1.

In the context of medical image registration, we do not believe this restriction to be problematic, since a global affine alignment of the images to be registered is always performed first. This factors out the largest rotations and it would be very surprising from an anatomical point of view to observe very large deformations (e.g., local rotations close to 180 degrees) of an anatomical structure from one individual to another after the anatomies of these individuals have already been affinely aligned.

Heavy Computational Burden at Step 3. Now, from a practical point of view, integrating the ODE given by Eq. (6.2) with the velocity vectors of Eq. (6.3) is quite computationally expensive, especially when one wishes to do this for all the points of a 3D regular grid, for example a $256 \times 256 \times 100$ grid, which is commonly in the case for T_1-weighted MR images. We will see in the rest of this section how one can drastically reduce this complexity by slightly modifying the speed vectors of Eq. (6.3).

6.2.2 Simpler Velocity Vectors for Affine Transformations

Let us see now how one can define much simpler velocity vectors for affine transformations than the ones given in Eq. (6.3). The basic idea is to rely on the logarithms of the transformations *themselves*, and not only on the logarithms of their *linear parts*. These logarithms can be defined in an abstract way in the context of the theory of Lie groups, as detailed in Chapter 2. Interestingly, thanks to the faithful representation of these transformations obtained with *homogeneous coordinates*, these logirithms can be computed in practice via matrix logarithms.

Details about our numerical implementation of the matrix logarithm are given in the Subsection 6.4.5 of the Appendix.

Homogeneous Coordinates. Homogeneous coordinates are a classical tool in Computer Vision. They are widely used to represent any n-dimensional affine transformation T by $(n+1)\times(n+1)$ matrix, written here $\tilde{T}$. Such a representation is called by mathematicians 'faithful' (in the sense of representation theory), which means that there is no loss of information in this representation. $\tilde{T}$ takes the following form:

$$T \sim \tilde{T} \overset{def}{=} \begin{pmatrix} M & t \\ 0 & 1 \end{pmatrix}, \tag{6.4}$$

where M is the linear part of T ($n \times n$ matrix) and t its translation. In this setting, points x of the ambient space are represented by $n+1$-dimensional vectors $\tilde{x}$, adding an extra '1' after their coordinates:

$$x \sim \tilde{x} \overset{def}{=} \begin{pmatrix} x \\ 1 \end{pmatrix}.$$

This way, the action of the affine transformation on a point x can be obtained simply in terms of matrix multiplication and is given by $\tilde{T}.\tilde{x}$.

Principal Logarithms of Affine Transformations. Using homogeneous coordinates, the principal logarithm of the affine transformations *themselves* can be computed in a simple way.

The main point here is that the principal logarithm of an affine transformation T is represented in homogeneous coordinates by the matrix logarithm of its representation $\tilde{T}$. This matrix logarithm takes the following form:

$$\log\left(\tilde{T}\right) = \begin{pmatrix} L & v \\ 0 & 0 \end{pmatrix},$$

where log stands for the principal matrix logarithm. L is an $n \times n$ matrix and v an n-dimensional vector. Exactly as in the former subsection, L is the principal matrix logarithm of M. But v is *not* equal in general to the translation t. Actually, the difference between our novel approach and the previous one resides essentially in this v.

Interestingly, the well-definedness of the principal logarithm of an affine transformation T is equivalent to the well-definedness of the principal logarithm of its linear part M. The reason for this is that the principal matrix logarithm of an invertible matrix is well-defined if and only if the imaginary parts of its (complex) eigenvalues of M do not lie on the (closed) half-line of negative real numbers [Cheng 01], as mentioned before. Because of the form taken by $\tilde{T}$ (see Eq. (6.4)), the spectrum of $\tilde{T}$ is exactly that of its linear part M plus an extra eigenvalue equal to 1. Hence the equivalence of the existence of both principal logarithms.

Simpler Velocity Vectors for Affine Transformations at Step 1 of the Polyaffine Framework. Using now principal logarithms of affine transformations instead of the principal logarithms of their linear parts, one can now associate to an affine transformation T a simpler family of velocity vector fields than in Eq. (6.3) in the following way:

$$V(x, s) = V(x) = v + L.x \text{ for } s \in [0, 1]. \tag{6.5}$$

What is remarkable here is that the velocity vector field at time s associated to T *does not depend on* s! To prove the consistence of this speed vector with T, let us write the associated ODE:

$$\dot{x} = v + L.x. \tag{6.6}$$

While the mathematical form taken by (6.6) might seem unfamiliar, it is much simpler (and more familiar) when expressed in homogeneous coordinates. It simply writes:

$$\dot{\tilde{x}} = \log(\tilde{T}).\tilde{x}, \tag{6.7}$$

which is this time a *linear* ODE. It is well-known from the theory of linear ODEs [Tenenbaum 85] that Eq. (6.7) can be solved analytically and that its solutions are well-defined for all time. With an initial condition x_0 at time 0, the value $x(s)$ of the unique mapping $x(.)$ satisfying Eq. (6.6) is given in terms of matrix exponential by:

$$\tilde{x}(s) = \exp\left(s.\log\left(\tilde{T}\right)\right).\tilde{x}_0. \tag{6.8}$$

By letting s be equal to 1, we thus see that our new velocity vectors are truly consistent with the transformation T.

The ODE of Eq. (6.6) is called *autonomous* (or equivalently *stationary*). Such ODEs have some very nice mathematical properties, which can be expressed in terms of *one-parameter subgroups* of transformations. These properties are detailled in Chapter 2, Sect. 2.2. In short, the flow $\Phi(., s)$ of an *autonomous* ODE is a *one-parameter subgroup of the group of diffeomorphisms*, which means the (possibly) large deformations obtained at time 1 result of the composition of a large number of artitrarily small identical deformations.

One-Parameter Subgroups of Affine Transformations. From the explicit form taken by the solutions of this ODE (see Eq. (6.8)), we can see that the associated flow is simply the family of affine transformations $(T^s(.))$, where T^s is the affine transformation represented by $\exp(s.\log(\tilde{T}))$, i.e. the s^{th} power of T.

From the general properties of flows associated to autonomous ODEs, we know that the family of transformations $(T^s(.))$ is a *one-parameter subgroup* of diffeomorphisms. From this point of view, its infinitesimal generator is the vector field $V(x) = v+L.x$. From the viewpoint of the *affine group* (in contrast to diffeomorphisms), (T^s) is also a one-parameter subgroups of affine transformations, whose infinitesimal generator is this time the principal logarithm of T. Interestingly, it can be shown with the classical tools of Lie groups theory that all continuous one-parameter subgroups of affine transformations are of this form [Sternberg 64].

6.2.3 Log-Euclidean Polyaffine Transformations

An Autonomous ODE for Polyaffine Transformations. With the velocity vectors defined by Eq. (6.5), one can define a novel type of polyaffine transformations using the steps 2 and 3 of the Polyaffine framework. In the sequel, we will refer to these new polyaffine transformations as *Log-Euclidean polyaffine transformations* (or LEPTs). This name comes from our work on diffusion tensors [Arsigny 05a, Arsigny 05b], where we have already used principal logarithms to process this other type of data.

More precisely, let (M_i, t_i) be N affine transformations, and let (L_i, v_i) be their respective principal logarithms. Then one can fuse them according to the weights $w_i(x)$ with the following ODE, which is this time *autonomous*, i.e. without any influence of the time parameter s in the second member of the equation:

$$\dot{x} = \sum_i w_i(x) \left(v_i + L_i.x \right). \tag{6.9}$$

Exactly as in the case of the non-autonomous polyaffine ODE based on Eq. (6.3), solutions to this novel ODE are well-defined for all time s (i.e. never go infinitely far in a finite time, do not 'blow up'), regardless of the initial condition. The proof is extremely similar (although simpler, in fact) to that given in [Arsigny 05c] for the previous polyaffine framework.

Now, we know from the general properties of stationary ODEs (which where presented above) that the flow $T(s, .)$ of this ODE forms a *one-parameter subgroup* of diffeomorphisms: $T(0, .)$ is the identity and $T(r, .) \circ T(s, .) = T(r + s, .)$.

One-Parameter Subgroups of LEPTs. Exactly like in the affine case, the ODE given by (6.9) defines not only a one-parameter subgroup of *diffeomorphisms*, it also yields a one-parameter subgroup of *Log-Euclidean polyaffine transformations.* More precisely, a simple change of variable ($s \mapsto \frac{s}{2}$) shows that the flow at time $\frac{1}{2}$, written here $T(\frac{1}{2}, .)$, corresponds to a polyaffine transformation whose parameters are the same weights as the original ones, but where the affine transformations have been transformed into their square roots (i.e. their logarithms have been multiplied by $\frac{1}{2}$). Similarly, the flow at time s, $T(s, .)$ corresponds to a polyaffine transformations with identical weights but with the s^{th} power of the original affine transformations.

As a consequence, $T(s, .)$ can be interpreted as the s^{th} power of the Log-Euclidean polyaffine transformation defined by $T(1, .)$. In particular, the inverse of $T(1, .)$ (resp. its square root) is given simply by $T(-1, .)$ (resp. $T(1/2, .)$), which is the polyaffine transformation with identical weights but whose affine transformations have been inverted (resp. have been transformed into their square roots).

One should note that our previous polyaffine transformations do *not* have the same remarkable algebraic properties as Log-Euclidean polyaffine transformations. In our previous framework, the inverse of a polyaffine transformation was not even in general a polyaffine transformation. LEPTs have very intuitive and satisfactory properties, because they are based on a fusion of velocity vectors much better adapted to the algebraic properties of affine transformations than the speed vectors we previously used.

In the next Section, we will see how this specific *algebraic* property of our novel framework can be used to alleviate drastically the computational cost of Step 3 of the polyaffine framework (i.e. the cost of the integration of the polyaffine ODE).

Affine-Invariance of LEPTs. Contrary to the previous polyaffine framework, our novel Log-Euclidean framework has another sound mathematical property: *affine-invariance.* This means the Log-Euclidean polyaffine fusion of affine transformations is invariant with respect to any affine change of coordinate system. This type of fusion is thus a fusion between *geometric transformations* and not *matrices* since it does not depend at all on the arbitrary choice of coordinate system chosen to represent them.

To see why this is so, let us see how the various ingredients of our framework are affected by a change of coordinate system induced by an affine transformation A. In homogeneous coordinates, these changes are the following:

- a point $\tilde{x}$ becomes $\tilde{A}.\tilde{x}$
- a weight function $\tilde{x} \mapsto w_i(\tilde{x})$ becomes $\tilde{y} \mapsto w_i(\tilde{A}^{-1}.\tilde{y})$
- an affine transformation $\tilde{T}_i$ becomes $\tilde{A}.\tilde{T}_i.\tilde{A}^{-1}$.

In our new coordinate system, the Log-Euclidean polyaffine ODE writes in homogeneous coordinates:

$$\dot{\tilde{y}} = \sum_i w_i(\tilde{A}^{-1}.\tilde{y}) \log\left(\tilde{A}.\tilde{T}_i.\tilde{A}^{-1}\right).\tilde{y}. \tag{6.10}$$

Then, using the property $\log\left(\tilde{A}.\tilde{T}_i.\tilde{A}^{-1}\right) = \tilde{A}.\log\left(\tilde{T}_i\right).\tilde{A}^{-1}$, the simple change of variable $\tilde{y} \mapsto \tilde{A}.\tilde{x}$ shows that a mapping $s \mapsto \tilde{x}(s)$ is a solution of the Log-Euclidean polyaffine ODE (6.9) if and only if $s \mapsto \tilde{A}.\tilde{x}(s)$ is a solution of (6.10). This means that the solutions of the Log-Euclidean polyaffine ODE in the new coordinate system are exactly the same as in the original coordinate system: our novel polyaffine framework is therefore not influenced by the choice of a coordinate system. Our previous polyaffine framework does not have this property, because it does not take sufficiently into account the algebraic properties of affine transformations.

Another Reason Why our Novel Polyaffine Framework is Called Log-Euclidean. In the special case where all the weight functions $w_i(x)$ do not depend on x, the Log-Euclidean polyaffine fusion of the affine transformations T_i simply yields an affine transformations T, which is given by the following *Log-Euclidean mean*:

$$T = \exp\left(\sum_i w_i \log(T_i)\right).$$

This is another reason why we refer to our novel polyaffine framework as Log-Euclidean. Indeed, the use of a generalization to rigid and affine transformations of our Log-Euclidean framework for tensors (which is presented in Chapter 3) is implicit in this novel framework. More details on the Log-Euclidean framework for linear transformations will be presented in Section 6.4.

Synthetic Examples. Examples of 2D LEPTs are shown in Figs. 6.3, 6.4 and 6.5. In these examples, one can see how antagonistic affine transformations (i.e. transformations whose direct fusion results in local singularities) can be globally fused into a regular and invertible polyaffine transformation.

Closeness to Previous Polyaffine Framework. Interestingly, we have observed in our experiments that the Log-Euclidean and the previous polyaffine frameworks provide similar results. Fig. 6.6 illustrates the striking closeness between both frameworks. Notable differences only appear when very large deformations are fused.

Therefore, the advantage of our Log-Euclidean polyaffine framework over the previous one does not reside in the quality of its results, which are very close to those of the previous one. Rather, it resides in its much better and more intuitive mathematical properties, which allow for much faster computations, as will be shown in the next Section. This situation is somehow comparable to the closeness between the affine-invariant and Log-Euclidean Riemannian frameworks used to process diffusion tensors, detailed in Chapter 4. They also yield very similar results, but in a simpler and faster way in the Log-Euclidean case.

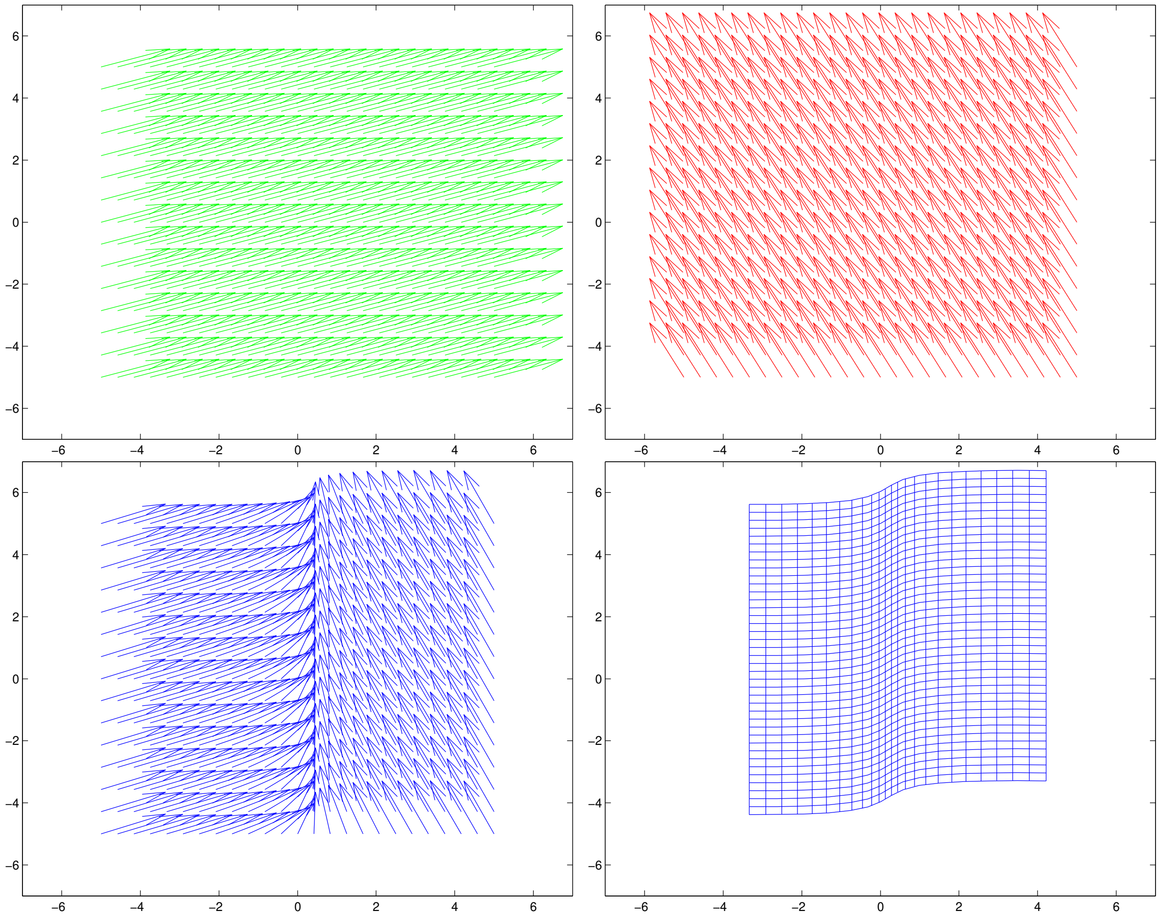

Figure 6.3: **Fusing velocity vectors of two translations. Top (in red and green):** Log-Euclidean polyaffine speed vectors (with the novel framework) of two affine transformations to be fused. **Bottom (in blue):** on the left, fused speed vectors, and on the right, regular grid deformed after integration of the autonomous ODE. Note how the antagonism between the two translations results in a progressive compression along the boundary between the two components. The fusion was carried out with two functions of the first coordinate as weights, as in Fig. 6.1.

6.3 Fast Polyaffine Transform

In this Section, we show how one can use the specific *algebraic* properties of the Log-Euclidean polyaffine framework to obtain fast computations of LEPTs. In particular, we propose an efficient algorithm to evaluate a Log-Euclidean polyaffine transformations on a regular grid. If N is the number of intermediate points chosen to discretize the continuous trajectory of each point, we present here an algorithm only requiring $\log_2(N)$ steps to integrate our autonomous polyaffine ODE, provided that the trajectories of all the points of the regular

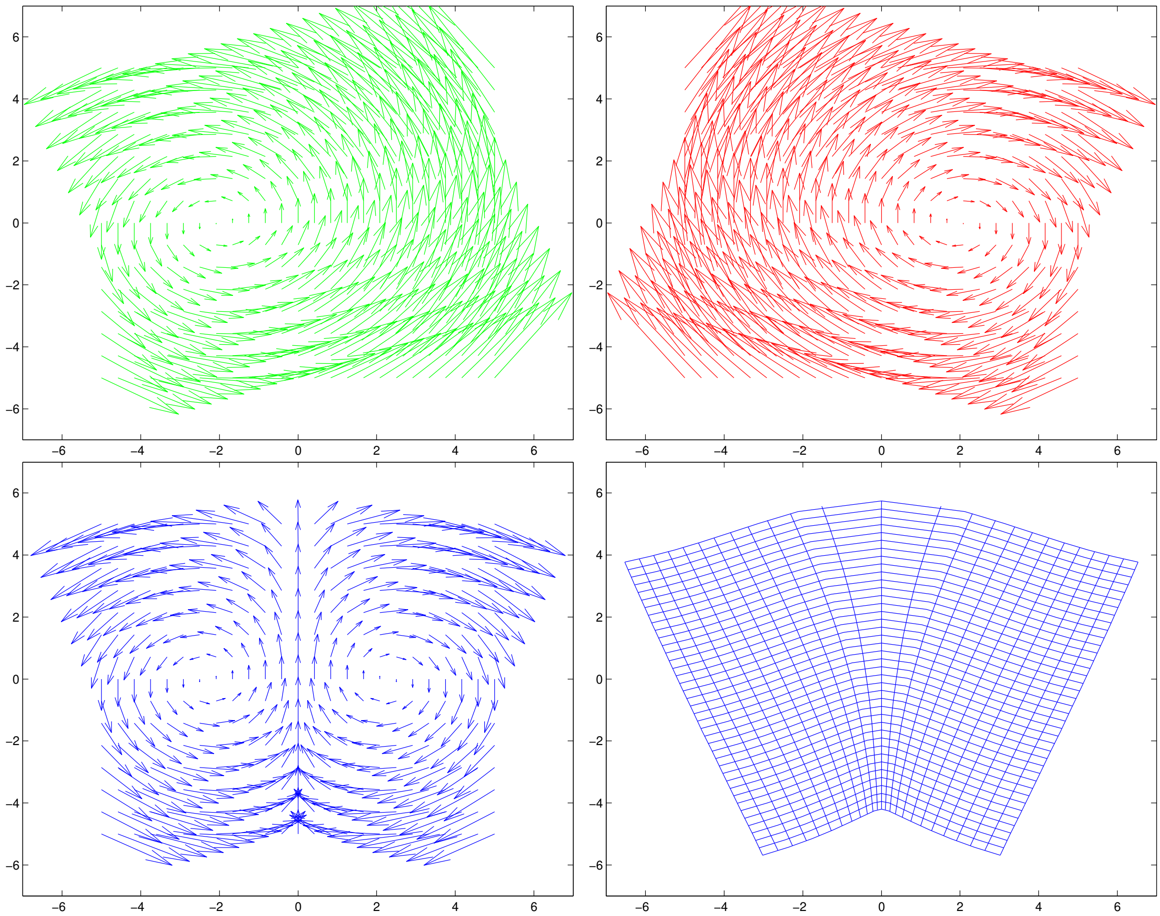

Figure 6.4: **Fusing velocity vectors of two rotations. Top (in red and green):** Log-Euclidean polyaffine speed vectors (with the novel framework) of two affine transformations to be fused. **Bottom (in blue):** on the left, fused speed vectors, and on the right, regular grid deformed after integration of the autonomous ODE. Note how regular and invertible the fused polyaffine transformation, however antagonistic the two fused rotations are locally. The fusion was carried out with two radial functions of the first coordinate as weights, as in Fig. 6.1.

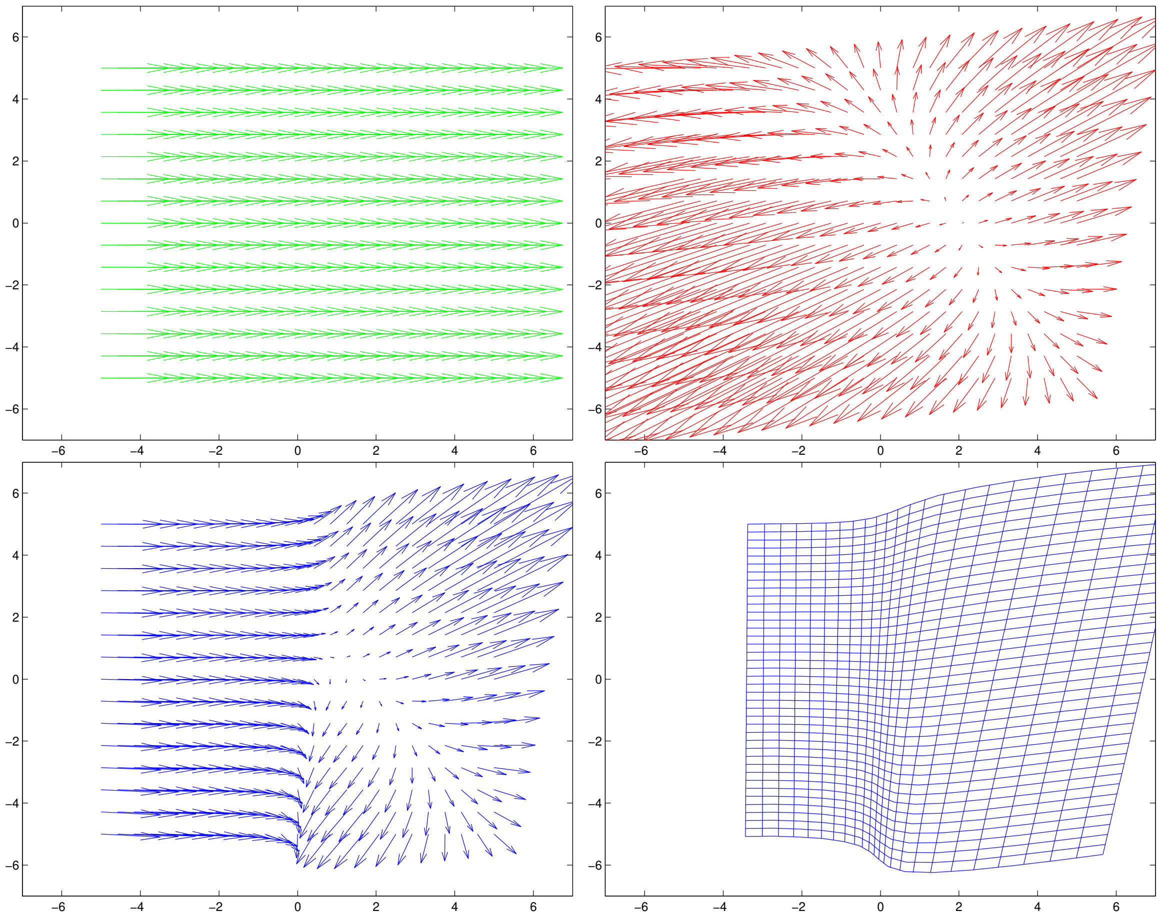

Figure 6.5: **Fusing velocity vectors of a translation and an anisotropic swelling. Top (in red and green):** Log-Euclidean polyaffine speed vectors (with the novel framework) of two affine transformations to be fused. **Bottom (in blue):** on the left, fused speed vectors, and on the right, regular grid deformed after integration of the autonomous ODE. Again, note how locally antagonistic displacements are invertibly fused, resulting in compressions or swelling at the boundary between the two components. The fusion was carried out with two radial functions of the first coordinate as weights, as in Fig. 6.1.

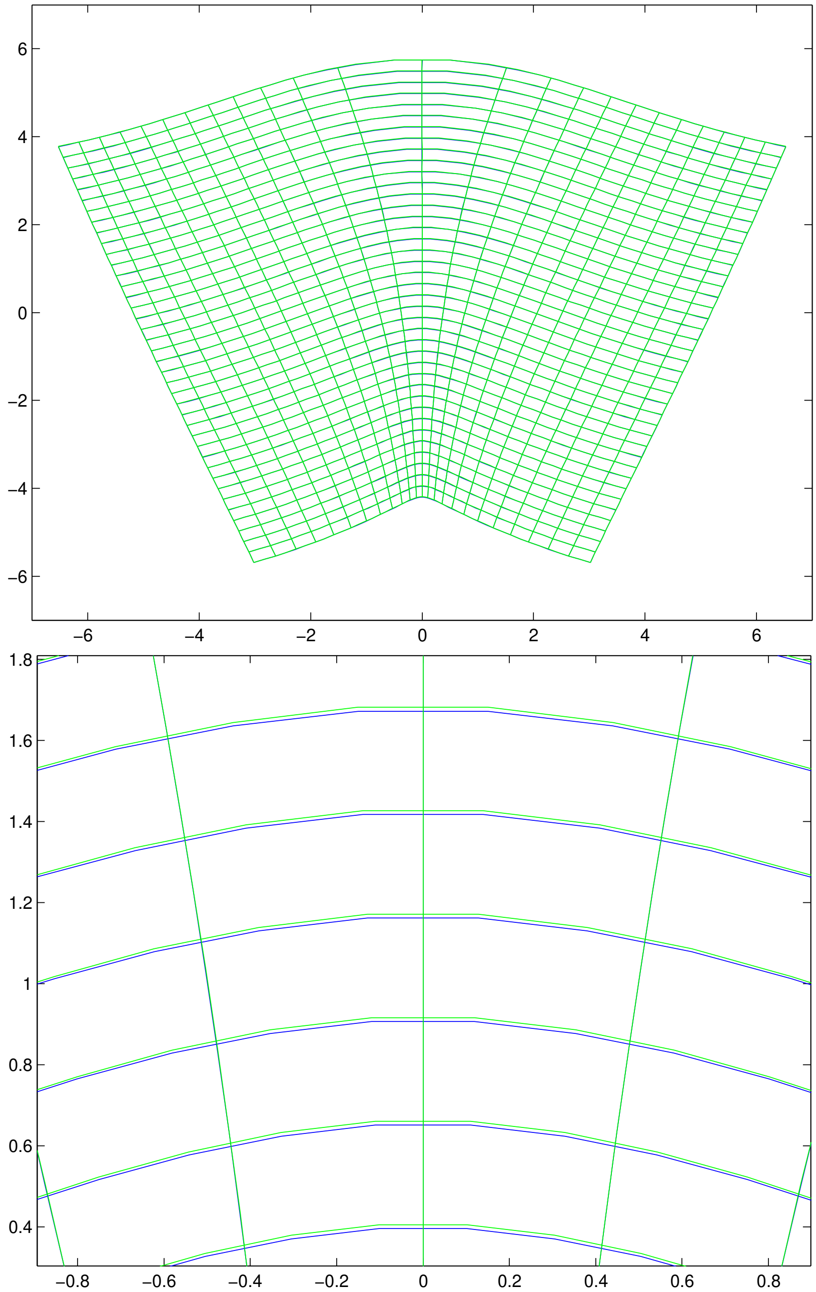

Figure 6.6: **Closeness between the Log-Euclidean polyaffine framework and the previous polyaffine framework.** Superimposed deformed grids in both cases. **Top:** whole grid and **bottom:** close-up. The blue grid corresponds to Log-Euclidean results and the green one to the previous framework (on top of the blue grid), in the case of the fusion of two rotations presented in Fig. 6.4.

grid are computed simultaneously. This drastic drop in complexity is somehow comparable to that achieved by the 'Fast Fourier Transform' in its domain.

Surprisingly, the key to this approach lies in the generalization to the non-linear case of a popular method which is widely used to compute numerically the exponential of a square matrix.

6.3.1 Matrix Exponential and the 'Scaling and Squaring' Method

The matrix exponential of a square matrix can be computed numerically in a large number of ways, with more or less efficiency [Moler 78]. One of the most popular of these numerical recipes is called the 'Scaling and Squaring' method, which is for example used by Matlab ™ to compute matrix exponentials [Higham 05]. Fundamentally, this method is very efficient because it takes advantage of the very specific algebraic properties of matrix exponential, which are in fact quite simple, as we shall see now. For any square matrix M, we have:

$$\exp(M) = \exp\left(\frac{M}{2}\right).\exp\left(\frac{M}{2}\right) = \exp\left(\frac{M}{2}\right)^2. \tag{6.11}$$

This comes from the fact that M commutes with itself in the sense of matrix multiplication. Iterating this equality, we get for any positive integer N:

$$\exp(M) = \exp\left(\frac{M}{2^N}\right)^{2^N}, \tag{6.12}$$

Then, the key idea is to realize that the matrix exponential is much simpler to compute for matrices *close to zero*. In this situation, one can for example use just a few terms of the infinite series of exponential, since high-order terms will be completely negligible. An even better idea is to use Padé approximants, which provide excellent approximations by rational fractions of the exponential around zero with very few terms. For more (and recent) details on this topic, see [Higham 05].

The 'Scaling and Squaring' Method for computing the matrix exponential of a square matrix M can be sketched as follows:

1. **Scaling step**: divide M by a factor 2^N, so that $\frac{M}{2^N}$ is close enough to zero (according to some criterion based on the level of accuracy desired: see [Higham 05] for more details).
2. **Exponentiation step**: $\exp\left(\frac{M}{2^N}\right)$ is computed with a high accuracy using for example a Padé approximant.
3. **Squaring step**: using Eq. (6.12), $\exp\left(\frac{M}{2^N}\right)$ is squared N times (only N matrix multiplications are required.) to obtain a very accurate estimation of $\exp(M)$.

In the rest of this Section, we will see how one can generalize this method to compute with an excellent accuracy polyaffine transformations based on autonomous ODEs.

6.3.2 A 'Scaling and Squaring' Method for LEPTs

Goal of the Method. We would like to compute efficiently and with a good accuracy the values of a Log-Euclidean polyaffine transformation at the vertices of a regular n-dimensional (well, 2D or 3D in practice) grid. The method described below will be referred to as the 'Fast Polyaffine Transform' (or FPT) in the rest of this Chapter.

Algebraic Properties of Log-Euclidean Polyaffine Transformations Revisited. Let $T(s,.)$ be the flow associated to the autonomous polyaffine ODE (6.9), as in Subsection 6.2.3. As mentioned before, this flow is a one-parameter subgroup of LEPTs:

$$T(0,.) = Id \quad \text{and for all } r, s\text{: } T(r,.) \circ T(s,.) = T(r+s,.)\,.$$

As a consequence, Exactly as Eq. (6.11) for the matrix exponential, we obtain for $r = s = \frac{1}{2}$:

$$T(1,.) = T\left(\frac{1}{2},.\right) \circ T\left(\frac{1}{2},.\right) = T\left(\frac{1}{2},.\right)^2 .$$

Iterating this equality, we get for any positive integer N:

$$T(1,.) = T\left(\frac{1}{2^N},.\right)^{2^N} . \tag{6.13}$$

Intuitively, Eq. (6.13) means that what the deformation observed at time 1 results of 2^N times the repetition of the small deformations observed at time $\frac{1}{2^N}$. The total deformation is entirely determined by the initial (and small) deformations occurring just at the beginning of the integration of our ODE (which is a well-known and general phenomenon with autonomous ODEs).

Fast Polyaffine Transform. We can now generalize the 'Scaling and Squaring' Method to the Log-Euclidean polyaffine case. This method, called the 'Fast Polyaffine Transform', follows the usual three steps:

1. **Scaling step**: divide $V(x)$ (the field of velocity vectors) by a factor 2^N, so that $\frac{V(x)}{2^N}$ is close enough to zero (according to the level of accuracy desired).
2. **Exponentiation step**: $T\left(\frac{1}{2^N},.\right)$ is computed using an adequate numerical scheme.
3. **Squaring step**: using Eq. (6.13), $T\left(\frac{1}{2^N},.\right)$ is squared N times (in the sense of the composition of transformations; only N compositions are required.) to obtain an accurate estimation of $T(1,.)$, i.e. of the polyaffine transformation to be computed (e.g., an average relative error of the order of 0.5%).

From a practical (or numerical) point of view, two points remain to be clarified. First what numerical scheme can be used to compute $T\left(\frac{1}{2^N},.\right)$ with a good precision during the 'exponentiation step'? Second, how should the composition (which is the multiplication operator for transformations) be performed during the 'squaring step'?

Exponentiation Step. Exactly as in the matrix exponential case, integrating an ODE during a very short interval of time (short with respect to the smoothness of the solution) is quite easy. We can use any of the methods classically used to integrate ODEs during short periods of times, like explicit schemes or Runge-Kutta methods, which are based on various uses of the Taylor development to compute solutions of ODEs (see [Lambert 91] for more details on these methods).

The simplest of these schemes is undoubtedly the first-order explicit scheme. In our case, it simply consists in computing the following value:

First Order Explicit Exponentiation Scheme (E.S):

$$T\left(\frac{1}{2^N}, x\right)_{\text{E.S.}} \stackrel{def}{=} x + \frac{1}{2^N}.V(x).$$

Generalizing the ideas already developed in [Arsigny 05c] for the previous polyaffine framework, we can also use a second-order scheme which takes into account the affine nature of all components, and which is *exact* in the case of a single component. We will refer to this scheme as the *affine exponentiation scheme* in the following. It writes:

Second Order Affine Exponentiation Scheme (A.S):

$$T\left(\frac{1}{2^N}, x\right)_{\text{A.S.}} \stackrel{def}{=} \sum_{i=1}^{N} w_i(x).T_i^{\frac{1}{2^N}}(x),$$

where $T_i^{\frac{1}{2^N}}$ is the $2^{N\text{th}}$ root of the affine transformation T_i. We will see later in this Section that the accuracy of this numerical scheme is slightly better than that of the explicit scheme, probably because it takes into account the linear nature of the components.

Composing Discrete Transformations. In this work, we are evaluating our transformation at a *finite* number of vertices of a regular grid. Practically, one has to resort to some kind of interpolation/extrapolation technique to calculate the value of such a transformation at *any* spatial position. Numerous possibilities exist in this domain, such a nearest-neighbor interpolation, bi- or tri-linear interpolation, continuous representations via the use of a basis of smooth functions like wavelets, radial basis functions... In the following, we use bi- and tri-linear interpolations, which are simple tools guaranteeing a continuous interpolation of our transformation. The best type of interpolation technique for the purposes of our Fast Polyaffine Transform remains to be determined and will be the subject of future work.

Algorithmic Complexity. Note that to compute polyaffine transformations using the FPT, the weight functions need only be evaluated *once* per voxel, and not at *every step* of the integration of the ODE, as was done in [Arsigny 05c]. When weight functions are stored in the computer memory as 3D scalar images, this offers the opportunity of removing them from the computer RAM after the exponentiation step. This could be particularly useful when a large number of affine components are used on high-resolution images.

Furthermore, the equivalent of 2^N intermediate points is achieved in only N steps, in contrast with the 2^N steps required by a traditional method. After the $2^{N\text{th}}$ root has been computed, only N compositions between transformations need to be computed, which is an operation based on interpolation techniques and therefore not very computationally expensive. Let N_{vox} be the number of voxels and let N_{pts} be the number of intermediary points chosen to integrate the polyaffine ODE. The complexity of our new algorithm is thus $O(N_{\text{vox}}.\log_2(N_{\text{pts}}))$, whereas the complexity of traditional methods of integration of this ODE is $O(N_{\text{vox}}.N_{\text{pts}})$.

Computing the Inverse of a Polyaffine Transformation. As pointed out in Subsection 6.2.3, in our new framework the inverse of a polyaffine transformation is simply the polyaffine transformation associated with the opposite vector field (i.e. the polyaffine transformation with the same weights but inverted affine components). As a consequence, the inverse of a polyaffine transformation can be also computed using the Fast Polyaffine Transform. Actually, any power (square root, etc.) a polyaffine transformation can be computed this way.

6.3.3 2D Synthetic Experiments

Throughout this results Section, we measure the accuracy of our results by computing the relative difference of the results with respect to accurate estimations of the real (continuous) transformations. These reference transformations are obtained by a classical integration (i.e., a fixed time step was used) of the Log-Euclidean polyaffine ODE for each of the pixels of the grid, using a small time step: 2^{-8}.

One should note that several parameters influence the accuracy of the results:

- the scaling 2^N
- the geometry of the regular grid
- the interpolation method
- the extrapolation method.

Thus, compared to the classical estimation method with a fixed time step, our fast transform possesses three new sources of numerical errors: the geometry of the regular grid (the transformation is evaluated only at a finite number of points, the more points the more precise the result will be), the interpolation method and the fact that regardless of the extrapolation method, some part of the information about what happens outside of the regular grid is lost. It is therefore important to check that the accuracy of the results obtained with the FPT are not spoiled by these new sources of error.

A Typical FPT. Figs. 6.7 and 6.8 display the results of a typical Fast Polyaffine Transform, using two rotations of opposite angles, and a scaling of 2^6 (and therefore 6 squarings). The regular grid chosen to sample the transformation is of 50×40 pixels. The affine exponentiation scheme is used.

On average, the results are quite good: the average relative error is approximately equal to 0.6%. However, much higher errors (around 11%) are obtained at the boundary, which comes from the fact that the bi-linear interpolation we use here does not take into account the rotational behavior of the transformation outside of the grid.

Using Bounding Boxes to Correct Boundary Effects. The numerical errors stemming from the loss of information at the boundary of the regular grid can be drastically reduced for example by enlarging the regular grid used. A simple idea consists in adding to the regular grid some extra points so that it contains the points of boundary deformed by Euclidean fusion of the affine components. This is illustrated by Fig. 6.9.

Fig. 6.10 presents the accuracy of the results given by the FPT, this time using a regular grid extended in the way described just above. This time, errors are much lower: the relative accuracy of the resulting estimation of the polyaffine transformation is on average of 0.21% (instead of approximately 0.6% without an enlarged grid), and the maximal relative error is below 3.2% (instead of 11% without an enlarged grid). This simple and efficient technique, which drastically reduces the effect of boundary effects on the FPT, is used systematically in the rest of this Chapter.

Influence of Scaling. What scaling should be chosen when the FPT is used? Of course, this depends on the quantity of high frequencies present in the polyaffine transformations. The more sharp changes, the smaller the scaling should be and the finer the sampling grid should also be.

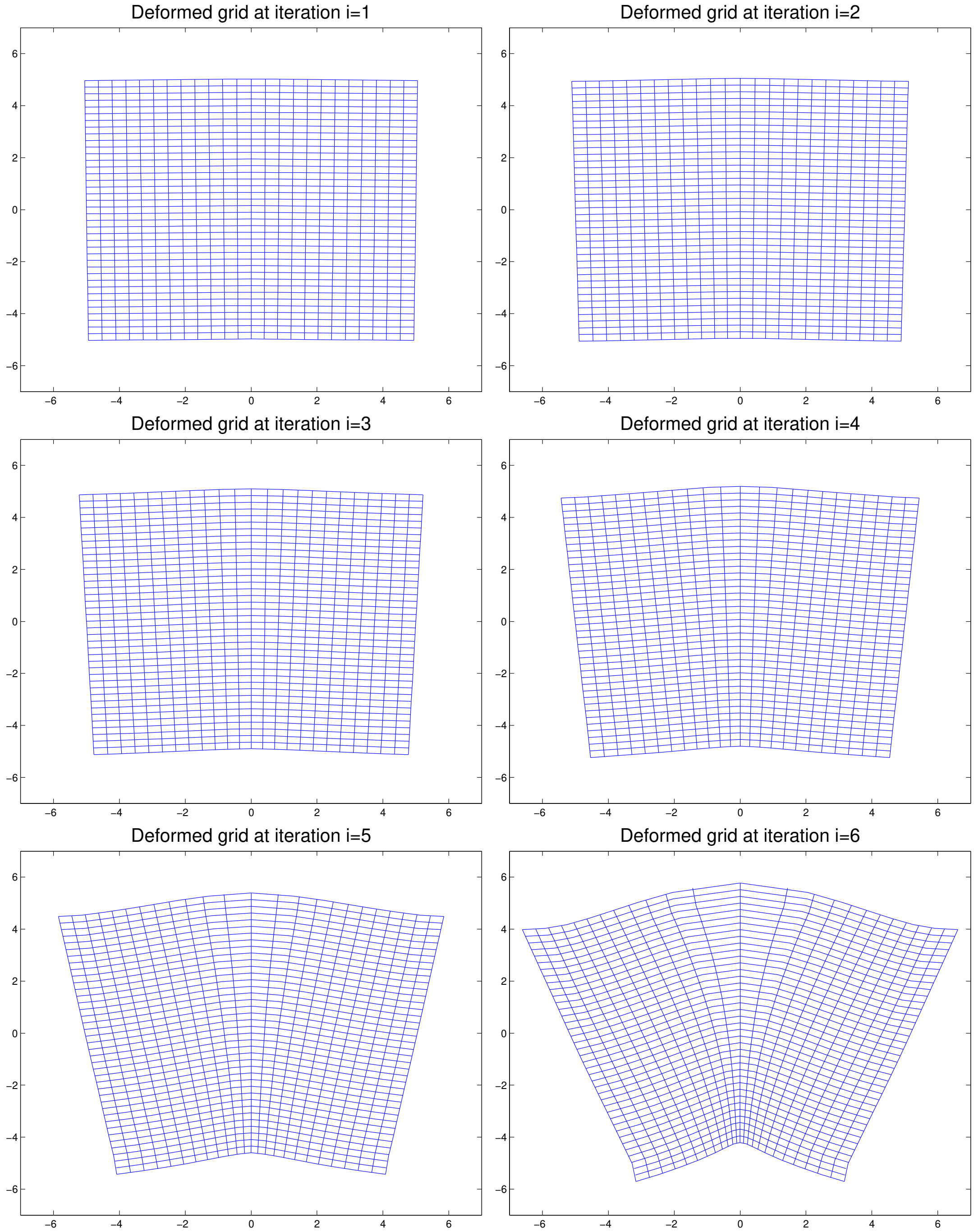

Figure 6.7: **Fast polyaffine transform for two rotations.** A scaling factor of 2^6 was used in this experiment, and there are therefore 6 squaring steps. Note how the deformation is initially very small, and increases exponentially. The accuracy of the FPT results was measured with respect to the results given by a classical integration (voxel by voxel) of the polyaffine ODE with 2^8 intermediate points. The relative error of the resulting estimation of the polyaffine transformation is below 0.6% on average and the maximal relative error, as expected, is made at the boundary and is below 11%.

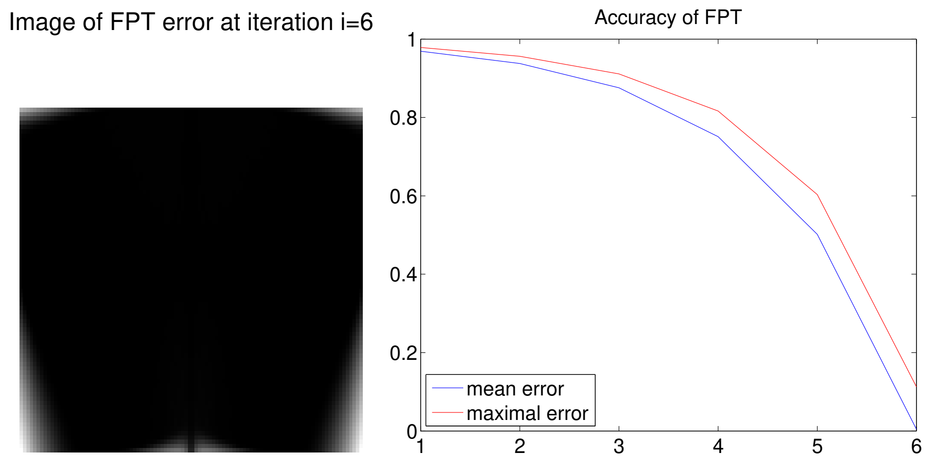

Figure 6.8: **Fast polyaffine transform for two rotations: errors localization and evolution. Left:** the errors at the vertices of our 50×40 regular grid are displayed as an image, after a FPT with 6 squarings. Note how the maximal relative errors are concentrated on the boundary of our grid. This is due to the inaccuracy of our extrapolation technique, which is only bi-linear and does not deal very precisely with the affine nature of the polyaffine transformation. **Right:** the evolution of errors along squarings is displayed. The relative error of the resulting estimation of the polyaffine transformation is below 0.6% on average and the maximal relative error is below 11%.

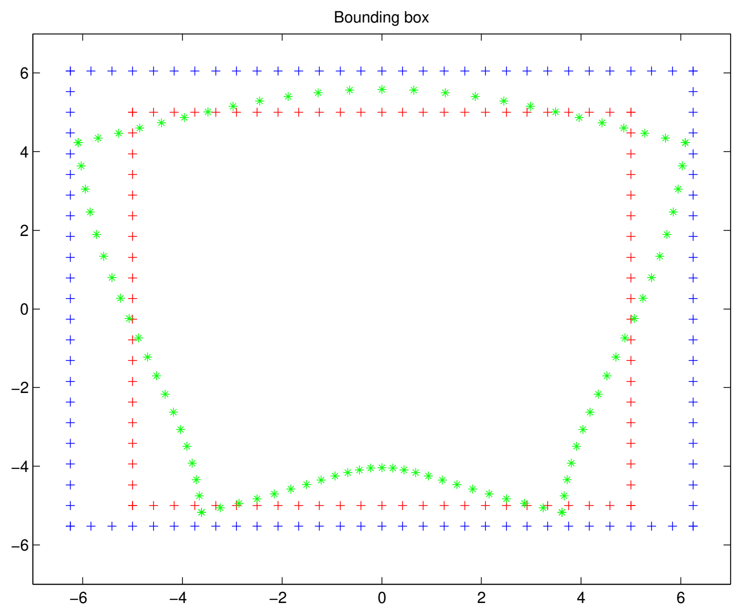

Figure 6.9: **Enlarging the original regular grid with a bounding box. In red (plus signs):** the original regular grid used to sample the Log-Euclidean polyaffine fusion between two rotations. **In green (stars):** grid deformed by direct fusion of the two rotations, which can be computed at a very low computational cost. **In blue (plus signs):** regular grid extended so that it now contains the green points. This enlarging procedure considerably reduces the impact on the Fast Polyaffine Transform of the loss of information beyond the boundaries of the regular grid, as shown in this Section.

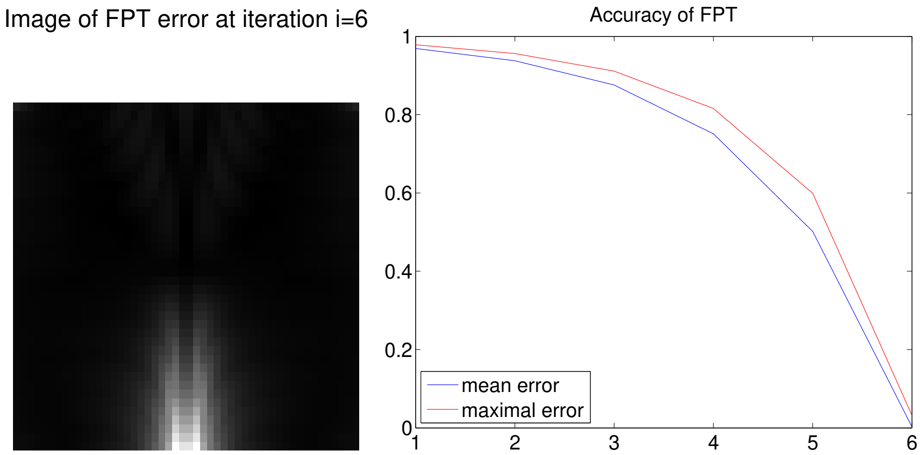

Figure 6.10: **Using an enlarged sampling grid: impact on errors localization and evolution for the fast polyaffine transform for two rotations. Left:** the errors at the vertices of our 50×40 regular grid are displayed as an image, after a FPT with 6 squarings. Note how the maximal errors are concentrated this time on the region of highest compression. **Right:** the evolution of errors along squarings is displayed. This time, errors are much lower: the relative error of the resulting estimation of the polyaffine transformation is on average below 0.21% (instead of below 0.6% without an enlarged grid), and the maximal relative error is below 3.2% (instead of 11% wihtout an enlarged grid).

Fig. 6.11 displays the performance in accuracy of the FPT when the number of iterations N varies. In this experiment, we use the same fusion of rotations as in the previous experiment. In this case, the optimal scaling is 2^5. Larger scalings do not result in better accuracy, essentially because of the missing information at the boundary.

We observed in the experiments on real 3D medical images described in the sequel of this Chapter that even much smaller scalings (typically 2^3 or 2^2) could be used without sacrificing the accuracy of the result. In short, introducing even a small number of intermediate points substantially regularizes the fused transformation with respect to the direct fusion, since this suffices to remove singularities in practice. Using more intermediary points, i.e. 5 or more squarings, offers the possibilities to be very close to the ideal polyaffine transformation, which provides a simple way to compute the inverse of the fused transformation with an excellent accuracy, as will be shown in this subsection.

Moreover, one should also note from Fig. 6.11 that our Fast Polyaffine Transform is very *stable*: using unnecessary iterations (or equivalently a very large scaling) does not result in numerical instabilities. The result is mostly independent of N for $N > 6$.

Comparison between Numerical Schemes. Here, we compare the explicit affine exponentiation schemes. We perform this comparison on our three favorite examples: the fusion of two rotations, the fusion of two antagonistic translations as in Fig. 6.3, and the fusion between a translation and an anisotropic swelling as in Fig. 6.5. The accuracy of the FPT using both numerical schemes is compared in all three cases. Fig. 6.12 shows the results.

Both numerical schemes make the FPT converge toward the same accuracy as the number of squarings increases, but the convergence is slightly faster in the affine exponentiation case: the average error is 40% smaller in the affine case for scalings smaller than 2^6. Interestingly, the two numerical schemes are identical for the fusion of the two translations, because the

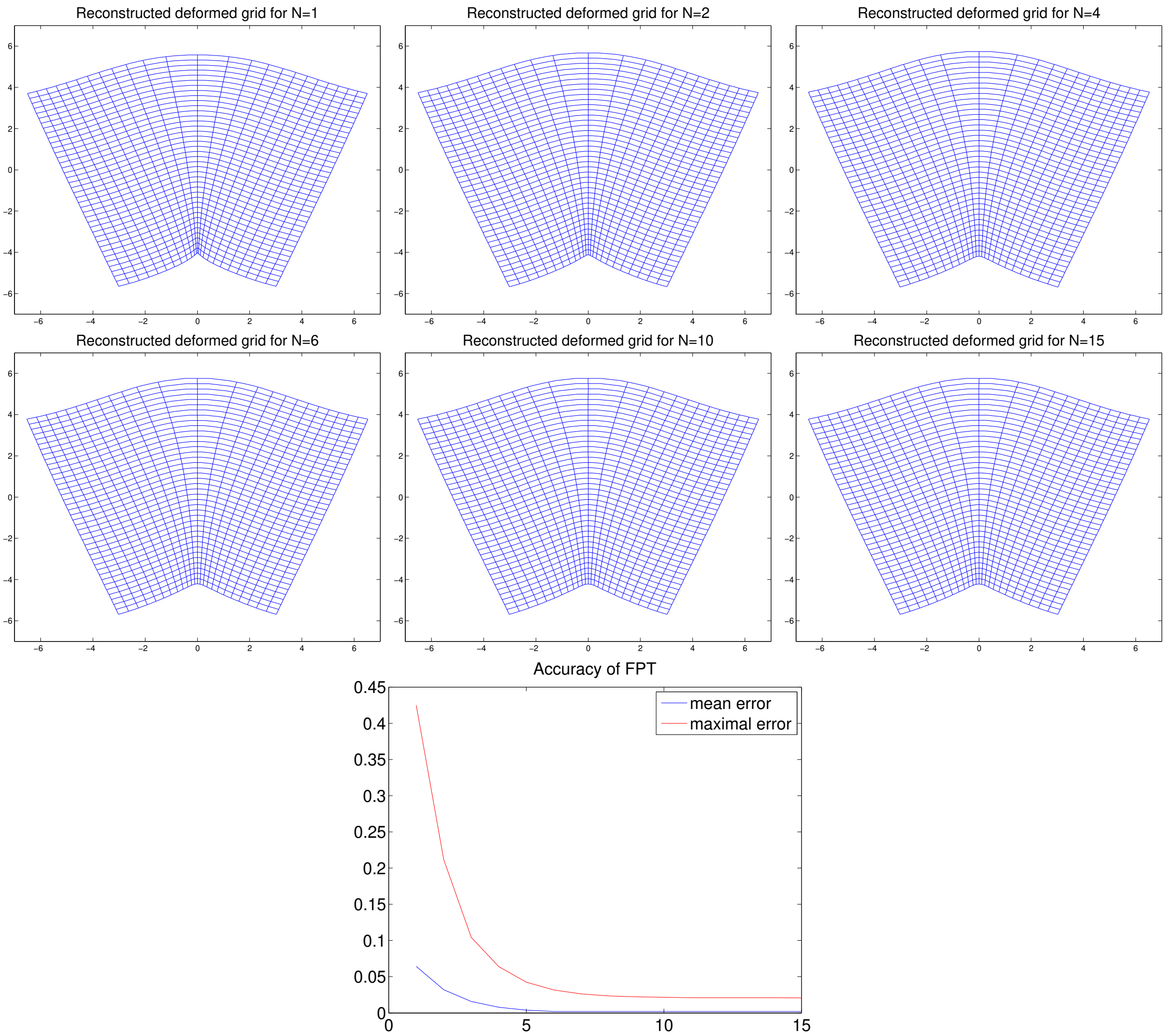

Figure 6.11: **Fast polyaffine transform for two rotations: influence of scaling. Above:** regular grids deformed by polyaffine transformations obtained with the FPT using different values of the scaling factor. The scaling factors are the following: 2^1, 2^2, 2^4, 2^6, 2^{10} and 2^{15}. Note how close the results are when the number of squarings N is larger or equal to 7. **Below:** accuracy of the estimations when N varies. The results are extremely *stable* for $N > 6$: it is unnecessary to use larger scalings in the example considered here. However, remarkably, using larger scalings *does not change* the results: our FPT is very stable. The relative error of the resulting estimation of the polyaffine transformation converges toward 0.2% on average and the maximal relative error converges toward 2% for large Ns. The residual maximal error is essentially due to the sampling of the transformation on a grid and the use of an interpolation method between the points of the grid, since an extended grid is used to drastically reduce errors at the boundary of the grid.

linear parts of these two affine translations are equal to the identity.

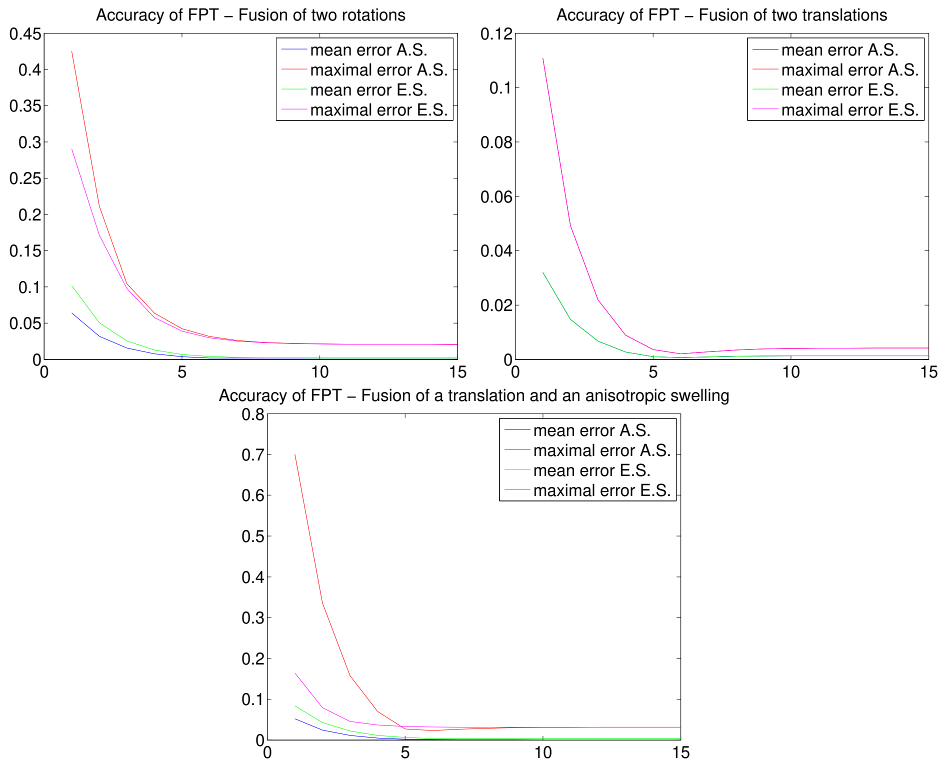

Figure 6.12: **Comparison between numerical schemes. From left to right and then from top to bottom:** fusion between two rotations, two translation and finally a translation and an anisotropic swelling. A.S. stands for 'affine exponentiation scheme' and E.S. for 'explicit exponentiation scheme'. Interestingly, the two numerical schemes are identical for the fusion of the two translations, because the linear parts of these two affine translations are equal to the identity. Both numerical schemes make the FPT converge toward the same accuracy as the number of squarings increases, but the convergence is substantially faster in the case of the affine exponentiation scheme: in the two cases where the schemes yield different results, the average relative error is 40% smaller in the affine case for scalings smaller than 2^6.

Inverting Polyaffine Transformations with the FPT. As pointed out previously, in our novel framework, the inverse of a polyaffine transformation is simply (and quite intuitively) the polyaffine transformation with the same weights and with inverted affine components. This inverse can also be computed using the Fast Polyaffine Transform, and in this experiment we tested the accuracy of the inversion obtained this way. The affine exponentiation scheme was used for exponentiation along with a 50×40 grid.

Fig. 6.13 presents with deformed grids the evolution of the accuracy of inversion when the number of squarings varies, in our example of fusion between two rotations. Fig. 6.14 presents the quantitative results in the three cases of fusion used in the previous experiment. We thus see that an excellent quality of inversion can be achieved using a small number of squarings, typically 6.

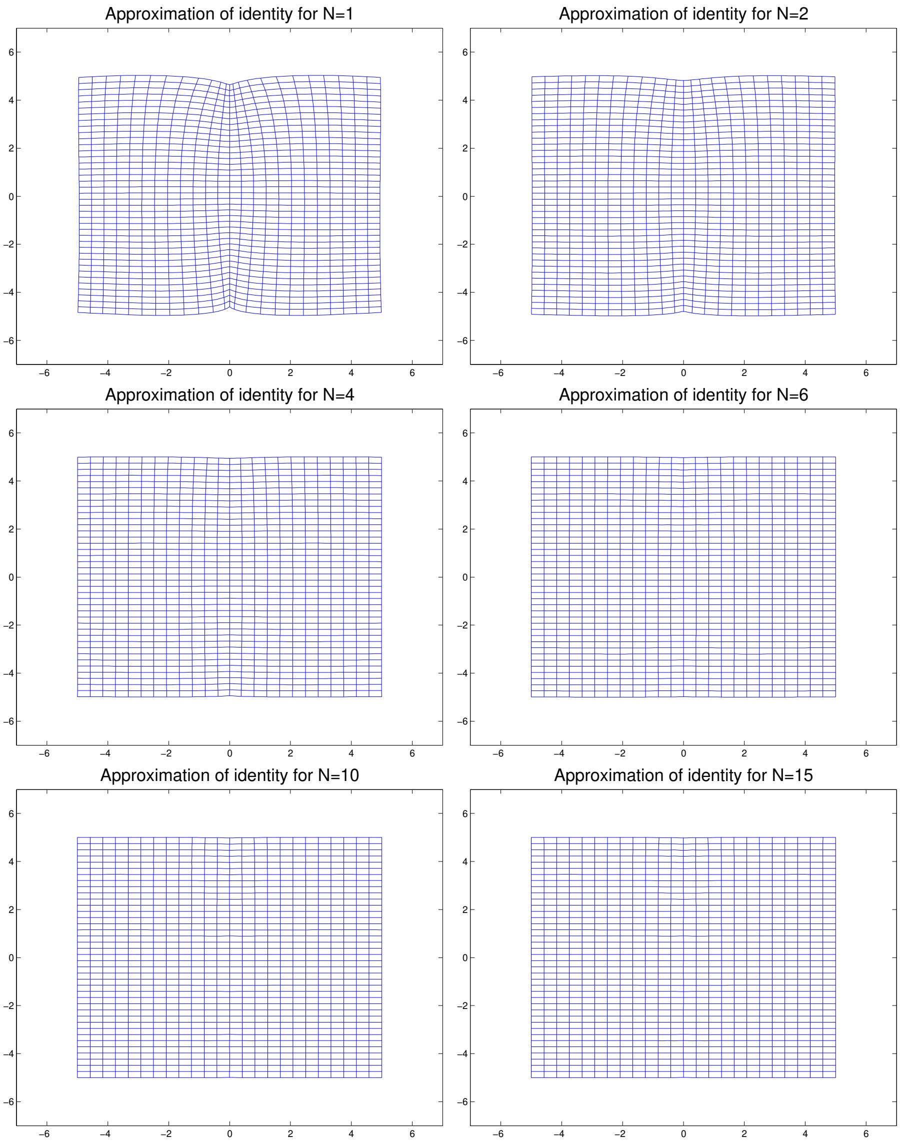

Figure 6.13: **Inverting a polyaffine transformation with the FPT. From left to right and then from top to bottom:** our regular grid is deformed by the composition between the FPT of the fusion between two rotations and the FPT of its inverse, for different numbers of squarings N. One can see that an excellent accuracy of inversion is already achieved with 6 squarings.

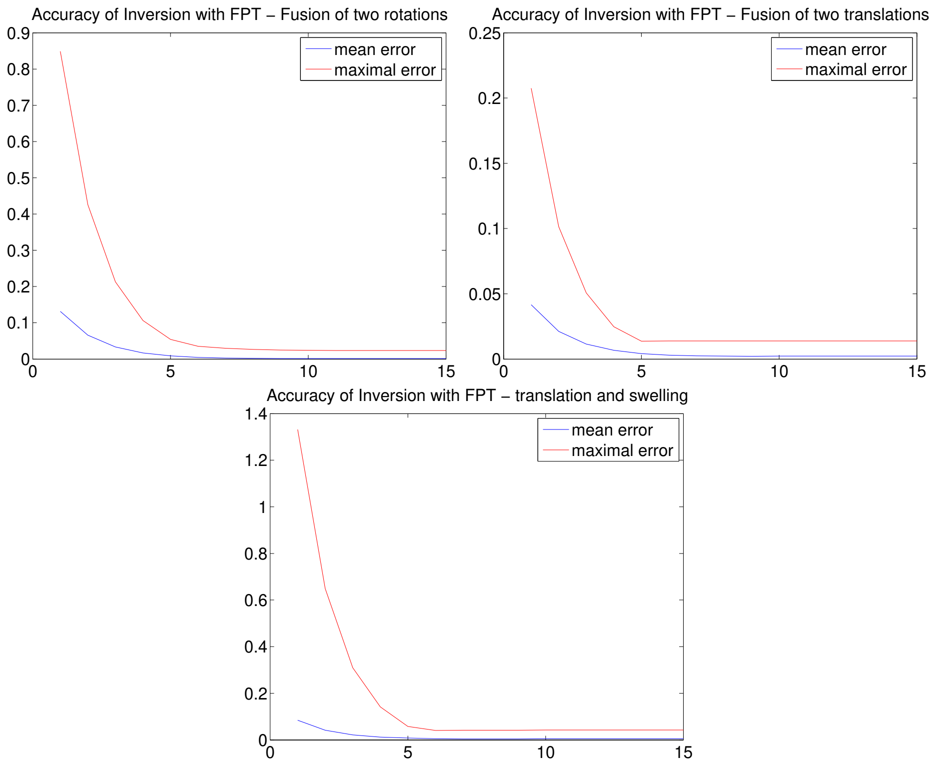

Figure 6.14: **Inverting a polyaffine transformation with the FPT: quantitative results. From left to right and then from top to bottom:** fusion between two rotations, two translation and finally a translation and an anisotropic swelling. The composition between the FPT of the transformation of the FPT of its inverse is carried out, for different numbers of squarings. The errors displayed are relative with respect the polyaffine transformation considered: the displacements are expected to be close to zero (i.e. the resulting transformation is expected to be close to the identity), and the errors are measured with respect to the displacements observed originally. One can see that an excellent accuracy of inversion is already achieved with 6 squarings. As expected, the maximal errors are observed at the boundary of the grid, which can be fixed for example by using a larger grid to compute the FPT.

6.3.4 3D Registration Example

In Chapter 5, we had seen how it was possible to optimize the parameters of polyrigid or polyaffine transformations in medical image registration experiments. However, this leads in practice to a high computational cost, and with Olivier Commowick (Ph.D. candidate in our team and also working for DOSISoft SA, Cachan, France), we have worked toward drastically reducing this cost.

To obtain short computation times (typically 10 minutes for whole 3D volumes in the locally affine case), our locally affine registration algorithm, presented in [Commowick 06a, Commowick 06b], estimates affine components using the *direct fusion*. The FPT is used in a *final step* to ensure the invertibility of the final transformation, as well as to compute its inverse. We have observed experimentally that this yields quite satisfactory results, as we will see below.

3D Atlas Registration Experiment. Let us consider a real 3D example of locally affine registration, between an atlas of $216 \times 180 \times 180$ voxels and a T_1-weighted MR image, with our multi-resolution and robust block-matching algorithm, without regularization. Seven structures of interest are considered: eyes (1 affine component each), cerebellum (2 components), brain stem (2 components), optic chiasm (1 component), 1 supplementary component (set to the identity) elsewhere. Weight functions are defined in the atlas geometry using mathematical morphology and a smoothing kernel in a preliminary step and remain unchanged during the registration process.

Philosophy of our Locally Affine Algorithm. Here, the idea is to use a registration procedure capable of registering finely a number of fixed structures of interest, with *very smooth* transformations. In contrast, many registration algorithms are able to register finely *the intensities* of the images of two anatomies, but this is done in most cases at the cost of the regularity of the resulting spatial transformation. This lack of smoothness leads to serious doubts regarding the anatomical likelihood of such transformations.

Fig. 6.15 provides a comparison between the typical smoothness of dense transformation and locally affine registration results. Interestingly, much smoother deformations are obtained in the locally affine case with an accuracy in the structures of interest which is comparable to the dense transformation case of [Stefanescu 04]. More details on this subject can be found in [Commowick 06b].

LEPTs as a Powerful Post-Processing Tool. As we mentionned before, our locally affine registration algorithm estimates affine components using the *direct fusion*. The FPT is used in a *final step* to ensure the invertibility of the final transformation, as well as to compute its inverse. Here, the scaling used in 2^8 and the FPT is computed in $40s$ on a Pentium4 Xeon™2.8 GHz on a $216 \times 180 \times 180$ regular grid.

As shown by Fig. 6.16, the direct fusion of components estimated by our locally affine algorithm can lead to singularities, which is not the case when the FPT is used. Remarkably, both fusions are very close *outside* of regions with singularities. This means that no artifacts are introduced by the FPT, which justifies *a posteriori* the estimation of affine components with the (faster) direct fusion.

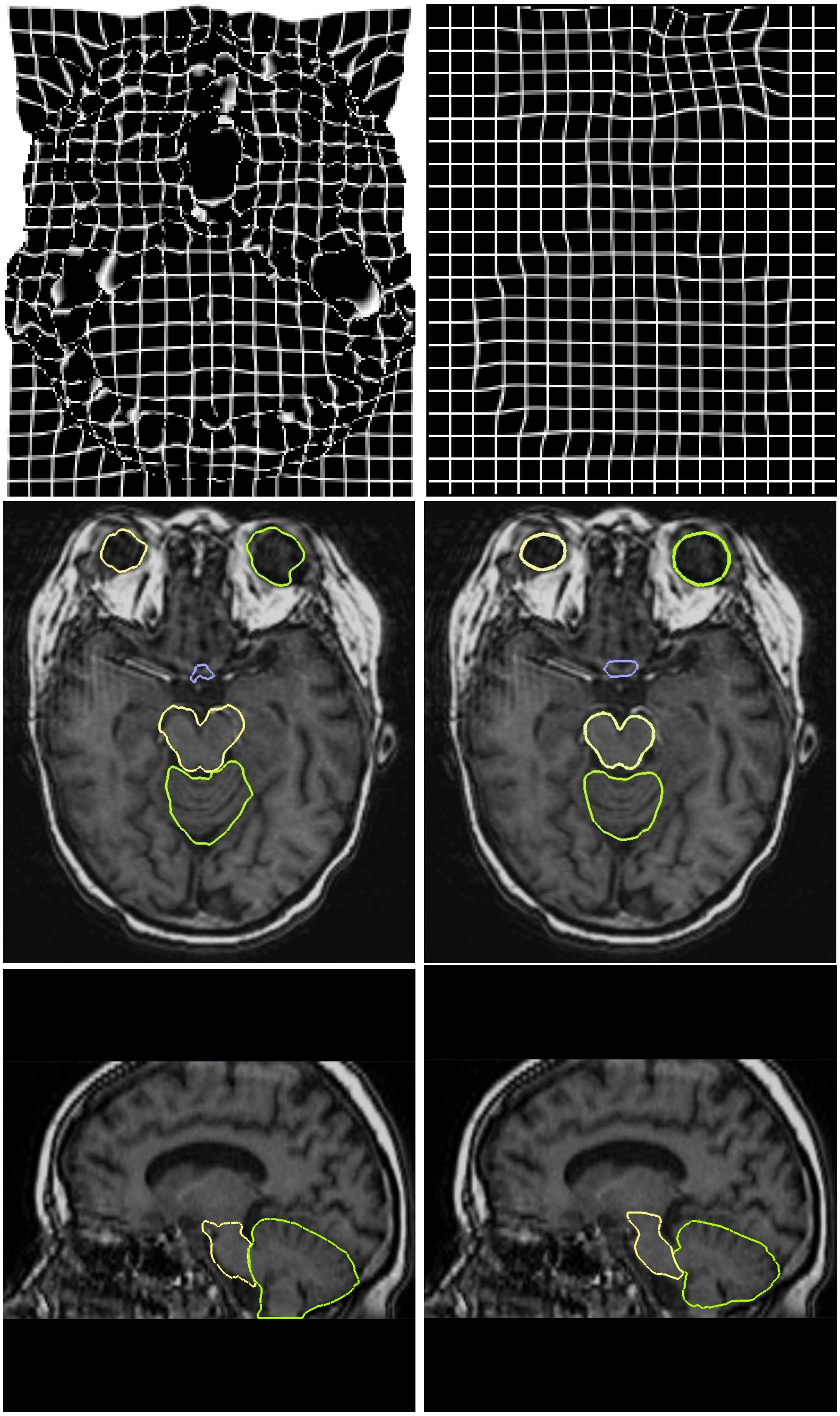

Figure 6.15: **Locally affine vs. dense-transformation: smoothness of deformations. Left:** results of dense transformation algorithm [Stefanescu 04]. **Right:** locally affine case. **Top:** deformed grids (axial slices). **Middle and bottom:** the contours of our structures of interest (eyes, brain stem, cerebellum, optic chiasm) are displayed on the subject (middle : axial slice, bottom: sagittal slice). These contours are obtained by deforming those of the atlas. Note how smoother deformations and contours are in the locally affine case, although both accuracies are comparable.

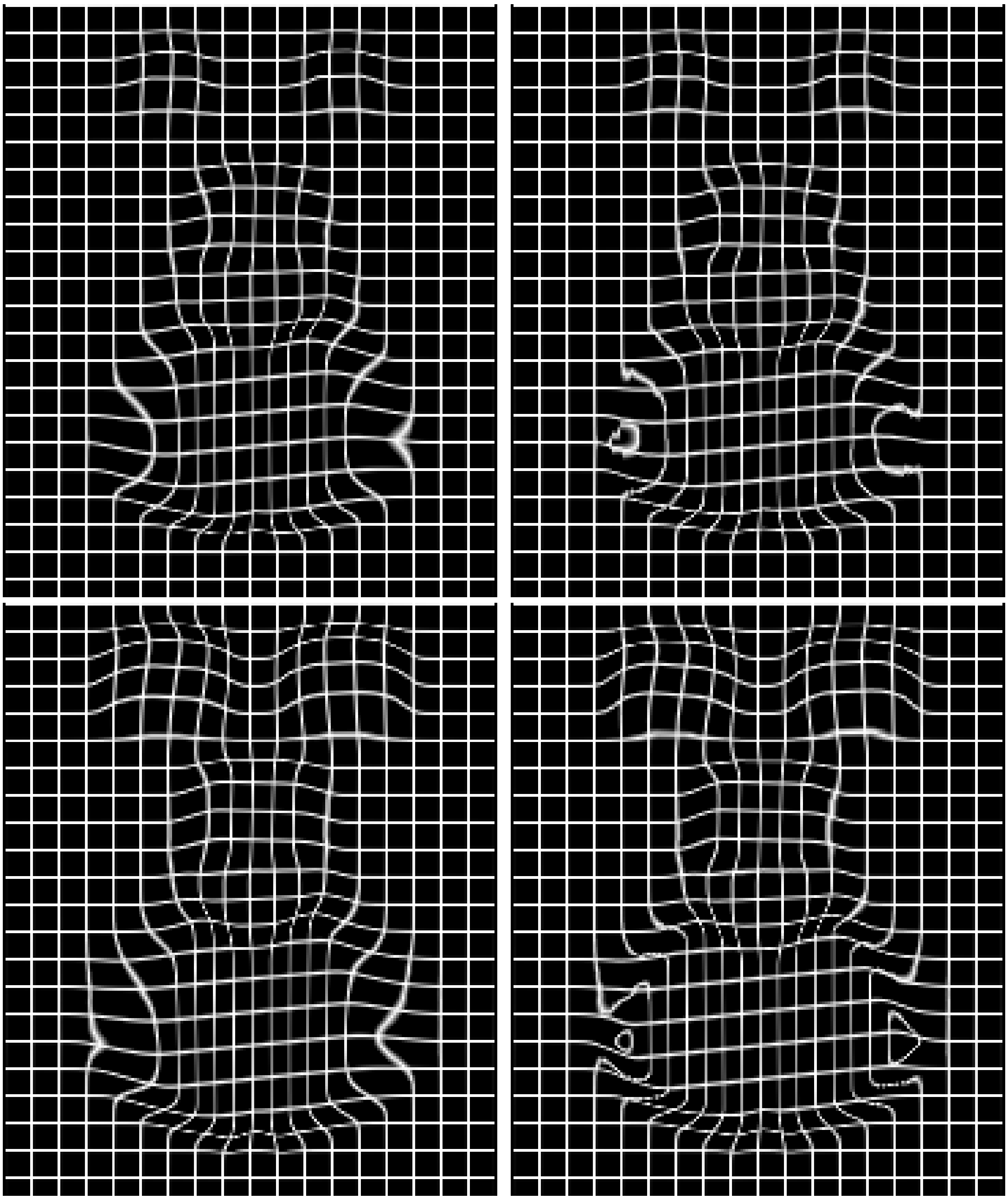

Figure 6.16: **Singularity removal with LEPTs.** A 3D regular grid is deformed with the locally affine transformation obtained with the algorithm of [Commowick 06a,Commowick 06b], two slices are displayed. **From left to right:** polyaffine fusion and direct fusion (two axial slices are displayed: one on top, one at the bottom). Note how the singularities of the direct fusion disappear with LEPTs. Remarkably, this is obtained without introducing any artifacts: outside singularities, both fusions yield very close results.

6.4 A Log-Euclidean Framework for Rigid and Affine Transformations

As we have seen in Section 6.2.3, the problem of fusing into a global invertible transformation several local rigid or affine components is closely linked to the averaging of rigid or affine transformations. Indeed, in the case where all the weight functions $w_i(x)$ do not depend on x, fusing several rigid or affine components in the Log-Euclidean polyaffine framework results in a *Log-Euclidean* averaging of these components. This means that the obtained global transformation T is rigid or affine (which is quite intuitive, since it would be surprising in this particular case to get a non-linear result) and is given as a *Log-Euclidean mean*:

$$T = \exp\left(\sum_i w_i \log(T_i)\right). \tag{6.14}$$

In our original polyaffine framework, we also have this natural property, but this time the global rigid or affine transformation is obtained in the following way:

$$T = (L, t), \text{ with: } L = \exp\left(\sum_i w_i \log(L_i)\right) \text{ and } t = \sum_i w_i . t_i. \tag{6.15}$$

The averaging procedure given by (6.15) is not entirely satisfactory, since it does not take into account the *semi-direct product* nature of the rigid and the affine groups. The averaging is carried out independently on the linear parts and the translation parts, which results in the case of affine transformations in the absence of *affine-invariance*: the averaging depends on the current coordinate system, and is only an averaging of *matrices* and *not* an averaging between *geometrical deformations* (which should not depend on the arbitrary choice of coordinate system). As we will see in this Section, a Log-Euclidean averaging has no such defect.

In the rest of this Section, we present the properties the general Log-Euclidean framework for linear transformations. Similarly to the Log-Euclidean framework for tensors which is presented in Chapter 3, it is simple to use and also has some excellent theoretical properties (althouth slightly fewer here than in the particularly neat tensor case). This framework can for example be used to compute statistics on linear transformations, which is an important issues in the medical imaging community, as detailled in Section 1.1.3. In this framework, the only restriction imposed on linear transformation to be processed is the following: their principal logarithm has to be well-defined (i.e. the processed transformations should not be too far away from the identity, in the sense of the precise criterion given in Section 2.3.4). At the end of this Section, we also present briefly the Log-Euclidean framework for general (possibly abstract) Lie groups.

We recently found out that in 2002, Alexa already proposed in [Alexa 02] to process linear geometrical transformations via their logarithms, in the context of the interpolation of transformations for computer graphics, and had also suggested to perform statistics on these transformations via their logs. One should note that our approach, which we developped independently, goes considerably deeper into the analysis of the properties of this framework. In particular, we were the first to our knowledge to put into light the geometric interpolation of determinants provided by the Log-Euclidean mean, as well as the *invariance properties* of this framework (i.e., inversion-invariance and affine-invariance). The extension of the Log-Euclidean framework to abstract Lie groups presented at the end of this Section is also entirely novel to our knowledge.

6.4.1 Log-Euclidean Metrics

The basic idea behind the Log-Euclidean framework for linear transformations is the following: the spaces of rigid or affine transformations can be linearized around the identity in a way that is *compatible* with many algebraic properites of these Lie groups. This linearization is simply given by the group logarithm that we described in Chapter 2. Once this linearization has been carried out, usual vector processing tools can be readiliy generalized on logarithms and the result is mapped back to the transformation space by the exponential when necessary.

Log-Euclidean Distances. Then, let T_1 and T_2 be two admissible affine transformations. A *Log-Euclidean distance* (or metric) between the two transformations will be of the following form:

$$\mathrm{dist}(T_1, T_2) = \| \log(T_1) - \log(T_2)\|, \tag{6.16}$$

where $\|.\|$ is a Euclidean norm, for example the Frobenius norm, which is defined by $\|M\|_{\mathrm{Frob}} = \left(\mathrm{Trace}(M^T.M)\right)^{\frac{1}{2}}$. This is the norm we will be using in the rest of this Chapter.

Log-Euclidean Mean. As in the tensor case, one can associate to Log-Euclidean distances a generalization of the *arithmetic mean*, in the classical way, called the *Fréchet* mean. The Log-Euclidean mean $\mathbb{E}(T_1, ..., T_2)_{\mathrm{LE}}$ of N admissible transformations $T_1, ...T_N$ with non-negative weights $w_1, ..., w_N$ (such that $\sum_i w_i = 1$) is defined as the point minimizing the following *metric dispersion*:

$$\mathbb{E}(T_1, ..., T_N)_{\mathrm{LE}} = \arg \min_{\text{admissible } T} \sum_{i=1}^{N} w_i \, \mathrm{dist}^2(T, T_i).$$

As in the tensor case, classical Euclidean geometry on the principal logarithms of the transformations show that the Log-Euclidean mean is indeed well-defined for transformations close enough to a scaled version of the identity. In this case, it is simply given by exponential of the arithmetic mean of data:

$$\mathbb{E}(T_1, ..., T_N)_{\mathrm{LE}} = \exp\left(\sum_{i=1}^{N} w_i . \log(T_i)\right). \tag{6.17}$$

For admissible transformations very far away from the identity, it might be possible that the arithmetic mean of their logarithms lies *outside* the logarithmic domain of admissible transformations! To our knowledge, although the set of complex numbers whose imaginary part lies in $]-\pi, \pi[$ is obviously convex, the set of real matrices whose eigenvalues are all in this domain is only *open*, and *not convex*. This set of matrices is only *locally* convex.

In practice, to check the well-definedness of the Log-Euclidean mean, it suffices to check that the Euclidean mean of the logarithms is still admissible. However, having a simple and general criterion on the T_i ensuring that all convex combinations of their logarithms are admissible would be very desirable, and this will be the subject of future work.

The Log-Euclidean Mean as a Geometric Mean. Exactly as in the tensor case, the determinant of the homogeneous representation of the Log-Euclidean mean is equal to the scalar *geometric mean* of the determinants of the data. In other words, we have:

$$\det(\mathbb{E}(T_1, ..., T_N)) = \exp\left(\sum_i w_i \ln\left(\det(T_i)\right)\right),$$

where ln is the (scalar) natural logarithm.

This can be shown with the same techniques as for tensors (well, actually nothing more than Jordan or Schur decompositions of matrices), see Section 3.4.3 for more details. The Log-Euclidean mean is therefore a generalization of the scalar geometric mean to affine transformations. The determinant of affine transformations has a very simple physical interpretation: it describes how volumes are changed by the affine transformation. Above 1, the transformation dilates volumes, and below 1, there is a contraction. The geometric interpolation of the determinants of the data performed by Log-Euclidean averaging thus guarantees that determinants are monotonically interpolated. This is not true for the arithmetic mean of affine-transformation, since Euclidean operations do not take into account the group structure of affine transformations. The arithmetic mean of several affine transformations can for example introduce more dilation than there originally was in the data, which is the 'swelling effect' well-known in the tensor case [Arsigny 05a].

6.4.2 Invariance Properties

Invariance by Inversion. It can be easily seen that any Log-Euclidean distance on affine transformations is invariant by *inversion*: the principal logarithm of an inverted transformation is simply the logarithm of the original transformation multiplied by -1, which does not change the value the distance (see Eq. (6.16)).

Rotational Invariance. A number of Log-Euclidean distances on affine transformations are *rotation-invariant*. This means that when the coordinate system is changed by a rotation, the Log-Euclidean distance between two affine transformations is unchanged. To see this, let us recall how an affine change of coordinates affects affine transformations. Let T be an affine transformation used to deformed locally the ambient space and let A be another affine transformation, used this time to change the coordinate system. When the point x is changed into $A.x$, then the affine transformation T is classically changed into $A.T.A^{-1}$. The principal logarithm of $A.T.A^{-1}$ is simply $A.\log(T).A^{-1}$. Using this equation, we get in homogeneous coordinates:

$$\log(A.T.A^{-1}) \sim \begin{pmatrix} M_2.L.M_2^{-1} & -M_2.L.M_2^{-1}.t_2 + M_2.v \\ 0 & 0 \end{pmatrix}, \tag{6.18}$$

where $A = (M_2, t_2)$ and where L and v are respectively the linear part and the translation part of $\log(T)$. As consequence, when M_2 is a rotation matrix R and when t_2 is equal to zero, we get:

$$\|\log(A.T.A^{-1})\|^2_{\text{Frob.}} = \|R.L.R^T\|^2_{\text{Frob.}} + \|R.v\|^2_{\text{Eucl.}} = \|L\|^2_{\text{Frob.}} + \|v\|^2_{\text{Eucl.}} = \|\log(T)\|^2_{\text{Frob.}},$$

which precisely means that this Log-Euclidean distance is rotation-invariant.

And Translation-Invariance? How does the choice of the *origin* of the coordinate system have an impact on Log-Euclidean metrics? To see this, let us re-write Eq. (6.18) in the case where the affine change of coordinate system is a pure translation and writes $A = (Id, t_2)$. Then Eq. (6.18) becomes:

$$\log(A.T.A^{-1}) \sim \begin{pmatrix} L & -L.t_2 + v \\ 0 & 0 \end{pmatrix},$$

which implies in turn that we have:

$$\|\log(A.T.A^{-1})\|_{\text{Frob.}} = \|L\|^2_{\text{Frob.}} + \|v - L.t_2\|^2_{\text{Eucl.}},$$

which depends on t_2 and is not equal in general to $\| \log(T)\|^2_{\text{Frob.}}$. Changing the origin of the coordinate system by a translation t_2 results in shifting by $-L.t_2$ the translation part v of the logarithm of the current affine transformation T, which changes the norm of $\log(T)$ in general. The Log-Euclidean Frobenius distance is therefore not *translation-invariant*.

Affine-Invariance of the Log-Euclidean Mean. Although Log-Euclidean distances are not even translation-invariant in general, the Log-Euclidean mean is *affine-invariant*: it is not biased *at all* by the current coordinate system. To see this, let $T_1, ..., T_N$ be N affine transformations with logarithms $(L_1, v_1), ..., (L_N, v_N)$, and let $w_1, ..., W_N$ N be N non-negative weights. Using Eq. (6.18), we have:

$$\begin{aligned} \log(\mathbb{E}(A.T_1.A^{-1}, ..., A.T_N.A^{-1})_{\text{LE}}) &= \textstyle\sum_i w_i \log(A.T_i.A^{-1}) \\ &= \textstyle\sum_i w_i A.\log(T_i).A^{-1} \\ &= A.\left(\textstyle\sum_i w_i \log(T_i)\right).A^{-1} \\ &= A.\log\left(\mathbb{E}(T_1, ..., T_N)\right).A^{-1} \\ &= \log\left(A.\mathbb{E}(T_1, ..., T_N).A^{-1}\right) \end{aligned}$$

which implies the affine-invariance of the Log-Euclidean mean:

$$\mathbb{E}(A.T_1.A^{-1}, ..., A.T_N.A^{-1})_{\text{LE}} = A.\mathbb{E}(T_1, ..., T_N)_{\text{LE}}.A^{-1}.$$

6.4.3 Regularization of Locally Affine Transformations

One of the great advantages of the Log-Euclidean framework is that classical vector tools can be readily recycled in this framework. Once affine-transformations have been transformed into their principal logarithm, one can simply perform Euclidean operations on them. This allows the direct generalization of classical *vectorial* regularization tools to locally affine deformations, which we use in our locally affine registration algorithm [Commowick 06a, Commowick 06b].

To speed up computations in our locally affine algorithm, we estimate the affine transformations of the components in two very simple steps. First, a *separate* estimation of modifications from the block-matching results, and second a *regularization* of the resulting affine components. Compared to a *coupled* estimation of the local transformations in one step, this is much faster and yields similar results in practice.

Elastic-Like Regularization. Here, the idea is essentially to quantify the regularity of a locally affine transformation by a measure of its deviation with respect to a global affine transformation. This can be done with the following *elastic-like regularity energy*:

$$Reg(T_i, w_i) = \frac{1}{2}.\sum_{i=1}^{N}\sum_{j>i} p_{i,j} \| \log(T_i) - \log(T_j)\|^2, \tag{6.19}$$

where Ω is the (bounded) image domain chosen for the registration experiment, and where the *correlation weights* $p_{i,j}$ between the components are defined in the following way:

$$p_{i,j} = \left(\int_\Omega w_i(x).w_j(x)dx\right) \times \frac{1}{2}\left(\frac{1}{\int_\Omega w_i(x)dx} + \frac{1}{\int_\Omega w_j(x)dx}\right),$$

One can show that this non-negative energy is equal to zero if and only if only components are equal, i.e. if the global transformation is affine.

This type of energy takes into account the spatial extensions of the components and their spatial relationships. Thus, the differences between two components with a large 'overlap' (i.e., a large correlation weight) will be more penalized than components 'far apart' (i.e., with a small $p_{i,j}$).

Elastic-Like Regularization and Log-Euclidean Means. From a practical point of view, to regularize the current locally affine transformation, we perform in [Commowick 06a, Commowick 06b] a gradient descent of the elastic-like energy given in (6.19). The partial derivative of this energy yields:

$$\frac{\partial}{\partial \log(T_k)} Reg(T_i, w_i) = \sum_{j \neq k} p_{j,k} \left(\log(T_k) - \log(T_j)\right).$$

As a consequence, performing a gradient descent of our elastic-like energy simply results in replacing affine transformations T_i by Log-Euclidean means between all affine transformations, the weights depending on the correlation weights $p_{i,j}$ and on the time step used (which should be small enough to ensure that the weights used are all non-negative). This guarantees the *affine-invariance* of our regularization approach, since performing Log-Euclidean means of affine transformations is an affine-invariant operation. This was not obvious at all, since the Log-Euclidean metric we use is only *rotation-invariant*, and *not* affine-invariant.

Extensions to Other Types of Regularizations. Since usual vectorial processing tools are easily recycled in the Log-Euclidean framework, other types of regularization techniques, such as *fluid-like regularization* (i.e. elastic-like regularization on the small modifications made to the components between the iterations of the registration algorithm), can be generalized to locally affine transformations in the same straightforward way as elastic-like regularization. In [Commowick 06a, Commowick 06b] for example, we use *visco-elastic* regularization (i.e. elastic-like *and* fluid-like regularization), exactly as was done in [Stefanescu 05] in the dense-transformation algorithm case.

Synthetic 2D Experiment. To illustrate the effect of our locally affine elastic regularization, let us regularize a synthetic polyaffine transformation using the Log-Euclidean elastic-like regularization energy of Eq. (6.19). 9 components were regularly defined on the grid, and their affine transformations were chosen randomly. A gradient descent on the energy (6.19) was performed, and the result is shown in Fig. 6.17. Note how the 9 components all converge toward the same global affine transformation as the degree of regularization increases.

6.4.4 Log-Euclidean Framework for General Lie Groups

In this Section, we have presented the Log-Euclidean frameworks for rigid or affine transformations. Of course, it can be extended to any Lie subgroup of invertible matrices (e.g., $GL(n)$ itself, $SL(n)$, etc.). In fact, such an extension is also available for *any finite-dimensional Lie group*. Indeed, since Lie algebras are by definition vector spaces, one can always perform vectorial operations on data living in a Lie group via their group logarithm. Of course, this is only entirely justified only *close to the identity*, where the group logarithm is well-defined.

By construction, this framework is *inversion-invariant*. Furthermore, the Log-Euclidean mean is *invariant with respect to the adjoint representation*: if the data (x_i) is shifted in the following fashion: $(x_i) \mapsto (m.x_i.m^{-1})$, so is their Log-Euclidean mean; this property is the abstract equivalent of the affine-invariance of the Log-Euclidean mean of affine transformations. This is a simple consequence of the properties of the logarithm and the exponential presented in the Lemma 7.1 of Chapter 7.

In the matrix case, the general properties of the Log-Euclidean framework are of course valid. There is in this case one extra property in terms of determinants: the determinant of the Log-Euclidean mean is equal to the geometric mean of the determinants of the data. The Log-Euclidean mean is therefore a generalization to invertible matrices of the (scalar)

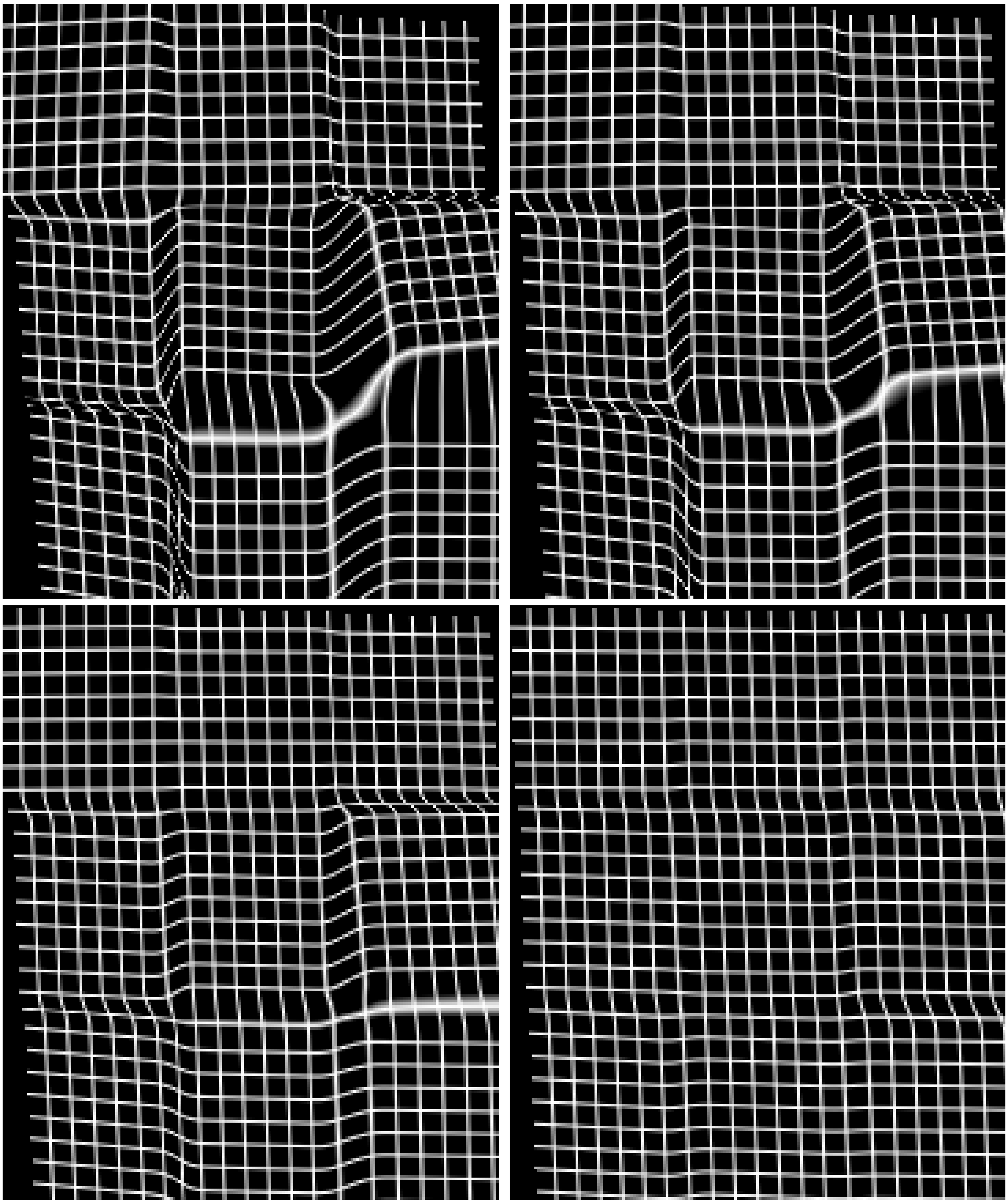

Figure 6.17: **Log-Euclidean regularization of a polyaffine transformation. From left to right and then from top to bottom:** regular grid deformed by original polyaffine transformation (with 9 affine components), the same regular grid deformed by the increasingly regularized polyaffine transformation. Note how the 9 components all converge toward a Log-Euclidean mean of the original affine transformations as the regularization increases.

geometric mean of positive numbers. Interestingly, we will see in Chapter 7 that this is not the only possible generalization: the bi-invariant mean presented therein also has this property (see Section 7.6.3 for more details on this subject).

6.4.5 Numerical Implementation of Matrix Logarithm.

In this work, we have used the 'Inverse Scaling and Squaring' method [Cheng 01] to compute matrix logarithms. This method, similarly to the 'Scaling and Squaring' method in the exponential case, is based on the idea that computing the logarithm of a matrix *close to the identity* is easier than computing the logarithm of a general matrix. Like in the case of the matrix exponential, this computation can be done very accurately and at a very small computational cost using Padé approximants.

In order to transform a matrix into another matrix closer to the identity, the 'Inverse Scaling and Squaring' method uses the computation of successive *square roots*. Once the $2^{N^{\text{th}}}$ root of a matrix M has been computed, one can use the following equality to compute the logarithm of M:

$$\log(M) = 2^N . \log\left(M^{2^{-N}}\right). \tag{6.20}$$

Actually, (6.20) is nothing more than Eq. (6.12) in the domain of logarithms.

More details on how square roots can be iteratively computed and on the choice of the level of squarings N can be found in [Cheng 01].

6.5 Conclusion and Perspectives

In this Chapter, we have presented a novel framework to fuse rigid or affine components into a global transformation, called *Log-Euclidean polyaffine*. Similarly to the previous polyaffine framework of [Arsigny 05c], it guarantees the *invertibility* of the result. However, this is achieved with more intuitive properties than previously: for example the inverse of a LEPT is a LEPT with identical weights and inverted affine components. Moreover, this novel fusion is *affine-invariant*, i.e. does not depend on the choice of coordinate system. We have also shown that the specific properties of LEPTs allow their fast computations on regular grids, with an algorithm called the 'Fast Polyaffine Transform', whose efficiency is somehow comparable to that of the Fast Fourier Transform.

In the example of locally affine 3D registration presented here, we use LEPTs in a final step to fuse the affine components estimated during the algorithm of [Commowick 06a]. With the FPT, this is done very efficiently. Remarkably, the novel fusion is *very close* to the direct fusion in regions without singularities. This suggests that our novel framework provides a general and efficient way of fusing local rigid or affine deformations into a global invertible transformation without introducing artifacts, *independently* of the way local affine deformations are first estimated.

We have also presented in this Chapter a Log-Euclidean framework for rigid and affine transformations, as well as its extension to any finite-dimensional real Lie group. This framework generalizes to linear transformations the Log-Euclidean framework which is described in Chapter 3 for tensors. Interestingly, this framework allows a straightforward and efficient generalization to linear transformations of classical vectorial tools, with excellent theoretical properties. In particular, we have already used this framework in [Commowick 06a] to define a simple visco-elastic regularization energy for locally rigid or affine deformations. In future work, we are planning to use this simple framework to compute statistics on rigid and affine local components of deformations, which could help to better constraint non-rigid registration algorithms, as some of us begun to do in [Pennec 05] and [Commowick 05] with (local) statistics on Cauchy-Green deformation tensors.

Remarkably, we have seen in this Chapter that the Log-Euclidean mean of linear transformations can be seen as a generalization to invertible linear transformations of the (scalar) geometric mean of positive number. This comes from the fact that the determinant of the mean is equal to the geometric mean of the data. In Chapter 7, we will see that at least another such generalization exists, called the *bi-invariant mean*.

Chapter 7

Bi-Invariant Means in Lie Groups

In the previous Chapter, we relied implicitly on the Log-Euclidean mean of linear transformations to obtain intuitive invariance properties for polyaffine transformations. This type of mean generalizes to rigid and affine transformations the geometric mean of positive number, in a way which is *invariant by inversion*.

In this Chapter, we define another general notion of mean in finite-dimensional Lie groups (e.g., the rigid or affine groups) which is this time *fully compatible* with the algebraic structure of these groups, contrary to the Log-Euclidean mean which is only invariant by inversion and invariant with respect to the action of the adjoint representation. Indeed, the *bi-invariant mean* presented below generalizes to any Lie group the invariance properties of the arithmetic mean. Interestingly, we do not rely on Riemannian geometry but on the general algebraic properties of Lie groups to define this mean. In fact, going beyond Riemannian metrics was unavoidable: we prove in this Chapter that no bi-invariant Riemannian metric exist for rigid transformations, in any dimension.

Finally, we use the bi-invariant mean to define a last class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components *arbitrarily far away from the identity.*

Contents

7.1 Introduction **135**

7.2 Means in Lie Groups **136**

7.2.1 Means and Algebraic Invariance 136

7.2.2 Bi-Invariant Fréchet Means via Invariant Metrics in Lie Groups . . . 137

7.2.3 Absence of Bi-Invariant Metrics for Rigid Transformations 139

7.3 Advanced Properties of the Exponential and Logarithm **141**

7.3.1 Preliminary Result 141

7.3.2 Group Geodesics. 141

7.4 Bi-Invariant Means in Lie Groups **143**

7.4.1 A Geometric Definition of the Mean 143

7.4.2 Stability of the Classical Iterative Scheme 144

7.4.3 Convergence: Special Case 146

7.4.4 Convergence: General Case 147

7.4.5 Higher Order Moments 148
7.5 Bi-Invariant Means in Simple Cases 149
7.5.1 Bi-Invariant Mean of Two Points 149
7.5.2 Scalings and Translations in 1D 150
7.5.3 The Heisenberg Group 152
7.5.4 On a Subgroup of Triangular Matrices 154
7.6 Linear Transformations 155
7.6.1 General Rigid Transformations 155
7.6.2 2D Rigid Transformations 159
7.6.3 General Linear Transformations 160
7.6.4 Tensors 161
7.7 Left-Invariant Polyaffine Transformations 161
7.7.1 Polyaffine Transformations 161
7.7.2 A Novel Type of Polyaffine Transformations 162
7.8 Conclusions and Perspectives 165

Abstract. In this Chapter, we present a general framework to define rigorously a novel type of mean in Lie groups, called the *bi-invariant mean*. This mean enjoys many desirable *invariance properties*, which generalize to the non-linear case the properties of the arithmetic mean: it is invariant with respect to left- and right-multiplication, as well as inversion. Previously, this type of mean was only defined in Lie groups endowed with a bi-invariant Riemannian metric, like compact Lie groups such as the group of rotations. But Riemannian bi-invariant metrics do not always exist. In particular, we prove in this work that such metrics do not exist in any dimension for *rigid transformations*, which form but the most simple Lie group involved in bio-medical image registration.

To overcome the lack of existence of bi-invariant Riemannian metrics for many Lie groups, we propose in this thesis to define bi-invariant means in *any finite-dimensional real Lie group* via a general *barycentric equation*, whose solution is by definition the bi-invariant mean. The existence and uniqueness of this novel type of mean is shown, provided the dispersion of the data is small enough. The convergence of an efficient iterative algorithm for computing this mean is also shown. The intuition of the existence of such a mean was first given by R.P.Woods (without precise definition) in the case of *matrix* groups [Woods 03]. Moreover, we briefly present how empirical higher order moments can be computed based on this novel notion of mean.

In the case of rigid transformations, we give a simple criterion for the general existence and uniqueness of the bi-invariant mean, which happens to be the same as for rotations. We also give closed forms for the bi-invariant mean in a number of simple but instructive cases, including 2D rigid transformations. For general linear transformations, we show that similarly to the Log-Euclidean mean the bi-invariant mean is a generalization of the (scalar) geometric mean, since the determinant of the bi-invariant mean is exactly equal to the geometric mean of the determinants of the data.

Finally, we use this new type of mean to define a novel class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components *arbitrarily far away from the identity*, contrary to Log-Euclidean polyaffine fusion, which we are presented in Chapter 6.

Related Publications. Most of this Chapter was published in an INRIA research report [Arsigny 06g].

7.1 Introduction

As we have seen in the Introduction of this thesis, in Section 1.1.3, the need for rigorous frameworks to compute statistics in non-linear spaces has grown considerably in the bio-medical imaging community in recent years.

Among statistics, the most fundamental is certainly the *mean*, which extracts from the data a central point, minimizing in some sense the dispersion of the data around it. In this Chapter, we focus on the generalization of the *Euclidean mean* to Lie Groups, which are a large class of non-linear spaces with relatively nice properties. Classically, in a Lie group endowed with a Riemannian metric, the natural choice of mean is called the *Fréchet mean* [Pennec 06a]. But this Riemannian approach is completely satisfactory *only* if a bi-invariant metric exists, which is for example the case for compact groups such as rotations [Pennec 06a, Moakher 02]. The bi-invariant Fréchet mean enjoys many desirable *invariance properties*, which generalize to the non-linear case the properties of the arithmetic mean: it is invariant with respect to left- and right-multiplication, as well as inversion. Unfortunately, bi-invariant Riemannian metrics do not always exist. In particular, in this work, we prove the novel result that such metrics do not exist in any dimension for *rigid transformations*, which form but the most simple Lie group involved in bio-medical image registration.

To overcome the lack of existence of bi-invariant Riemannian metrics for many Lie groups, we propose in this thesis to define a *bi-invariant mean* generalizing the Fréchet mean induced by bi-invariant metrics, even in cases when such metrics *do not* exist. The intuition of the existence of such a mean was actually first given in [Woods 03] (without precise definition), along with an efficient algorithm for computing it (without proof of convergence), in the case of *matrix* groups.

In this work, we present a general framework to define rigorously bi-invariant means, this time in *any* finite dimensional real Lie group. To do this, we rely on a general *barycentric equation*, whose solution is by definition the bi-invariant mean. The existence and uniqueness of this novel type of mean is shown, provided the dispersion of the data is small enough. The convergence of the iterative algorithm of [Woods 03] is also shown. Moreover, we briefly present how empirical higher order moments can be computed based on this novel notion of mean.

In the case of rigid transformations, we have been able to determine a simple criterion for the general existence and uniqueness of the bi-invariant mean, which happens to be the same as for rotations. We also give closed forms for the bi-invariant mean in a number of simple but instructive cases, including 2D rigid transformations. Interestingly, for general linear transformations, we show that similarly to the Log-Euclidean mean, the bi-invariant mean is a generalization of the (scalar) geometric mean, since the determinant of the bi-invariant mean is exactly equal to the geometric mean of the determinants of the data.

Finally, this new type of mean is used to define a novel class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components *arbitrarily far away from the identity*, contrary to Log-Euclidean polyaffine fusion.

The sequel of this Chapter is organized as follows. First, we detail some fundamental notions and properties about means and invariant Riemannian metrics in Lie groups. Using these properties, we prove that bi-invariant Riemannian metrics do not exist for rigid transformations. Then, we detail some advanced properties of the exponential and logarithm in Lie groups not described in Chapter 2. In the next Section, we rely on these properties to obtain a novel definition of bi-invariant means in any finite-dimensional real Lie groups, along with a proof of its existence and uniqueness and we also prove the convergence of the efficient iterative scheme proposed in [Woods 03] to compute this mean in practice. Then, we explicit the form taken by the bi-invariant mean in a number of simple cases where a closed form exists for this mean, e.g. the Heisenberg group. Afterwards, we focus on linear transformations, and in particular on rigid transformations and tensors. Before concluding, we rely on bi-invariant means to define a novel class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components *arbitrarily far away from the identity.*

7.2 Means in Lie Groups

Please refer to Chapter 2, Section 2.3 for a short presentation of Lie groups and of their fundamental properties.

7.2.1 Means and Algebraic Invariance

Lie groups are *not* vector spaces in general but have a more complicated structure: instead of a (commutative) addition '+' and a scalar multiplication '.', they only have a (non-commutative in general) multiplication '×' and an inversion operator (which corresponds to the scalar multiplication by -1 for vector spaces).

In order to generalize to Lie groups the invariance properties of the arithmetic mean, one can rely on the *invariance properties* that the mean should *a priori* satisfy.

Indeed, in the case of vector spaces, the arithmetic means presents strong invariance properties: invariance with respect to any translation and with respect to any multiplication by a scalar. This means that the arithmetic mean is invariant with respect to all the algebraic operations induced by the vector space structure. It makes good sense that the notion of mean and the algebraic structure should be *compatible.*

In the case of groups, the invariance with respect to left- and right-multiplications (the group can be non-commutative) and the inversion operator are the equivalent of the invariance properties of the mean in vector spaces. When we translate a given set of samples or a probability measure, it is reasonable to wish that their mean be translated exactly in the same way, and the same property is desirable when we take the inverses of the samples.

Example 7.1. The Geometric Mean of Positive Numbers. *We can give to the set of positive numbers a structure of commutative group with the usual scalar multiplication. In this context, let (x_i) be N positive numbers and (w_i) be N non-negative normalized weights ($\sum_i w_i = 1$). The arithmetic mean of the data is invariant with respect to multiplication, but not with respect to inversion:*

$$\Sigma_i w_i \frac{1}{x_i} \neq \frac{1}{\Sigma_i w_i x_i}, \text{ in general.} \tag{7.1}$$

Thus the arithmetic mean is not fully adapted to this multiplicative structure. On the contrary, the geometric mean, written here $\mathbb{E}((x_i),(w_i))$, is fully adapted. It is given by:

$$\mathbb{E}((x_i),(w_i)) = \exp(\Sigma_i w_i \log(x_i)). \tag{7.2}$$

We recall the classical convexity inequality between the two means:

$$\exp(\Sigma_i w_i \log(x_i)) < \Sigma_i w_i x_i, \tag{7.3}$$

whenever the data is not reduced to a single point.

Means and Distances. A well-established approach to define a notion of mean compatible with algebraic operations is to define first a *distance* (or metric) compatible with these operations and then to rely on this distance to define the mean [Pennec 06a].

We have seen in Section 2.4.4 that in the general setting of *metric spaces*, one can generalize the classical notion of arithmetic mean by relying on the intuitive idea of *minimal variance or dispersion*. The mean can be defined as the point $\mathbb{E}(X)$ which minimizes a metric dispersion with respect to the data $(X_i)_{i=1}^N$ and the non-negative weights (w_i) in the following way:

$$\mathbb{E}(X_i) = \underset{Y \in E}{arg\ min} \sum_i w_i.dist(X_i, Y)^\alpha. \tag{7.4}$$

The case $\alpha = 2$ corresponds in vector spaces to the arithmetic mean, and in more general (non-linear) spaces we recall that this provides the *Fréchet* mean.

7.2.2 Bi-Invariant Fréchet Means via Invariant Metrics in Lie Groups

In the case of Lie groups, we will see here how one can (or cannot) define a distance compatible with algebraic operations *and* the differentiable structure of these groups. We recall that the distances (or metrics) *compatible* with the differentiable structure of differentiable manifolds are called *Riemannian metrics*. See Chapter 2, Section 2.4 for a short presentation of this general type of metrics.

Invariance Properties of Riemannian Metrics. Let us now detail the different types of invariance (or compatibility) that can exist between a Riemannian metric on a Lie group and its algebraic properties. They are the following:

- 'left-invariance': the metric is invariant with respect to any multiplication on the left. Another useful way of phrasing this is to say that left-multiplications are *isometries* of $\mathcal{G}$, i.e. do not change distances between elements of $\mathcal{G}$.

  In terms of scalar products and differentials, this means precisely that for any two points m and h of $\mathcal{G}$ and any vectors v and w of $T_m\mathcal{G}$, we have: $< D_m L_h.v, D_m L_h.w >_{T_{h.m}\mathcal{G}} = < v, w >_{T_m\mathcal{G}}$.
- 'right-invariance': invariance with respect to any multiplication on the right.
- 'inversion-invariance': invariance with respect to inversion. The inversion operator is then an isometry of $\mathcal{G}$.

These properties are *not* independent. This simply comes from the fact that for any two elements m, n of $\mathcal{G}$, we have $(m.n)^{-1} = n^{-1}.m^{-1}$. This implies for example that the left-multiplication can be obtained smoothly from one right-multiplication and two inversions in the following way: $L_m = Inv \circ R_{m^{-1}} \circ Inv$.

A simple (but rarely mentioned in classical references on Lie groups) consequence of this is that *all* right-invariant metrics can be obtained from left-invariant metrics by 'inversion', and *vice versa*. Indeed, we have:

Proposition 7.1. *Let $<,>$ be a left-invariant Riemannian metric defined on $\mathcal{G}$. Then the 'inverted' metric $\ll,\gg$, defined below, is right-invariant, and moreover we have $<,>_e = \ll,\gg_e$.*

For any two points m and h of $\mathcal{G}$ and any vectors v and w of $T_m\mathcal{G}$, we define the inverted metric $\ll,\gg$ as follows:

$$\ll v, w \gg_m \stackrel{def}{=} < D_m Inv.v, D_m Inv.w >_{T_{m^{-1}}\mathcal{G}} .$$

Proof. Actually, the proof relies only on differentiating the equality $(h.m)^{-1} = m^{-1}.h^{-1}$. This yields:

$$D_{h.m}Inv \circ D_m L_h = D_{m^{-1}} R_{h^{-1}} \circ D_m Inv.$$

This allows to show directly that:

$$\ll D_m R_h.v, D_m R_h.w \gg_{T_{h.m}\mathcal{G}} = \ll v, w \gg_{T_m \mathcal{G}},$$

which means that $\ll, \gg$ is right-invariant.

Last but not least, the equality $<,>_e = \ll, \gg_e$ comes from the fact that quite intuitively $D_e Inv = -Id$, where Id is the identity operator in $T_e\mathcal{G}$. This can be easily seen from the classical result $D_e \exp = D_0 \log = Id$ and the equality (valid in an open neighborhood of e) $m^{-1} = \exp(-\log(m))$, where exp and log are the group exponential and logarithm, presented in detail in Chapter 2. □

Corollary 7.1. *'Left-invariance' (resp. 'right-invariance') and 'inversion-invariance' imply 'right-invariance' (resp. 'left-invariance').*

Proof. We have just seen in Proposition 7.1 that right-invariant metrics can be obtained by composition between left-invariant metrics and the inversion operator, and vice versa for left-invariant metrics. If left-multiplications and inversion are isometries, so are right-multiplications by composition. □

Riemannian metrics which are simultaneously left- and right- invariant are called *bi-invariant*. On these special metrics, we have the very interesting result:

Theorem 7.1. *Bi-invariant metrics have the following properties:*

1. *A bi-invariant metric is also invariant w.r.t. inversion*
2. *It is bi-invariant if and only if for all $m \in \mathcal{G}$, the adjoint operator $Ad(m)$ (see definition in Section 2.3.3) is an isometry of the Lie algebra $\mathfrak{g}$*
3. *One-parameter subgroups of $\mathcal{G}$ are geodesics for the bi-invariant metric*

Proof. See [Sternberg 64], chapter V. □

From this result and Proposition 7.1, we see that *any two invariance properties imply the third.*

Bi-Invariant Means. We have seen that a metric structure induces a notion of mean called the Fréchet mean. The Fréchet mean associated to a bi-invariant metric is called the *bi-invariant mean*. Actually, it does not depend on the particular choice of bi-invariant metric, since whenever the bi-invariant mean is uniquely defined, it is given as the solution of a *barycentric equation* [Pennec 06a] which is independent from the arbitrary choice of bi-invariant metric.

Since the metric inducing the notion of mean is bi-invariant, so is the mean, which is then fully compatible with the algebraic properties of the Lie group. As a consequence, this notion of mean is particularly well-adapted to Lie groups [Pennec 06a]. However, contrary to left- or right- invariant metrics, which always exist[1], bi-invariant metrics may fail to exist, and we will now see under which conditions bi-invariant metrics exist for a given Lie group.

[1]It suffices to propagate an arbitrary scalar product defined on $T_e\mathcal{G}$ to all tangent spaces by left- or right-multiplication to generate all left- or right-invariant Riemannian metrics.

Compactness of the Adjoint Representation From Theorem 7.1, we see that if a bi-invariant metric $< .,. >$ exists for the Lie group $\mathcal{G}$, then $\forall m \in \mathcal{G}$, $Ad(m)$ is an isometry of $\mathfrak{g}$ and can thus be looked upon as an element of the orthogonal group $O(n)$ where $n = dim(\mathcal{G})$. Then, note that $O(n)$ is a *compact* group, and that therefore $Ad(\mathcal{G})$ is necessarily *included in a compact set*, a situation called *relative compactness*. This notion provides indeed an excellent criterion, since we have:

Theorem 7.2. *The Lie group $\mathcal{G}$ admits a bi-invariant metric if and only if its adjoint representation $Ad(\mathcal{G})$ is relatively compact.*

Proof. We have already seen the first implication. For the converse part, the theory of differential forms and their integration can be used to explicitly construct a bi-invariant metric. This is done in [Sternberg 64], Theorem V.5.3. □

Compactness, Commutativity and Bi-Invariant Metrics. In the case of *compact* Lie groups, we have the property that their adjoint representation is the image of a compact set by a continuous mapping and is thus also compact. Then, Theorem 7.2 implies that bi-invariant metrics exist in such a case. In particular, this is the case of *rotations*, for which bi-invariant means have been extensively studied and used [Pennec 06a]. This is also trivially the case of *commutative* Lie groups, where only one type of multiplication exists, which reduces the adjoint representation to $\{Id\}$. An illustration of this situation is given by the Lie group structure on symmetric positive-definite matrices we present in Chapter 3.

As shown by Theorem 7.2, the general non-compact and non-commutative case is not so nice, and one has to carefully check the properties of the adjoint representation of the Lie group to see whether a bi-invariant metric exists or not. This verification has to be done all the more carefully that non-commutativity and non-compactness of the group are necessary but *not* sufficient to prevent the existence of bi-invariant metrics, as shown in the following paragraph.

An Example in the Non-Compact and Non-Commutative Case. We have already seen that any compact or commutative Lie group has at least one bi-invariant metric. From this remark, one can easily construct an example of non-compact and non-commutative group having a bi-invariant metric: let $\mathcal{G}_1$ be a commutative non-compact group and $\mathcal{G}_2$ be a compact non-commutative group. They both have a bi-invariant metric. Let $\mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2$ be their *direct product*, i.e. the group obtained with the multiplication $(g'_1, g'_2).(g_1, g_2) = (g'_1.g_1, g'_2.g_2)$. Then $\mathcal{G}$ is neither commutative nor compact, but *has* a bi-invariant metric! In fact, let $< .,. >_1$ and $< .,. >_2$ be respectively a bi-invariant metric of $\mathcal{G}_1$ and $\mathcal{G}_2$. Then $< P_{\mathcal{G}_1}(.), P_{\mathcal{G}_1}(.) >_1 + < P_{\mathcal{G}_2}(.), P_{\mathcal{G}_2}(.) >_2$ is a bi-invariant metric of $\mathcal{G}$, where $P_{\mathcal{G}_i}$ is the canonical projection on $\mathcal{G}_i$.

One typical example of such a situation is the Lie group of matrices of the form $s.R$, where s is a positive scalar and R a rotation matrix (group of rotations and scalings). It can be seen as the direct product of $(\mathbb{R}^{\star}_{+}, \times)$ (commutative and non-compact) with $(SO(n), \times)$ (compact and non-commutative).

7.2.3 Absence of Bi-Invariant Metrics for Rigid Transformations

As we have seen in the previous Subsection, bi-invariant metrics always exist for compact groups, which is the case of rotations. But when one tries to extend the use of bi-invariance metrics to more general transformation groups, one is very limited. In biomedical imaging, the simplest possible registration procedure between two anatomies uses rigid transformations.

Such transformations seem quite close to rotations and one could hope for the existence of bi-invariant metrics. But we have the following general result, which is new to our knowledge[2]:

Proposition 7.2. *The action of the adjoint representation Ad of the group of rigid transformations $SE(n)$ at the point (R, t) on an infinitesimal displacement (dR, dt) is given by:*

$$Ad(R,t).(dR,dt) = (R\,dR\,R^T,\ -R\,dR\,R^T\,t\ + R\,dt).$$

As a consequence, no bi-invariant Riemannian metric exists on the space of rigid transformations (for $n > 1$, of course).

Proof. In the case of matrix Lie groups, we have the following formula [Hall 03] $Ad(h).dh = h.dh.h^{-1}$ for $dh \in \mathfrak{g}$. Classically, using homogeneous coordinates, the Lie group of rigid transformations is faithfully represented by the following matrix Lie group [Sattinger 86]:

$$(R,t) \sim \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}.$$

Using this, we get:

$$Ad(R,t).(dR,dt) \sim \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}.\begin{pmatrix} dR & dt \\ 0 & 0 \end{pmatrix}.\begin{pmatrix} R^T & -R^T.t \\ 0 & 1 \end{pmatrix},$$

which yields the announced formula. In this formula, the translation 't' introduces a unbounded term which prevents the adjoint group from being bounded. Applying Theorem 7.2, it is then clear that no bi-invariant metric exists for rigid transformations in nD (n>1). □

We thus see that the Riemannian approach based on bi-invariant metrics cannot be extended to rigid transformations, and even less so to affine transformations. Other examples of non-compact and non-commutative groups with no bi-invariant metrics can be found in Section 7.5.

One should note that our result contradicts a statement in [Woods 03], which claimed that a bi-invariant metric existed when $n = 2$. The reference backing this claim was [Sattinger 86], in which it is only stated that though $SE(2)$ is non-compact, it has a bi-invariant *measure* (Chapter 7, page 92). But whereas the existence of a metric implies that of a measure (see [Pennec 06a], page 6: such a measure can be obtained via the square roof of the determinant of the metric), the existence of a measure does *not* imply the existence of a metric. This subtle mistake is of no consequence, since there truly are examples of non-compact groups which have bi-invariant metrics. As long as the group is commutative, such metrics obviously exist (think of vector spaces!).

In the sequel, we will see how it is possible to define general bi-invariant means in Lie groups *without* relying on bi-invariant Riemannian means, which can fail to exist. The key to our approach is to use the general *algebraic* properties of Lie groups, and in particular the group exponential and logarithm.

[2]We have recently found that the non-existence of bi-invariant Riemannian metrics for $SE(3)$ was already known in the literature [Zefran 99]. However, our result does not depend on the dimension and is obtained in a very economical way, using short and abstract arguments rather than long and direct computations as in [Zefran 99] for $n = 3$.

7.3 Advanced Properties of the Exponential and Logarithm

Before defining general bi-invariant means in Lie groups, we detail in this Section some advanced properties of the group exponential and logarithm, which will be very useful in the sequel. Please refer to Chapter 2, Section 2.3 for the general definition of the exponential and logarithm in Lie groups, along with their fundamental algebraic and differential properties.

7.3.1 Preliminary Result

A very useful property of the (Riemannian) exponential map is that given any point, there exists a open neighborhood of this point, called *geodesically convex*, in which for any coupe of points, there exists a unique minimizing geodesic between them (see for example [Gallot 93], page 84-85). We now prove an similar result for the group exponential:

Theorem 7.3. *Let* $\Phi : \mathcal{G} \times \mathfrak{g} \to \mathcal{G} \times \mathcal{G}$*, defined by* $\Phi(g, v) = (g, g.\exp(v))$*. Then* Φ *is always locally diffeomorphic. More precisely, for all* g *in* $\mathcal{G}$*, it defines a diffeomorphism from some open neighborhood of* $(g, 0)$ *to a open neighborhood of* (g, g)*.*

Proof. Since Φ is smooth, one can apply anew the 'Implicit Function Theorem' provided that the differential of Φ at $(g, 0)$ is invertible. To see this, note that we have:

$$\begin{cases} \frac{\partial \Phi}{\partial g}|_{(g,v)=(e,0)} = (Id, Id) \\ \frac{\partial \Phi}{\partial v}|_{(g,v)=(e,0)} = (0, D_e L_g \circ Id) = (0, D_e L_g), \end{cases}$$

where we have used the fact the property that the differential of the exponential at 0 is the identity (see Theorem 2.1). Since L_g is a diffeomorphism, its differential at e, $D_e L_g$, is always invertible. As a consequence, the differential of Φ is also always invertible, and the 'Implicit Function Theorem' applies. This proof is very similar to the proof given in [Gallot 93] to show the analogous property of the metric exponential. □

7.3.2 Group Geodesics.

Theorem 7.3 essentially shows that for every point g of $\mathcal{G}$, there exists a open neighborhood of g in which every couple of points can be joined by a unique 'group geodesic' of the form $g.\exp(t.v)$ such that $g.\exp(v) = h$. By symmetry, the same result also holds for the geodesics of the type $\exp(t.v).g$. In fact, those two types of 'group geodesic' are the same, since we have the following result:

Theorem 7.4. *For all* g *in* $\mathcal{G}$*, there exists a two open neighborhoods* $\mathcal{V}_g$ *and* $\tilde{\mathcal{V}}_g$ *of* 0 *in* $\mathfrak{g}$ *such that for all* g *in* $\mathcal{G}$ *and for all* v *in* $\mathcal{V}_g$*, there exist a unique* w *in* $\tilde{\mathcal{V}}_g$ *such that* $g.\exp(t.v) = \exp(t.w).g$ *for all* $t \in \mathbb{R}$*. More precisely,* $w = Ad(g).v$*. Moreover, in this open neighborhood of* 0*, the relationship* $g.\exp(v) = \exp(w).g$ *implies* $w = Ad(g).v$*.*

The proof of this theorem is simply based on the following relationships between the Adjoint representation, the exponential and the logarithm:

Lemma 7.1. *Let* v *be in* $\mathfrak{g}$ *and* g *in* $\mathcal{G}$*. Then we have:*

$$g.\exp(v).g^{-1} = \exp(Ad(g).v).$$

Also, for all g *in* $\mathcal{G}$*, there exists a open neighborhood* $\mathcal{W}_g$ *of* e *such that for all* m *in* $\mathcal{W}_g$*:* $\log(m)$ *and* $\log(g.m.g^{-1})$ *are well-defined and are linked by the following relationship:*

$$\log(g.m.g^{-1}) = Ad(g).\log(m).$$

These equations are simply the generalization to (abstract) Lie groups of the well-known matrix properties: $G.\exp(V).G^{-1} = \exp(G.V.G^{-1})$ and $G.\log(V).G^{-1} = \log(G.V.G^{-1})$.

Proof. The first relationship of this Lemma can be proved in the following way: $(g.\exp(t.v).g^{-1})_t$ is a continuous one-parameter subgroup, whose infinitesimal generator is $\frac{d}{dt}g.\exp(t.v).g^{-1}|_{t=0} = Ad(g).v$ (see Proposition 1.81 on page 29 of [Gallot 93]). Using the fact that continuous one-parameter subgroups are of the form $\exp(t.w)$, we obtain the first equality. To prove the second (and this time *local*) equality, we see that since $\Psi_g : m \mapsto g.m.g^{-1}$ is smooth and $\Psi_g(e) = e$, there exists a open neighborhood of e where $\log(m)$ and $\log(g.m.g^{-1})$ are well-defined. Then the second equality is deduced from the first. □

Proof. Proof of Theorem 7.4: just see that $g.\exp(t.v) = g.\exp(t.v).g^{-1}.g$ and apply Lemma 7.1.

□

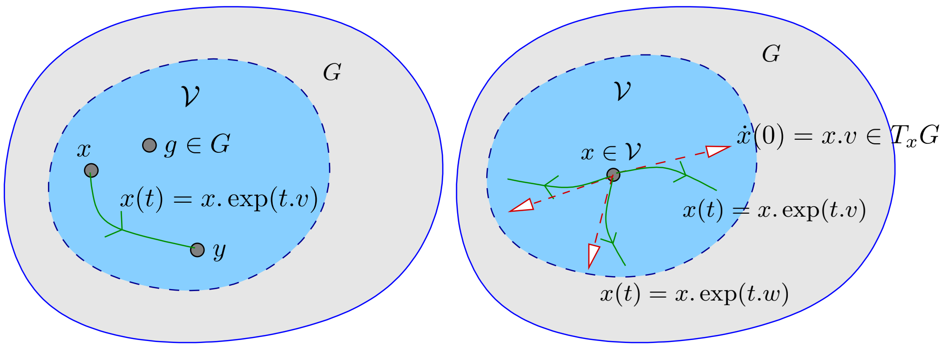

Figure 7.1: **Properties of Group Geodesics. Left:** group geodesic convexity, which means that for any point g of $\mathcal{G}$, there exists a open neighborhood $\mathcal{V}$ of g, such that any couple of points x and y in $\mathcal{V}$ can be joined by a unique group geodesics of the form $x(t) = x.\exp(t.v)$ satisfying $x.\exp(v) = y$. Note that the geodesic is entirely contained in $\mathcal{V}$. **Right:** at any point g, a unique group geodesic is associated to any initial velocity vector, provided this speed is small enough.

Definition of Group Geodesics. Essentially, we have just shown that the exponential and its translated versions can be looked upon as some sort of 'group geodesic' in a Lie group. Any couple of points can be joined by a unique 'group geodesic', provided they are close enough. This leads to the following definitions:

Definition 7.1. *Any continuous path of $\mathcal{G}$ of the form $g.\exp(t.v)$, which has the property of Theorem 7.4 (i.e. v is small enough) is called a* **group geodesic**. *Furthermore, an open set $\mathcal{O}$ of $\mathcal{G}$ is called* **groupwise geodesically convex** *(or GGC) if and only if any couple of points of $\mathcal{O}$ can be joined by a group geodesic. We have just shown that every g in $\mathcal{G}$ has a groupwise geodesically convex open neighborhood.*

To conclude this subsection, let us now present a last property for group geodesics, which generalizes the other well-known property of (metric) geodesics. The properties of group geodesics are illustrated in Fig. 7.1.

Proposition 7.3. *Let g be in $\mathcal{G}$ and z be in the tangent space at g. Then there exists a unique smooth path of the form $g.\exp(t.v)$ that $g(0) = g$ and $\frac{d}{dt}g|_{t=0} = z$. When z is small enough, this smooth path is a group geodesic.*

Proof. The only possible choice is $v = D_g L_{g^{-1}}.z$, since $D_g L_{g^{-1}}$ is always invertible. □

7.4 Bi-Invariant Means in Lie Groups

7.4.1 A Geometric Definition of the Mean

Let us recall the classical definition of a mean in an *affine* space F, i.e. a space of *points*, associated to a *vector space* E such that to any couple of points M, N we can associate the vector $\overrightarrow{MN}$, which is simply the difference between the two points: $M + \overrightarrow{MN} = N$. In this context, the *barycenter* (or mean) of a system of points $(X_i)_{i=1..n}$ associated to the non-negative normalized weights (w_i) ($\sum_i w_i = 1$) is the unique point M that verifies the following equation, called *barycentric*:

$$\Sigma_i w_i \overrightarrow{MX_i} = \overrightarrow{0}. \tag{7.5}$$

This equation means geometrically that M is the mean of the (X_i) with respect to the *weights* (w_i). Since F is a flat space, we can get a closed form for M:

$$M = X_1 + \Sigma_i w_i \overrightarrow{X_1 X_i}. \tag{7.6}$$

This kind of mean or averaging procedure is the direct generalization in the affine case of the arithmetic mean of real numbers. It gives a *geometrical* interpretation to the weighted mean: at the mean, the sum of the weighted displacements to each of the sample points is null, i.e. *the mean is at the center of the data* (with respect to the weights).

Fréchet Means and Barycenters. The Fréchet mean m of N points (x_i) with respect to the non-negative normalized weights w_i which is induced by a Riemannian metric on a manifold is defined *implicitly* by the following barycentric equation [Pennec 06a]:

$$\sum_{i=1}^{N} w_i \log_m (x_i) = 0, \tag{7.7}$$

where $\log_m$ is the logarithmic map at the point m, which is defined only locally around this point. In the particular case of bi-invariant metrics, this equation is written as:

$$\sum_{i=1}^{N} w_i \log(m^{-1}.x_i) = 0, \tag{7.8}$$

where this time log is the inverse of the *group* exponential, defined locally around the neutral element e: the (metric) logarithmic map is expressed simply in term of group logarithm. Eq. (7.8) has particularly nice invariance properties: left-, right- and inverse-invariance, since it derives from a bi-invariant metric. One should note that Eq. (7.7) (and (7.8) in the bi-invariant case) provides a *geometrical* definition of the mean, exactly as in the case of affine spaces. The Fréchet mean is defined as a *barycenter*, i.e. the element positioned at the center of the data in a *vectorial* sense. This situation is illustrated in Fig. 7.2.

The key idea developed in this Section in the following: although bi-invariant metrics may fail to exist, the group logarithm *always* exists in a Lie group and one can try to define a bi-invariant mean *directly* via Eq. (7.8). As will be shown in the next subsections, this equation has all the desirable invariance properties, even when bi-invariant metrics do not exist.

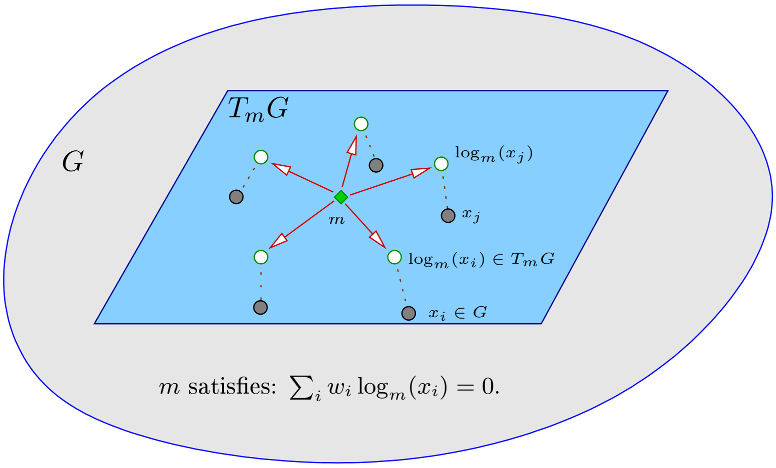

Figure 7.2: **Geometric Property of the Fréchet mean**. When well-defined, the Fréchet mean of a set of points (x_i) with non-negative (normalized) weights (w_i) satisfies a *barycentric equation*. This has a geometric interpretation: in the tangent space at the mean m, 0 (i.e. m) is precisely the barycenter of the vectors $\log_m(x_i)$ associated to the weights w_i. In this geometrical sense, m is at the center of the points x_i.

7.4.2 Stability of the Classical Iterative Scheme

To compute Fréchet means associated to Riemannian metrics, a very efficient iterative strategy can be used to solve iteratively the barycentric equation given by Eq. (7.7) [Pennec 06a]. In the case of bi-invariant metrics, this yields the following algorithm:

1) `Initialize for example` $m_0 := x_1$.

2) `Update the estimate of the mean by:` $m_{t+1} := m_t.\exp\left(\sum_{i=1}^N w_i \log({m_t}^{-1}.x_i)\right)$.

3) `Test convergence. If not reached, go to step 2.`

It was proposed in [Woods 03] to compute empirically bi-invariant means of invertible matrices with the same algorithm, even though no bi-invariant Riemannian metrics exist for such transformations. This works well in practice, but no precise definition of bi-invariant means was given in [Woods 03]. Furthermore, no proof of convergence of the iterative strategy was given, and neither the existence nor uniqueness of bi-invariant means were proved.

Here, one of our contributions is to provide a *general and precise definition* of bi-invariant means, which is *valid for any finite-dimensional real Lie group*, via Eq. (7.8). This will allow us to show the existence and uniqueness of the bi-invariant mean provided the dispersion of the data is small enough.

Interestingly, the mapping $\Phi : m \mapsto m.\exp\left(\sum_{i=1}^N w_i \log(m^{-1}.x_i)\right)$ plays a central role our approach. Let us now detail some of its properties.

Proposition 7.4. *Let (w_i) be N fixed non-negative weights. Then mapping $\Psi : \mathfrak{g}^{N+1} \to \mathfrak{g}$ defined by $\Psi(v_1, ..., v_N, z) = \log\left(\exp(z).\exp\left(\sum_{i=1}^N w_i \log(\exp(-z).\exp(v_i))\right)\right)$ is analytic near* 0*.*

Proof. The multivariate nature of Ψ complicates the proof a little bit, but this comes from the simple fact that Ψ is a composition of other analytic mappings: namely the mapping H defined in Subsection 2.3.6, the mapping $v \mapsto -v$ and the weighted sum $(v_1, ..., v_N) \mapsto \sum_i w_i.v_i$. This suffices to ensure that near 0, Ψ is the sum an absolutely converging infinite multivariate series whose variables are the $v_1, ..., v_N$ and z. For more details on multivariate analytic functions in Lie algebras, see [Godement 82], Chapter VI. □

Actually, Ψ is also an analytic function of the non-negative normalized *weights* (w_i). Since these weights live in a compact set, we can guarantee the existence and uniqueness of the bi-invariant mean *independently of the weights considered* (provided the dispersion of the data is small enough). But for more simplicity and clarity, we will skip these details in this thesis and consider fixed weights (w_i) in the proofs.

In the following, $\|.\|$ will be any norm on $\mathfrak{g}$ such that for all x, y in $\mathfrak{g}$, we have: $\|[x, y]\| \leq \|x\|.\|y\|$.

Corollary 7.2. *Let us suppose that the x_i and m are sufficiently close to e. Then we have the following development:*

$$\log(\Phi(m)) = \bar{l} + O((\sum_{i=1}^{N} \|\log(x_i)\| + \|\log(m)\|)^2), \tag{7.9}$$

where $\bar{l} = \sum_i w_i \log(x_i)$.

Proof. Successive applications of the BCH formula (see Chapter 2, Section 2.3.6 for a description of this powerful tool) yield the first term $(\bar{l})$ of the infinite series of Ψ, which is intuitively the usual arithmetic mean obtained when all the data and m commute. The bound obtained on the deviation with respect to $\bar{l}$ is a direct consequence of the fact that Ψ is analytic: the order of any remaining term of the infinite series is equal or larger to two and as a consequence the other terms can be bounded by a $O((\sum_{i=1}^{N} \|\log(x_i)\| + \|\log(m)\|)^2)$. □

Corollary 7.2 has the following consequence:

Corollary 7.3. *For all α in $]0, 1[$, there exists a $R > 0$ such that whenever $\|\log(x_i)\| \leq \alpha.R$ and $\|\log(m)\| \leq R$ then we also have $\|\log(\Phi(m))\| \leq R$.*

Proof. Just notice in Eq. (7.9) that the norm of the first order term is less or equal than $\alpha.R$ and that for the second-order term, which is a $O((\sum_{i=1}^{N} \|\log(x_i)\| + \|\log(m)\|)^2)$, there exists a constant C such that the second-order term is bounded in the following way:

$$\|O((\sum_{i=1}^{N} \|\log(x_i)\| + \|\log(m)\|)^2)\| \leq C.(N.\alpha^N + 1).R^2.$$

Since R^2 is a $o(R)$, $C.(N.\alpha^N + 1).R^2 \leq (1 - \alpha).R$ provided that R is sufficiently small. From this we obtain $\|\log(\Phi(m))\| \leq \alpha.R + (1 - \alpha).R = R$, which concludes the proof. □

Corollary 7.3 shows that provided the x_i and m are close enough to e, we can iterate indefinitely Φ over the successive estimates of the 'mean' of the x_i. This shows that the iterative scheme presented before is *stable* and remains indefinitely well-defined when the data is close enough to e (that is, without taking numerical errors into account). For the moment, we have only considered the case where all elements are close to e. We will see in the next subsection how this extends to the general case, where all the data are only assumed to be close to one another, possibly very far from e.

7.4.3 Convergence: Special Case

The Bi-Invariant Mean as a Fixed Point. Let α be in $]0,1[$. Accordingly with Corollary 7.3, let us take the same $R > 0$ such that for all i, $\|\log(x_i)\| \leq \alpha.R$ and $\|\log(m)\| \leq R$. Then, we know from Corollary 7.3 that $\|\log(\Phi(m))\| \leq R$.

Now, let us define $\Omega = \{m \in \mathcal{G} : \ \|\log(m)\| \leq R\}$. From Corollary 7.3, we know that Φ defines a mapping from Ω to Ω. Now, let us note that $\tilde{m} \in \Omega$ is a solution of Eq. (7.8) if and only if $\tilde{m}$ is a *fixed point* of Φ, i.e. $\Phi(\tilde{m}) = \tilde{m}$. To show the existence of a solution of Eq. (7.8), we can therefore use a *fixed point Theorem.*

The mathematical literature abounds with fixed point theorems. Let us first consider Brouwer's fixed point theorem:

Theorem 7.5. *Brouwer's Fixed Point Theorem [Samelson 63]. Let $\Psi : B^n \to B^n$ be a continuous mapping, where B^n is the n-dimensional Euclidean closed ball, i.e. $B^n = \{x \in \mathbb{R}^n : \sum_i (x_i)^2 \leq 1\}$. Then Ψ has at least one fixed point.*

Corollary 7.4. *With the assumptions made at the beginning of this subsection, then Eq. (7.8) has at least one solution in Ω.*

Proof. In our case, this result applies, since we can define $\Psi : \log(\Omega) \to \log(\Omega)$ by $\Psi(v) = \log(\Phi(\exp(v)))$. Since $\log(\Omega)$ is precisely a closed ball, and thus homeomorphic to the Euclidean closed ball, then Brouwer's theorem applies and guarantees the existence of at least one fixed point of Ψ, which is also a fixed point of Φ and therefore a solution of Eq. (7.8). □

The existence of a solution to Eq. (7.8) is thus guaranteed. However, in order to prove the convergence of the iterative strategy to a fixed point of Φ, the mathematical tool we need is another type of fixed point theorem. We will now recall Picard's fixed point Theorem, which is the following:

Theorem 7.6. *Picard's Fixed Point Theorem. Let (E, d) be a complete metric space and $f : E \to E$ be a $K-$contraction, i.e. for all x, y of E, $d(f(x), f(y)) \leq K.d(x, y)$, with $0 < K < 1$. Then f has a unique fixed point p in E and for all sequence $(x_n)_n > 0$ verifying $x_{n+1} = f(x_n)$, then $x_n \to p$ when $n \to +\infty$, with at least a $K-$linear speed of convergence.*

Proof. This is classical undergraduate topology. The usual proof consists of taking any sequence satisfying $x_{n+1} = f(x_n)$, and of showing that it is a Cauchy sequence and thus has a limit in E, which is the unique fixed point of E. Moreover, one can show that $d(x_{n+1}, p) \leq K.d(x_n, p)$, which prooves that the speed of convergence is at least $K-$linear. □

Here, (Ω, d) is the complete metric space in which the successive evaluations of the 'mean' live. The distance d is simply given by $d(m, n) = \|\log(m) - \log(n)\|$. To obtain the existence, uniqueness of a solution of Eq. (7.8) and linear convergence of our iterative scheme to this point, it only remains to show that Φ is a contraction. This leads to the following Proposition:

Proposition 7.5. *When the number R in Corollary 7.3 is chosen small enough, Φ is a contraction.*

Proof. Let us consider $E = \log(\Omega)$ with $\Theta : E \to E$ defined as in the proof of Corollary 7.4 by $\Theta(v) = \log(\Phi(\exp(v)))$. The key idea is to see that Θ is smooth with respect to $\log(m)$ *and* the $(\log(x_i))$, with the property that the norm of the differential of Θ is uniformly bounded in the following way:

$$\|D_{\log(m)}\Theta\| \leq O(\|\log(m)\| + \sum_i \|\log(x_i)\|). \tag{7.10}$$

In fact, Eq. (7.10) is a simple consequence of the fact that Ψ is analytic: $D_{\log(m)}\Theta$ is simply one of its partial derivative, which is therefore also analytic. Its value at 0 is precisely 0, and therefore all the terms of its infinite series are of order one or larger, which yields the bound in $O(\|\log(m)\| + \sum_i \|\log(x_i)\|)$.

With the bound given by Eq. (7.10), we can ensure that when R is small enough, there exists β in $]0,1[$ such that $\|D_{\log(m)}\Theta\| \leq \beta$ for all m. Then we have the classical bound:

$$\|\Theta(v) - \Theta(w)\| \leq (\sup_{z\in E}\|D_z\Theta\|)\|v - w\| \leq \beta\|v - w\|.$$

Since $\beta < 1$, Θ is by definition a contraction, and so is Φ.

□

Corollary 7.5. *As a consequence, when the data (x_i) are given close enough to e, there exists a open neighborhood of e in which there exists a unique solution to Eq. (7.8). Moreover, the iterative strategy given above always converges towards this solution, provided that the initialization to this algorithm is chosen sufficiently close to the data (so that Corrolary 7.2 can apply). Moreover, the speed of convergence is at least linear.*

Proof. Just apply Picard's Theorem to Φ and recall that being a fixed point of Φ is equivalent to being a solution of Eq. (7.8). □

7.4.4 Convergence: General Case

In the previous subsection, we have shown the existence and uniqueness of a solution of the exponential barycentric equation living in a open neighborhood of e, as long as the data were all close enough to e. In fact, this can be greatly generalized, as shown by the following result:

Theorem 7.7. *Let g be in $\mathcal{G}$. Then there exists a groupwise geodesically convex open neighborhood $\mathcal{V}$ of g, such that whenever the data (x_i) are in $\mathcal{V}$, then there exists a unique solution of Eq. (7.8) in an open neighborhood of g. Moreover, the classical iterative strategy always converges towards this solution, provided the initialization is taken close enough to g; also, the speed of convergence is at least linear.*

Proof. Just multiply the data by g^{-1} on the left to shift all the points in an adequate open neighborhood of the neutral element. Then one can run the iterative scheme to obtain the unique solution of the barycentric equation which lives close to e. Then just multiply on the left by g this solution to find the unique solution of the barycentric equation with the real data. Then note that the normal (non-shifted) iterative scheme is just the shifted version of the scheme associated to the shifted data. □

This leads to the following definition:

Definition 7.2. *Let the $(x_i)_{i=1}^N$ be some data belonging to a small enough groupwise geodesically convex set of $\mathcal{G}$. Then, for any system of (normalized) non-negative weights $(w_i)_{i=1}^N$, we call* **bi-invariant mean** *of (x_i) with respect to the weights (w_i) the unique solution (in a neighborhood of the data) of the group barycentric equation (7.8).*

Proposition 7.6. *The bi-invariant mean is left-, right- and inverse-invariant.*

Proof. The data is by hypothesis close enough to one another so that we can apply Lemma 7.1, so that we have:

$$\begin{aligned} Ad(m).\left(\sum_{i=1}^N w_i \log(m^{-1}.x_i)\right) = \;& \sum_i w_i \log\left(m.(m^{-1}.x_i).m^{-1}\right) \\ & \sum_{i=1}^N w_i \log(x_i.m^{-1}). \end{aligned}$$

Since $Ad(m)$ is invertible, the usual barycentric equation, which is left-invariant, is equivalent to a right-invariant barycentric equation, which shows that the barycenter is both left- and right-invariant. Now, to prove the invariance with respect by inversion, note that:

$$\begin{aligned}(-1) \times \left(\sum_{i=1}^{N} w_i \log(m^{-1}.x_i)\right) = & \sum_i w_i \log\left((m^{-1}.x_i)^{-1}\right) \\ & \sum_{i=1}^{N} w_i \log(x_i^{-1}.(m^{-1})^{-1}),\end{aligned}$$

which shows that whenever m is the bi-invariant mean of the x_i, m^{-1} is that of the ${x_i}^{-1}$, which is exactly inverse-invariance. $\square$

Some Comments on Bi-Invariant Means. We have rigorously generalized to any real Lie group the notion of bi-invariant mean normally associated to bi-invariant Riemannian metrics, even in the case where such metrics fail to exist. This novel mean enjoys all the desirable invariance properties, and can be iteratively computed in a very efficient way.

One should note that as usual with means in manifolds, the bi-invariant mean only exists provided the data are close enough to one another: the dispersion should not be too large. In the next section, we will see more precisely in various situations what practical limitation is imposed on the dispersion of the data. One does not seem to lose much in this regard with respect to existing Riemannian bi-invariant means: we will show for example that the bi-invariant mean of rigid transformations exists if and only if the bi-invariant mean of their rotation parts exists.

In the general case, it would be very interesting to have more precise criteria for the existence and uniqueness of the bi-invariant mean, such as the ones that exist in the Riemannian case and which are based on the concept of *regular geodesic balls* (see [Pennec 06a], pages 8 and 9). Generalizing these powerful Riemannian criteria to the bi-invariant case will be the subject of future work. The key difficulty for this generalization is that our bi-invariant mean is *not* defined via the *minimization* of a metric dispersion, but only via a barycentric equation. Our bi-invariant mean is thus somehow more like a *critical point* than a *minimizer*, which is weaker and more difficult to handle theoretically.

7.4.5 Higher Order Moments

In this Section, we have presented a complete and rigorous framework for defining and computing the bi-invariant mean in any finite-dimensional Lie group. Based on this remarkable choice of mean, how can one compute more sophisticated statistics, and in particular higher order moments?

Linearization of the Data around the Mean. Let us suppose that we have N samples (x_i) of a random variable X living in a Lie group $\mathcal{G}$. Let us also assume that their bi-invariant mean m is well-defined. Then, around m, the initial speeds of group geodesics provide a local linearization of $\mathcal{G}$ *compatible* with the notion of bi-invariant mean. By this we mean that the arithmetic mean of the vectors associated to the data (x_i) is precisely 0, which generalizes the property of the Fréchet mean associated to a Riemannian metric given by Eq. (7.7). In the following, let us write $v(x)$ for this linearization.

To see why this linearization has this property, let us recall that the unique group geodesic $\gamma_{m,x}(t)$ starting from m and joining a (close enough) point x is given by $\gamma_{m,x}(t) = m.\exp(t.\log((m^{-1}.x))$. As a consequence, for any datum x_j, $v(x_j)$ is equal to $v(x_j) = D_e L_m.\log(m^{-1}.x_j)$. From the barycentric equation (7.8), we obtain

$$\sum_i v(x_i) = \sum_i D_e L_m . \log(m^{-1}.x_i) = D_e L_m . \left(\sum_i \log(m^{-1}.x_i) \right) = 0,$$

which is the announced result.

Computing Higher Order Moments. With the linearization $x \mapsto v(x)$ around $\mathbb{E}(x_i)$, one can represent the data (x_i) by vectors belonging to the tangent space of $\mathcal{G}$ at $\mathbb{E}(x_i)$. Then, exactly as in the Riemannian case (see [Pennec 06a] Section 5 for a definition of the Riemannian covariance), one can for example compute from these vectors the empirical *covariance matrix* $\widehat{Cov}(X)$ of the random variable X with respect to *any* Riemannian metric defined on $\mathcal{G}$, with

$$\widehat{Cov}(X) \stackrel{def}{=} \frac{1}{N-1} \sum_i v(x_i).v(x_i)^T,$$

where the transposition operator $.^T$ is the one associated to the chosen Riemannian metric at $\mathbb{E}(x_i)$. Higher order empirical moments can be computed in the same way. In fact, using a Riemannian metric for computing moments of order 2 or higher is unavoidable, since this requires computing scalar products of vectors, which is precisely what Riemannian metrics are all about. In the case where no bi-invariant Riemannian metric exists, one can for example rely on left-invariant, or right-invariant Riemannian metrics to perform this task.

7.5 Bi-Invariant Means in Simple Cases

Let us now detail several insightful cases where the algebraic mean can be *explicitly* or directly computed, without using the classical iterative scheme.

7.5.1 Bi-Invariant Mean of Two Points

There is a closed form for the bi-invariant mean of two points:

Proposition 7.7. *Let x be in G and y be in a GGC open neighborhood of x. Then their bi-invariant mean m with respect to the couple of weights $(1-\alpha, \alpha)$ is given by:*

$$m = x. \exp\left(\alpha \log(x^{-1}.y)\right) = x.(x^{-1}.y)^{\alpha}. \tag{7.11}$$

Proof. We can simply check that m is a solution to the adequate barycentric equation. We have:

$$\log(m^{-1}.x) = \log(\exp(-\alpha \log(x^{-1}.y)).x^{-1}.x) = -\alpha \log(x^{-1}.y).$$

Also, we have that:

$$\log(m^{-1}.y) = \log(\exp(-\alpha \log(x^{-1}.y)).x^{-1}.y) = \log(n^{-\alpha}.n),$$

with $n = x^{-1}.y$. Therefore:

$$\begin{cases} \alpha. \log(m^{-1}.y) = \log(n^{\alpha \times (1-\alpha)}) \\ (1-\alpha). \log(m^{-1}.x) = \log(n^{-\alpha \times (1-\alpha)}) = -\alpha. \log(m^{-1}.y). \end{cases}$$

Thus, m is the bi-invariant mean of x and y. □

Notice that the explicit formula given by Eq. (7.11) is quite exceptional. In general, there will be *no* closed form for the bi-invariant mean, as soon as $N > 2$. However, there are some specific groups where a closed form exists for the bi-invariant mean in *all cases*, and we will now detail some examples of this rare phenomenon.

7.5.2 Scalings and Translations in 1D

Here, we will devote some time to a very instructive group: *the group of scalings and translations in 1D*. The study of this (quite) simple group is relevant in the context of this work, because it is one of the most simple cases of non-compact *and* non-commutative Lie groups which does not possess any bi-invariant Riemannian metric. This group has many of the properties of rigid or affine transformations, but with only two degrees of freedom, which simplifies greatly the computations, and allows a direct (2D) geometric visualization in the plane. For these reasons, this is a highly pedagogical case. In the rest of this Subsection, we will let this group be written $ST(1)$.

Elementary Algebraic Properties of $ST(1)$.

- An element g of $ST(1)$ can be uniquely represented by a couple (λ, t) in $\mathbb{R}_+^\star \times \mathbb{R}$. λ corresponds to the scaling factor and t to the translation part.
- The action of $ST(1)$ on scalars is given by: $(\lambda, t).x = \lambda.x + t$ for every scalar x.
- The multiplication in $ST(1)$ is: $(\lambda', t').(\lambda, t) = (\lambda'.\lambda, \lambda'.t + t')$. $ST(1)$ is thus a *semi-direct product* between the multiplicative group $(\mathbb{R}_+^\star, \times)$ and the additive group $(\mathbb{R}, +)$. Both groups are commutative, but this semi-direct product is *not*.
- Inversion: $(\lambda, t)^{-1} = (\frac{1}{\lambda}, -\frac{t}{\lambda})$.
- $ST(1)$ can be faithfully represented by the subgroup of triangular matrices of the form:
$$\begin{pmatrix} \lambda & t \\ 0 & 1 \end{pmatrix}.$$
- The elements of the Lie algebra of $ST(1)$ are of the form $(d\lambda, dt)$, where $d\lambda$ and dt are any scalars.
- The group exponential $\exp(d\lambda, dt)$ has the following form:
$$\exp(d\lambda, dt) = \begin{cases} (e^{d\lambda}, \frac{dt}{d\lambda}(e^{d\lambda} - 1), \text{ when } d\lambda \neq 0, \\ (1, dt), \text{ when } d\lambda = 0, \end{cases}$$
where e^λ is the scalar exponential of λ. Thus, we see that the group exponential is simply given by the scalar exponential on the scaling part, whereas the translation part mixes the multiplicative and additive influences of both components. Moreover, we see geometrically than in the upper half plane $\mathbb{R}_+ \times \mathbb{R}$, the curve given by $\exp(s.(d\lambda, dt))$ with s varying in $\mathbb{R}$ is on a *straight line*, whose equation is $t = \frac{dt}{d\lambda}\lambda - 1$.
- $ST(1)$ is entirely groupwise geodesically convex: any two points can be joined by a unique group geodesic. In particular, the group logarithm is always well-defined and given by:
$$\log(\lambda, t) = \begin{cases} (\ln(\lambda), t.\frac{\ln(\lambda}{1-\lambda}), \text{ when } \lambda \neq 1, \\ (0, t), \text{ when } \lambda = 1, \end{cases}$$
where $\ln(\lambda)$ is the natural (scalar) logarithm of λ. Same remark as for the exponential: we get the classical logarithm on the scaling part and a mixture of the multiplicative and additive logarithms on the translation part. We recall that in the case of an additive

group such $(\mathbb{R},+)$, both additive exponential and logarithm are simply the identity. This is what we get both for the exponential and the logarithm when there is no scaling.

The unique group geodesic joining (λ, t) and (λ', t') of the form $(\lambda, t).\exp(s.(d\lambda, dt))$ with s in $[0,1]$ has its parameters $(d\lambda, dt)$ given by:

$$(d\lambda, dt) = \left(\ln(\frac{\lambda'}{\lambda}), \left(\frac{t'-t}{\lambda}\right) . \left(\frac{\ln(\frac{\lambda'}{\lambda})}{\frac{\lambda'}{\lambda}-1}\right) \right). \tag{7.12}$$

Absence of Bi-Invariant Metrics. $ST(1)$ is one of the most simple non-compact and non-commutative Lie groups. In terms of bi-invariant metrics, it exhibits the typical tendency of such Lie groups: it has no such metric. As usual (see Section 7.2), to see this, we use the fact it is necessary and sufficient that the adjoint representation of $ST(1)$ be not bounded to ensure that no bi-invariant metric exists for this group. To show this, we use again the classical matrix representation of $ST(1)$:

$$\begin{aligned} Ad((\lambda,t)).(d\lambda,dt) &\sim \begin{pmatrix} \lambda & t \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} d\lambda & dt \\ 0 & 0 \end{pmatrix} . \begin{pmatrix} \frac{1}{\lambda} & -\frac{t}{\lambda} \\ 0 & 1 \end{pmatrix} \\ &\sim \begin{pmatrix} d\lambda & -t.d\lambda + \lambda.dt \\ 0 & 0 \end{pmatrix} \\ &= (d\lambda, -t.d\lambda + \lambda.dt). \end{aligned}$$

Both factors 't' and 'λ' in $-t.d\lambda + \lambda.dt$ are not bounded and thus $Ad(ST(1))$ cannot be bounded. As a consequence, $ST(1)$ has no bi-invariant metric. Both $(\mathbb{R}_+{}^\star, \times)$ and $(\mathbb{R},+)$ are commutative and thus have bi-invariant metrics, but interestingly, their semi-direct product has no such metric.

A Closed Form for the Bi-Invariant Mean. We recall that the bi-invariant mean in a Lie group is defined implicitly by a barycentric equation, given by Eq. (7.8). Here, since we have explicit formulae for the group exponential and logarithm, one can use these formulae to try to solve directly the barycentric equation. This leads to the following result:

Proposition 7.8. *Let $((\lambda_i, t_i))$ be N points in $ST(1)$ and (w_i) be N non-negative (normalized) weights. Then the associated bi-invariant mean $(\bar{\lambda}, \bar{t})$ is given explicitly by:*

$$\begin{cases} \bar{\lambda} = e^{\sum_i w_i \ln(\lambda_i)}, \textit{ (weighted geometric mean of scalings)}, \\ \bar{t} = \frac{1}{Z}.\sum_i w_i.\alpha_i.t_i, \textit{ (weighted arithmetic mean of translations influenced by scalings)}, \end{cases} \tag{7.13}$$

with:

$$\begin{cases} \alpha_i = \frac{\ln(\frac{\lambda_i}{\bar{\lambda}})}{\frac{\lambda_i}{\bar{\lambda}}-1}; \textit{ note that } \alpha_i = 1 \textit{ when } \lambda_i = \bar{\lambda}. \\ Z = \sum_i w_i.\alpha_i \end{cases} .$$

Proof. Just replace in the barycentric equation the exponentials and logarithms by the formulae given above. Since the scaling component is independent from the translation one, we simply obtain the geometric mean, which is the bi-invariant mean for positive numbers. The translation part can be handled simply using directly Eq. (7.12), which yields this simplified expression for the barycentric equation:

$$\sum_i w_i \left(\frac{t_i - \bar{t}}{\bar{\lambda}}\right) . \left(\frac{\ln(\frac{\lambda_i}{\bar{\lambda}})}{\frac{\lambda_i}{\bar{\lambda}}-1}\right) = 0.$$

Hence the result. □

Comparison Between Group and Metric Geodesics. In Figure 7.3, one can visually compare the group geodesics to some of their left-invariant and right-invariant (metric) counterparts.

Interestingly, one of the left-invariant metrics on $ST(1)$ induces an isometry between this group and Poincaré half-plane model for hyperbolic geometry (see [Gallot 93], page 82-83 for more details on this space). The scalar product of this scalar metric is the most simple at the $(1,0)$: it is the usual Euclidean scalar product. Geodesics take a very particular form in this case: they are the set of all the *half-circles perpendicular to the axis of translations* and of all (truncated below the axis of translations) lines perpendicular to the axis of translations (these lines can be seen as half-circles of infinite diameter anyway).

Thanks to Proposition 7.1, we know that the right-invariant Riemannian metric whose scalar product at $(1,0)$ is the same as the previous metric can be obtained simply by 'inverting' this left-invariant metric. As a consequence, its geodesics can be computed simply by inverting the initial conditions, computing the associated left-invariant geodesic and finally inverting it. The right-invariant geodesics visualized in Fig. 7.3 are by consequence some sort of 'inverted half-circles'. In fact, simple algebraic computations show that these geodesics are all *half-hyperbolas*.

One should note that the simple form taken by left-invariant geodesics is indeed exceptional. In general, there are no closed form for neither left- nor right-invariant geodesics, and group geodesics are simpler to compute, since in most practical cases they only involve the computation of a matrix exponential and a matrix logarithm, for which very efficient methods exist [Higham 05, Cheng 01]. Another nice Lie group where left-invariant metrics (and by consequence also right-invariant metrics) take a simple (closed) form is the group of rigid transformations. See [Boisvert 06] for examples of left-invariant statistics on rigid transformations in the context of a statistical study of human scoliotic spines.

Extension to $ST(n)$. One can directly generalize the results obtained for $ST(1)$ to the more general group $ST(n)$ of scalings and translations in nD. Instead of being a scalar, the translation is in this general case a $n-$dimensional vector. This does not change anything: all the algebraic properties of $ST(1)$ are also valid for $ST(n)$. In particular, one can use Eq. (7.13) to compute bi-invariant means in $ST(n)$.

7.5.3 The Heisenberg Group

With the group $ST(1)$, we had seen a simple case of mixing between a 1D multiplicative group and a 1D additive group. In this subsection, we study instead a 3D group where this time 2 additive groups (one 2D and the other 1D) are mixed.

The Heisenberg group. It is the group of 3D upper triangular matrices M of the form:

$$M = \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}.$$

To simplify notations, we will also write (x, y, z) to represent an element of this group.

Elementary Algebraic Properties. They are the following:

- Multiplication: $(x_1, y_1, z_1).(x_2, y_2, z_2) = (x_1 + x_2, y_1 + y_2, z_1 + z_2 + x_1.y_2)$. The first two parameters thus live in a 2D additive group which is independent of the third

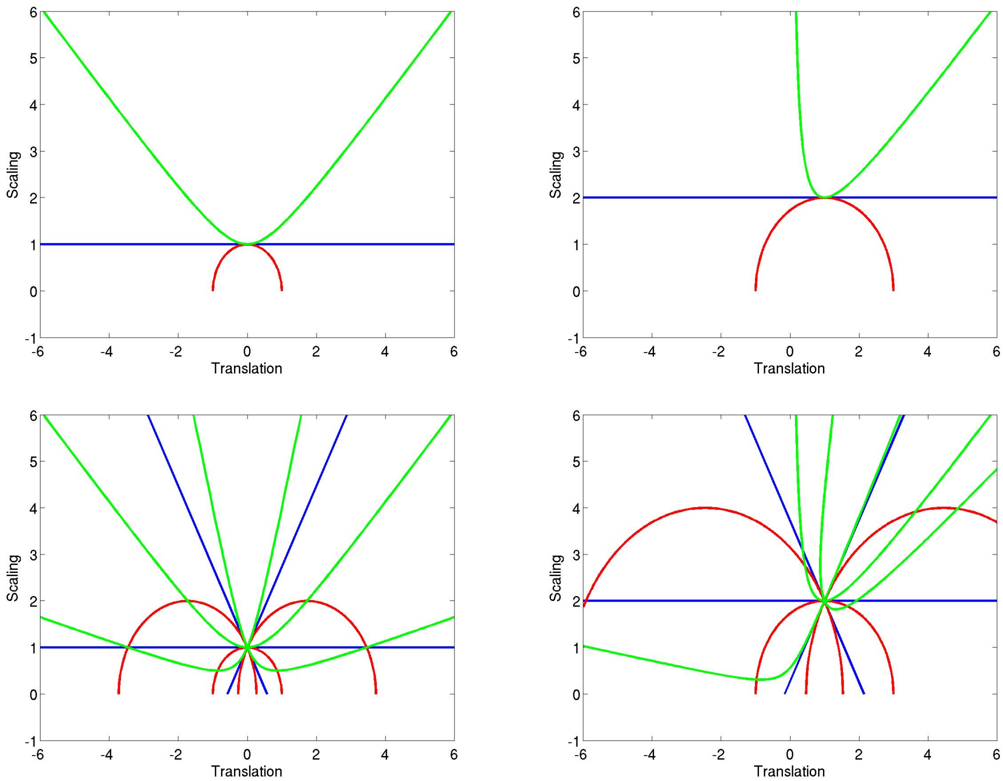

Figure 7.3: **Examples of geodesics in the group of scalings and translations in 1D. Top row**: two examples of left- and right- and group geodesics. **Bottom row**: two examples of geodesics with each time three possible orientations. **Blue**: group geodesics, **red**: left-invariant geodesics and **green**: right-invariant geodesics. Note the particular form taken by group geodesics, which are part of straight lines and of the left-invariant geodesics, which are half-circles perpendicular to the horizontal axis. Right-invariant geodesics are also given in a closed form and are in fact half-hyperbolas.

parameter, whereas the third additive parameter is influenced by the first two. The Heisenberg group is thus a *semi-direct product* between $(\mathbb{R}^2, +)$ and $(\mathbb{R}, +)$, which is *not* commutative.

- Inversion: $(x, y, z)^{-1} = (-x, -y, -z + x.y)$. Neutral element: $(0, 0, 0)$.
- As in the $ST(1)$ case, the Heisenberg group is entirely groupwise geodesically convex and we have:
$$\begin{cases} \exp((dx, dy, dz)) = (dx, dy, dz + \frac{1}{2}.dx.dy) \\ \log((x, y, z)) = (x, y, z - \frac{1}{2}x.y). \end{cases}$$
The unique group geodesic joining (x_m, y_m, z_m) and (x, y, z) of the form $(x_m, y_m, z_m).\exp(s.(dx, dy, z))$ with s in $[0, 1]$. Its parameters (dx, dy, dz) are given by:

$$(dx, dy, dz) = \left(x - x_m, y - y_m, z - z_m + \frac{1}{2}.(x_m.y_m - x.y + x_m.y - x.y_m)\right). \quad (7.14)$$

Bi-Invariant Metrics and Bi-invariant Means. As in the $ST(1)$ case, no bi-invariant metrics exists and one has the closed form for the bi-invariant mean. Interestingly, the bi-invariant mean yields a simple arithmetic averaging of the first two parameters. The third parameter is also averaged arithmetically, except that this arithmetic mean is 'corrected' by a quadratic function of the first two parameters of the data.

Proposition 7.9. *The action of the adjoint representation Ad of the Heisenberg group at a point (x, y, z) on an infinitesimal displacement (dx, dy, dz) is given by:*

$$Ad(x, y, z).(dx, dy, dz) = (dx, dy, -y\,dx + x\,dy + dz).$$

As a consequence, no bi-invariance metric exists for the Heisenberg group.

Proof. Proceed exactly as in Proposition 7.2. □

Proposition 7.10. *Let $((x_i, y_i, z_i))$ be N points in the Heisenberg group and (w_i) be N non-negative (normalized) weights. Then the associated bi-invariant mean $(\bar{x}, \bar{y}, \bar{z})$ is given explicitly by:*

$$(\bar{x}, \bar{y}, \bar{z}) = \left(\sum_i w_i x_i, \sum_i w_i y_i, \sum_i w_i z_i + \frac{1}{2}\left(\bar{x}.\bar{y} - \sum_i w_i x_i.y_i \right) \right).$$

Proof. Just replace in the barycentric equation the exponentials and logarithms by the formulae given above. Since the first two components are additive and independent from the third one, their bi-invariant mean is simply their arithmetic mean. The third coefficient case can be handled simply using directly Eq. (7.14), which yields this simplified expression for the barycentric equation:

$$\sum_i w_i \left(z_m - z_i + \frac{1}{2}.(x_m.y_m - x_i.y_i + x_m.y_i - x_i.y_m) \right).$$

Hence the result. □

7.5.4 On a Subgroup of Triangular Matrices

We can generalize the results obtained on the Heisenberg group to the following subgroup of triangular matrices:

Definition 7.3. *Let $UT(n)$ be the group of $n \times n$ upper triangular matrices M of the form:*

$$M = \lambda.Id + N,$$

where λ is any positive scalar, Id the identity matrix and N an upper triangular nilpotent matrix ($N^n = 0$) with only zeros in its diagonal.

The Heisenberg group is the subgroup of matrices of $UT(3)$ whose λ is always equal to 1. The situation in this case is particularly nice, since thanks to the fact that N is nilpotent, one can perform *exactly* all the usual algebraic operations in $UT(n)$:

- Group exponential:

$$\begin{aligned} \exp(dM) &= \exp(d\lambda.Id + dN) = \exp(d\lambda.Id).\exp(dN) \\ &= e^{d\lambda}.\textstyle\sum_{k=0}^{n-1} \frac{dN^k}{k!}. \end{aligned}$$

- Group logarithm:

$$\begin{aligned} \log(M) &= \log(\lambda.Id + N) = \log\left((\lambda.Id).(Id + \tfrac{1}{\lambda}.N)\right) \\ &= \ln(\lambda).Id + \textstyle\sum_{k=1}^{n-1} \frac{(-1)^{k+1}}{k} (\frac{N}{\lambda})^k. \end{aligned}$$

- Inversion:

$$\begin{aligned}(M)^{-1} &= (\lambda.Id+N)^{-1} = \lambda^{-1}.(Id+\tfrac{N}{\lambda})^{-1}\\ &= \lambda^{-1}.\textstyle\sum_{k=0}^{n-1}(-1)^k.(\tfrac{N}{\lambda})^k.\end{aligned}$$

- Multiplication:

$$M'.M = (\lambda'.Id+N').(\lambda.Id+N) = (\lambda'.\lambda).Id + (\lambda'.N+\lambda.N'+N'.N).$$

Using these closed forms, one can derive the following equation:

$$\log(M'.M) = \ln(\lambda'.\lambda).Id + \sum_{k=1}^{n-1}\frac{(-1)^{k+1}}{k}.\left(\frac{1}{\lambda}.N + \frac{1}{\lambda'}.N' + \frac{1}{\lambda'.\lambda}.N.N'\right)^k,$$

which in turns allows us to compute the equation satisfied by the bi-invariant mean $\bar{M} = \bar{\lambda}.Id + \bar{N}$ in $UT(n)$:

$$\begin{gathered}-\textstyle\sum_i w_i \log(\bar{M}^{-1}.M_i) = \sum_i w_i \log(\bar{M}.{M_i}^{-1}) = 0\\ \Longleftrightarrow\\ \textstyle\sum_i w_i\left(\ln(\bar{\lambda}.{\lambda_i}^{-1}).Id + \sum_{k=1}^{n-1}\frac{(-1)^{k+1}}{k}.\left(\frac{1}{{\lambda_i}^{-1}}.{N_i}^{-1} + \frac{1}{\lambda}.\bar{N} + \frac{1}{\bar{\lambda}.{\lambda_i}^{-1}}.{N_i}^{-1}.\bar{N}\right)^k\right) = 0,\end{gathered} \tag{7.15}$$

where ${N_i}^{-1}$ is the nilpotent part of ${M_i}^{-1}$. From Eq. (7.15), we see that $\bar{\lambda}$ is simply the geometric mean of the λ_i, and that the coefficient of $\bar{N}$ can be recursively computed, starting from coefficients above the diagonal. The key idea is that the k^{th} power of a nilpotent matrix N will have non-zero coefficients only in its k^{th} upper diagonal.

As a consequence, to compute the coefficients of $\bar{M}$ above the diagonal, one only needs to take into account the the following terms: $\frac{1}{{\lambda_i}^{-1}}.{N_i}^{-1} + \frac{1}{\lambda}.\bar{N}$. These coefficients will simply be a weighted arithmetic mean of the coefficients in the data, the weights being equal to $(w_i.\frac{\lambda_i}{\lambda})/S$ with $S = \sum_j w_j.\frac{\lambda_j}{\lambda}$. Using this result, then one can compute the coefficients above, which are an weighted arithmetic mean of the corresponding coefficients in the data, with a quadratic correction involving the previous coefficients. The same phenomenon appears for the next set of coefficients above, with an even more complex correction involving all the previously computed coefficients. One can continue this way until all the coefficients of the mean have been effectively computed.

7.6 Linear Transformations

7.6.1 General Rigid Transformations

We recall that the Lie group of rigid transformations in the n-dimensional Euclidean space, written here $SE(n)$, is the semi-direct product between $(SO(n), \times)$ (rotations) and $(\mathbb{R}^n, +)$ (translations) defined has follows:

- An element of $SE(n)$ is uniquely represented by a couple $(R, t) \in SO(n) \times \mathbb{R}^n$ and its action on a point x of $\mathbb{R}^n$ is given by $(R, t).x = R.x + t$.
- Multiplication: $(R', t').(R, t) = (R'.R, R'.t + t')$.
- Neutral element: $(Id, 0)$, inverse: $(R^T, -R^T.t)$.

- Representation of (R, t) by a $(n+1) \times (n+1)$ matrix using homogenous coordinates:

$$\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}.$$

- Lie algebra: thanks to the matrix representation of $SE(n)$, it is simple to see that the Lie algebra of $SE(n)$ can be faithfully represented by the following vector space of matrices:

$$\begin{pmatrix} dR & dt \\ 0 & 0 \end{pmatrix},$$

where dR is any skew $n \times n$ matrix and dt any vector of $\mathbb{R}^n$. In this representation, the Lie bracket $[.,.]$ is simply given by the matrix Lie bracket: $[A, B] = A.B - B.A$.

- Group exponential: it can be computed using directly the matrix representation, or by identifying the one-parameter subgroups of $SE(n)$. This yields:

$$\exp(dR, dt) = \left(e^{dR}, e^{dR}.\left(\int_0^1 e^{-u.dR} du\right).dt\right),$$

where e^{dR} is the matrix exponential of dR.

Existence of the Logarithm. From Section 7.3, we know that it is only defined locally in a neighborhood of the neutral element $(Id, 0)$. However, since we have a faithful representation of $SE(n)$ in terms of matrices, we can use the matrix criterion for the existence of the principal logarithm: from Subsection 2.3.4, we know that an invertible matrix with no (complex) eigenvalue on the closed half-line of negative real numbers has unique matrix logarithm with eigenvalues having imaginary parts in $]-\pi, \pi[$. In the case of rotations, this means that the various angles of rotation (there can be *several* angles of rotation in the general n-dimensional case, whereas only one exists in 2D or 3D) of a rotation R should not go outside $]-\pi, \pi[$ if we want the logarithm of R to be well-defined. Otherwise, one cannot define a unique logarithm. This is typically the case for $-Id$ in 2D (i.e. a rotation of 180 degrees), whose two 'smallest' real logarithms are the following:

$$\begin{pmatrix} 0 & -\pi \\ \pi & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & \pi \\ -\pi & 0 \end{pmatrix}.$$

Going back to $SE(n)$, we have the following result:

Proposition 7.11. *The logarithm of a rigid transformation (R, t) is well-defined if and only if the logarithm of its rotation part R is well-defined.*

Proof. The logarithm of (R, t) is well-defined if and only if the matrix representing (R, t) has a principal logarithm, which is equivalent to the fact that it has no eigenvalue on the closed negative line. Then, this is equivalent to the fact that R has no eigenvalue on the closed negative line, since the eigenvalues of the upper triangular matrix (in terms of blocks) $\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$ depend only on the blocks in its diagonal, i.e. only on R, and *not* t. As a consequence, the logarithm of a rigid transformation is well-defined if and only if the logarithm of its rotation part is well-defined. □

Criterion for the Existence of the Bi-Invariant Mean. We have seen in Section 7.2.3 that no bi-invariant metric exists in the rigid case. One may now ask the question: is there a simple criterion for the existence of the bi-invariant mean of rigid transformations? When bi-invariant metrics exist, one has ideed such a criterion, as mentioned before: the bi-invariant mean exists and is unique as long as all the data is strictly included in a *regular geodesic ball* of radius r such that the geodesic ball of radius $2.r$ is still regular ([Pennec 06a], page 9).

In the case of 3D rotations, this means that the data is included in a regular geodesic ball of radius strictly inferior to $\pi/2$. This is equivalent to saying that there exists a point R such that for all the rotations R_i of the data, the rotation $R^{-1}.R_i$ has an angle of rotation strictly smaller than $\pi/2$ [Pennec 98a]. This implies that for all R_i and R_j in the data, the angle of rotation of ${R_j}^{-1}.R_i$ is smaller than $\pi - C$ where C is a positive constant.

In n dimensions, the situations is more complicated, since an arbitrary rotation has *several* angles of rotations. We have the following result: *any rotation can be decomposed into a sum of independent 2D rotations* [Lang 04]. For more details on this remarkable spectral decomposition, see Lemma 7.2 below. If the data are included in a *regular geodesic ball* of radius r such that the geodesic ball of radius $2.r$ is still regular, then for any couple of data R_i and R_j, all of the angles of rotations of ${R_j}^{-1}.R_i$ are bounded by $\pi - C$. This simply comes from the fact that the principal logarithm of ${R_j}^{-1}.R_i$ is well-defined (which imposes that the angles of rotations are all stricly smaller than π) and from compactness of the regular geodesic ball in which the data are included.

Remarkably, one can guarantee the existence and uniqueness of the bi-invariant mean of rigid transformations directly from a Riemannian criterion on their rotation parts:

Theorem 7.8. *Let (R_i, t_i) be N rigid transformations such that the bi-invariant Fréchet mean of their rotation parts is well-defined (i.e., satisfies the criterion given above). Then for any set of non-negative weights (w_i), there exists a unique bi-invariant mean for (R_i, t_i).*

Proof. Let use write $\bar{R}$ for the bi-invariant Fréchet mean of the rotations parts of the data. The bi-invariant mean of the data is necessarily of the form $(\bar{R}, t)$, since in an open neighborhood of the rotation parts, $\bar{R}$ is the only solution of the rotation part of the bi-invariant barycentric equation, which does not depend on translations.

$\bar{R}$ is included in the same geodesic ball as the rotations and therefore, for any R_i of the data, the angles of rotations of $\bar{R}^{-1}.R_i$ are all smaller than or equal to $\pi - C$, where C is a positive constant smaller than π.

Let us now check whether there exists a unique translation $\bar{t}$, which satisfies the barycentric equation of bi-invariant means, which writes here:

$$\sum_i w_i \log((\bar{R}, \bar{t}).(R_i, t_i)^{-1}) = 0. \tag{7.16}$$

From Proposition 7.11, we know that the logarithm of $(\bar{R}, t).(R_i, t_i)^{-1})$ is well-defined for *any value of* t, since the logarithm of $\bar{R}.{R_i}^T$ is well-defined for all i. Now, does there exist a unique value of t (by definition $\bar{t}$) satisfying Eq. (7.16)?

We have: $(\bar{R}, t).(R_i, t_i)^{-1} = (\bar{R}.R_i^T, \bar{R}.(-R_i^T.t_i)+t)$. Let us write $M(dR) = e^{dR}. \int_0^1 e^{-u.dR} du$. In terms of translations, Eq. (7.16) writes:

$$\begin{array}{c} \sum_i w_i M\left(\log(\bar{R}.R_i^T)\right)^{-1}.\left(\bar{R}.(-R_i^T.t_i) + \bar{t}\right) = 0. \\ \Longleftrightarrow \\ \left(\sum_i w_i M\left(\log(\bar{R}.R_i^T)\right)^{-1}\right).\bar{t} = \sum_i w_i M\left(\log(\bar{R}.R_i^T)\right)^{-1}.R_i^T.t_i. \end{array} \tag{7.17}$$

Thus, we see that the existence and uniqueness of $\bar{t}$ is equivalent to the invertibility of the following matrix: $\sum_i w_i \; M\left(\log(\bar{R}.R_i^T)\right)^{-1}$. Under the assumptions described above on rotations, this matrix is indeed invertible by Lemma 7.2 (see below), and this concludes the proof.

□

Lemma 7.2. *Let (dR_i) be N a skew symmetric matrices, such that the norm of their largest (complex) eigenvalue is always smaller than $\pi - C$, with $C > 0$.*

Let $M(dR)$ be equal to $e^{dR}.\int_0^1 e^{-u.dR}du$ for any skew symmetric matrix. Then for all dR_i, $M(dR_i)$ is invertible, and for any non-negative weights (w_i), $\sum_i w_i M(dR_i)^{-1}$ is also invertible.

Proof. The key idea is to see the form taken by $M(dR)$ in an appropriate orthonormal basis. From classical linear algebra, we know that any skew symmetric matrix dR has a remarkable spectral decomposition. More precisely, there exists a specific decomposition of the geometrical space $\mathbb{R}^n$ in a direct sum of *mutually orthogonal* subspaces, which are all *stable* for dR [Lang 04]. These subspaces are of two kinds:

- k (possibly equal to zero) 2-dimensional vector subspaces E_k.
- A single subspace F of dimension $n - 2.k$ (the orthogonal complement of the other subspaces), which is the kernel of dR.

For any E_j, there exists an orthonormal basis of E_j such that dR restricted to E_j is in this basis of the following matrix form:

$$\begin{pmatrix} 0 & -\theta_j \\ \theta_j & 0 \end{pmatrix}.$$

θ_j ($\neq 0$) is the j^{th} angle of rotation of the $n-$dimensional rotation e^{dR}. This spectral decomposition allows for the explicit computation of $M(dR)$ is the various subspaces mentionned above. First, in the kernel F of dR, $M(dR)$ is simply the identity. In an E_j, we have:

$$\exp(dR)|_{E_j} \sim \begin{pmatrix} cos(\theta_j) & -sin(\theta_j) \\ sin(\theta_j) & cos(\theta_j) \end{pmatrix}.$$

A few extra manipulations yield:

$$M(dR)|_{E_j} = \left(\exp(dR).\int_0^1 \exp(-u.dR)du\right)|_{E_j} \sim \begin{pmatrix} \frac{\sin(\theta_j)}{\theta_j} & \frac{\cos(\theta_j)-1}{\theta_j} \\ -\frac{\cos(\theta_j)-1}{\theta_j} & \frac{\sin(\theta_j)}{\theta_j} \end{pmatrix}.$$

The decomposition detailed above shows that whenever for all j, $|\theta_j| < 2.\pi$, $M(dR)$ is always invertible (which is more than we need), since the determinant of the latter matrix is equal to $\left(\frac{\sin(\theta_j)}{\theta_j}\right)^2 + \left(\frac{\cos(\theta_j)-1}{\theta_j}\right)^2$, which is positive for $|\theta_j| < 2.\pi$. Furthermore, a direct computation shows that the inverse of $M(dR)$ takes the following form in E_j:

$$M(dR)^{-1}|_{E_j} \sim \begin{pmatrix} \frac{\theta_j.\sin(\theta_j)}{2.(1-\cos(\theta_j)} & \frac{\theta_j}{2} \\ -\frac{\theta_j}{2} & \frac{\theta_j.\sin(\theta_j)}{2.(1-\cos(\theta_j)} \end{pmatrix}.$$

For $|\theta_j| < \pi - C$, some elementary calculus shows that there exists a constant $K > 0$, such that $\frac{\theta_j.\sin(\theta_j)}{2.(1-\cos(\theta_j)} > K$. As a consequence, we have:

$$M(dR)^{-1}|_{E_j} \sim \begin{pmatrix} a & b \\ -b & a \end{pmatrix},$$

with $a > K > 0$. Under the assumption that for all j, $|\theta_j| < \pi - C$, this has the interesting consequence that $M(dR)^{-1}$ has the following decomposition:

$$M(dR)^{-1} = S + A,$$

where S is a symmetric positive-definite matrices with all its eigenvalues larger than K and where A is a skew symmetric matrix. Then let us take N skew matrices whose eigenvalues are smaller than $\pi - C$. Any convex combination of the $M(dR_i)^{-1}$ writes:

$$\sum_i w_i M(dR_i)^{-1} = \left(\sum_i w_i S_i\right) + \left(\sum_i w_i A_i\right) = \tilde{S} + \tilde{A},$$

where $\tilde{S}$ is still symmetric definite positive and $\tilde{A}$ is skew symmetric. $\sum_i w_i M(dR_i)^{-1}$ is therefore invertible, since *any* matrix of the form $\tilde{S} + \tilde{A}$ is invertible. To see this, just remark that if there exists one x such that $(\tilde{S} + \tilde{A}).x = 0$, this implies $x^T.\tilde{S}.x + x^T.\tilde{A}.x = 0$. Then notice that $x^T.\tilde{A}.x = (x^T.\tilde{A}.x)^T = -x^T.\tilde{A}.x = 0$. Thus $(\tilde{S} + \tilde{A}).x = 0$ implies $x^T.\tilde{S}.x = 0$, which is equivalent ($\tilde{S}$ is symmetric positive-definite) to $x = 0$. Consequently $\tilde{S} + \tilde{A}$ is invertible and this ends the proof. □

7.6.2 2D Rigid Transformations

Contrary to the general case, 2D rigid transformations have a particularity: one has a *closed form* for the bi-invariant mean. The reason behing this is that $SO(2)$, the group of 2D rotations, is *commutative*. As a consequence, one can compute explicitely the bi-invariant mean of the rotation parts of the data and deduce from it the translation part using the barycentric equation, like in the proof of Theorem 7.8. More precisely, we have:

Proposition 7.12. *Let (R_i, t_i) be N 2D rigid transformations, such that the angles of rotation of the rotations $R_i.R_j^T$ are all strictly inferior to π (regular geodesic ball criterion in 2D). Then the bi-invariant mean $(\bar{R}, \bar{t})$ associated to the weights (w_i) is given explicitely by:*

$$\begin{cases} \bar{R} = R_1.\exp\left(+\sum_i w_i \log\left(R_1^T.R_i\right)\right) \\ \bar{t} = \sum_i w_i\, Z^{-1}.M\left(\log\left(\bar{R}.R_i^T\right)\right)^{-1}.R_i^T.t_i, \end{cases} \tag{7.18}$$

with the following formulae for M and Z:

$$M\left(\left(\begin{array}{cc} 0 & -\theta \\ \theta & 0 \end{array}\right)\right)^{-1} \stackrel{def}{=} \left(\begin{array}{cc} \frac{\theta.\sin(\theta)}{2.(1-\cos(\theta)} & \frac{\theta}{2} \\ -\frac{\theta}{2} & \frac{\theta.\sin(\theta)}{2.(1-\cos(\theta)} \end{array}\right), Z \stackrel{def}{=} \sum_i w_i M\left(\log(\bar{R}.R_i^T)\right)^{-1}.$$

Example of Bi-Invariant Mean. Let us take a look at the example chosen in [Pennec 06a], page 31. Let $f_1 = (\pi/4, -\sqrt{2}/2, \sqrt{2}/2)$, $f_2 = (0, \sqrt{2}, 0)$ and $f_3 = (-\pi/4, -\sqrt{2}/2, -\sqrt{2}/2)$ be three rigid transformations. The first coefficient corresponds to the angle of rotation (chosen here in $[-\pi, \pi[$) and last two to the translation.

We can compute exactly the bi-invariant mean of these rigid transformations with (7.18). Furthermore, we can easily calculate the Log-Euclidean mean of these transformations, given in a closed form by Eq. (6.17). A left-invariant Fréchet mean can also be computed explicitely in this case thanks to the simple form taken by the corresponding geodesics (see [Pennec 06a] for more details). And finally, thanks to Proposition 7.1, the analogous right-invariant Fréchet mean can be computed by inverting the data, computing their left-invariant mean and then inverting this Fréchet mean. This yields (after a number of simple but tedious algebraic manipulations):

- Left-invariant Fréchet mean: $(0, 0, 0)$,
- Log-Euclidean mean: $\left(0, \frac{\sqrt{2}-\frac{\pi}{4}}{3}, 0\right) \simeq (0, 0.2096, 0)$,
- Bi-invariant mean: $\left(0, \frac{\sqrt{2}-\frac{\pi}{4}}{1+\frac{\pi}{4}.(\sqrt{2}+1)}, 0\right) \simeq (0, 0.2171, 0)$,
- Right-invariant Fréchet mean: $\left(0, \frac{\sqrt{2}}{3}, 0\right) \simeq (0, 0.4714, 0)$.

Interestingly, we thus see that the mean rotation is exactly the same in all four cases. But the mean translations are different, and the bi-invariant mean is located nicely *between* the left- and right-invariant Fréchet means. This is quite intuitive, since the bi-invariant mean can be looked upon as an in-between alternative with regard to left- and right-invariant Fréchet means. Remarkably, the Log-Euclidean mean (which is much simpler to compute) is very close to the bi-invariant mean, which is a result somehow comparable to what are observed on diffusion tensors in Chapter 4 (in this context, the bi-invariant mean of tensors will be referred to as the *affine-invariant* mean; please refer to Section 7.6.4 to understand why).

7.6.3 General Linear Transformations

The Bi-Invariant Mean as a Geometric Mean. It is possible to show that in the linear group $GL(n)$, the determinant of the bi-invariant mean is equal to the *scalar geometric mean* of the determinant of the data. This mean can thus be looked upon as a *generalization to invertible linear transformations of the geometric mean of positive numbers.* This generalization is not the only possible one, since the Log-Euclidean mean, which we have described in Chapter 6, has the same property. However, the Log-Euclidean mean is restricted to linear transformations whose principal logarithm is well-defined, which is *not* the case for the bi-invariant mean.

Proposition 7.13. *Let T_i be N linear transformations in $GL(n)$ and let (w_i) be N (normalized) non-negative weights, such that their bi-invariant mean $\mathbb{E}(T_i)$ exists. Then, we have:*

$$\begin{cases} \text{if } det(S_i) > 0, \text{ for all } i, \text{ then } det(\mathbb{E}(T_i)) = \exp\left(\sum_i w_i.\ln(det(S_i))\right), \\ \text{if } det(S_i) < 0, \text{ for all } i, \text{ then } det(\mathbb{E}(T_i)) = -\exp\left(\sum_i w_i.\ln(-det(S_i))\right), \end{cases}$$

Proof. When the bi-invariant mean is well-defined, then all of the determinants of the data have the same sign, which is also the sign of $\mathbb{E}(T_i)$. Otherwise, one of the products $\mathbb{E}(T_i)^{-1}.T_i$ would have a negative determinant and its principal logarithm would fail to exist.

To prove our result, we will rely only on two ingredients: the barycentric equation (7.8) and the following property: $\det(M) = \exp(\mathrm{Trace}(\log(M)))$, which holds for any square matrix with a principal logarithm. This (classical) equality can be shown for example using the Jordan (or Schur) decomposition of the matrix M.

Taking the trace of the barycentric equation and then the (scalar) exponential, we get:

$$1 = \Pi_i \exp\left(w_i \mathrm{Trace}\left(\ln\left(\det\left(\mathbb{E}(T_i)^{-1}.T_i\right)\right)\right)\right).$$

Then, using $\det(A.B) = \det(A).\det(B)$ and $\ln(a.b) = \ln(a) + \ln(b)$ and $\det(\mathbb{E}(T_i)^{-1}.T_i) = |\det(\mathbb{E}(T_i))|^{-1}.|\det(S_i)|$, we get the geometrical interpolation of determinants:

$$1 = |\det\left(\mathbb{E}(T_i)\right)|^{-1}.\exp\left(\sum_i w_i.\ln\left(|\det(S_i)|\right)\right),$$

which yields the result. □

Practical Computation of the Bi-Invariant Mean. Whe have seen in Section 7.4 that an efficiently iterative scheme could be used to compute bi-invariant means. It relies on successive computations involving inversions, exponentials and logarithms. To actually compute numerically the exponential and logarithm, we recommend using modern and efficient algorithms like the 'Scaling and Squaring' method for the matrix exponential [Higham 05] and the 'Inverse Scaling and Squaring Method' [Cheng 01] for the matrix logarithm.

In the case of rigid and affine transformations, one can use their representation by matrices given by homogeneous coordinates, and use the general iterative scheme on these matrices to compute their bi-invariant mean.

7.6.4 Tensors

Let us now say a few words about the tensor case. As explained in Chapter 3, a number of teams proposed almost simultanously in 2004 to endow this space with *affine-invariant metrics*. Interestingly, the Fréchet mean associated to affine-invariant Riemannian metrics on the tensor space coincides with the bi-invariant mean of tensors, looked upon as elements of $GL(n)$. Indeed, the affine-invariant Fréchet mean $\mathbb{E}_{Aff}(S_1, ..., S_N)$ of N tensors $S_1, ..., S_N$ is defined implicitely by the following barycentric equation:

$$\sum_{i=1}^{N} w_i . \log(\mathbb{E}_{Aff}(S_1, ..., S_N)^{-\frac{1}{2}} . S_i . \mathbb{E}_{Aff}(S_1, ..., S_N)^{-\frac{1}{2}}) = 0, \tag{7.19}$$

which happens to be exactly equivalent our general equation (7.8) for bi-invariant means (just multiply (7.19) on the left by $\mathbb{E}_{Aff}(S_1, ..., S_N)^{-\frac{1}{2}}$ and on the right by $\mathbb{E}_{Aff}(S_1, ..., S_N)^{+\frac{1}{2}}$ to obtain (7.8)). Intuitively, this means that our bi-invariant mean naturally unifies into a very general framework a number of well-established notions of means for various types of data living in Lie groups (e.g, tensors, rotations, translations).

7.7 Left-Invariant Polyaffine Transformations

Why (Again) a Novel Polyaffine Framework? The properties of the Log-Euclidean polyaffine framework presented in Chapter 6 are excellent, and do not suffer from the defects of our original polyaffine framework. However, this Log-Euclidean framework is limited to rigid or affine transformations *whose principal logarithm is well-defined*, i.e. which are close enough to the identity. So far, we did not find this restriction limiting in our work on 3D locally affine registration, mainly because we perform first a global affine alignment of images before any locally affine registration. Still, it would be very interesting to have an infinitesimal strategy of fusion capable of handling any type of local rigid or affine deformations (provided their dispersion is not too large, of course), *regardless of their distance to the identity.*

Before presenting the last polyaffine frameworks of this thesis, let us briefly recall the two polyaffine frameworks we have already presented in Chapters 5 and 6.

7.7.1 Polyaffine Transformations

The idea is to define transformations that exhibit a locally rigid or affine behavior, with nice invertibility properties. Following the seminal work of [Little 96], we model such transformations by a finite number N of affine *components*. Precisely, each component i consists of an affine transformation T_i and of a non-negative *weight function* $w_i(x)$ which models its spatial extension: the influence of the i^{th} component at point x is proportional to $w_i(x)$. Furthermore, we assume that for all x, $\sum_{i=1}^{N} w_i(x) = 1$, i.e. the weights are normalized.

Fusion of Displacements. In order to obtain a global transformation from several weighted components, the classical approach to fuse the N components, given in [Sheppard 68], simply consists in averaging the associated displacements according to the weights:

$$T(x) = \sum_{i=1}^{N} w_i(x) T_i(x). \tag{7.20}$$

The transformation obtained using (7.20) is smooth, but this approach has one major drawback: although each component is invertible, the resulting global transformation is *not invertible* in general. To remedy this, we proposed in Chapter 5 to rely on the averaging of some *infinitesimal* displacements associated to each affine component instead. The resulting global transformation is obtained by integrating an Ordinary Differential Equation (ODE), which is computationally more expensive but guarantees its invertibility and also yields a simple form for its inverse.

Log-Euclidean Polyaffine Transformations. However, the first polyaffine framework we proposed lacks some important properties: the inverse of a polyaffine transformation is not polyaffine in general, and the polyaffine fusion of affine components is not invariant with respect to a change of coordinate system.

This is the reason why we proposed a novel framework in Chapter 6, called *Log-Euclidean polyaffine*, which overcomes these defects. We also showed that this novel type of geometrical deformations can be computed very efficiently (as well as their inverses) on regular grids, with a simple algorithm called the *Fast Polyaffine Transform*.

Let us now see what the Log-Euclidean polyaffine fusion consists of. Let (T_i) be N affine (or rigid) transformations, and let $(\log(T_i))$ be their logarithms. Using these logarithms, one can fuse the T_i infinitesimally according to the weights $w_i(x)$ with a *stationary* (or *autonomous*) ODE, called the 'Log-Euclidean polyaffine ODE'. In homogeneous coordinates, this ODE is the following:

$$\dot{x} = \sum_i w_i(x) \log(T_i).x, \tag{7.21}$$

which is a nice infinitesimal analogous of Eq. (7.20). The value at a point x of the Log-Euclidean polyaffine transformation (LEPT) defined by (7.21) is given by integrating (7.21) between time 0 and 1 with x as initial condition.

7.7.2 A Novel Type of Polyaffine Transformations

Left-Invariant Polyaffine Transformations (LIPTs). To define our novel polyaffine fusion, called *left-invariant*, we will rely on bi-invariant means of rigid or affine transformations.

Let (T_i) be N affine (or rigid) transformations, let $(w_i(x))$ be some non-negative weights functions. Let also (α_i) be N non-negative weights, which intuitively correspond to the *global* weights of components, whereas weight functions provide *local* information. Finally, let $\bar{T}$ be the weighted left-invariant mean of the T_i and the weights (α_i). The bi-invariant polyaffine transformation Φ associated to all of these data is defined as follows:

1. Starting from a position x_0, the following ODE (in homogeneous coordinates) is integrated during one unit of time, which yields a final position $\Psi(x_0)$ in homogeneous coordinates:

$$\dot{x} = \sum_i w_i(x) \log(\bar{T}^{-1}.T_i).x. \tag{7.22}$$

2. We obtain the value at x_0 of the LIPT Φ by computing: $\Phi(x_0) = \bar{T}.\Psi(x_0)$.

This allows to fuse infinitesimally the T_i provided only that the logarithms of the $\bar{T}^{-1}.T_i$ exist, which does not require that *any* of the logarithms of the T_i exist. The properties of this fusion are quite nice: left-invariance and affine-invariance. They are summarized in the following Proposition:

Proposition 7.14. *The left-invariant polyaffine fusion Φ of the components $(T_i, w_i(x))$ with respect to the global weights (α_i) has the following invariance properties:*

- *left-invariance: any left-multiplication (by an affine transformation) of the T_i results in a left-multiplication of Φ*
- *affine-invariance: the fusion does not depend on the current choice of coordinate system*
- *at any point x such that $w_i(x) = \alpha_i$ for all i, we have $\Phi(x) = \bar{T}.x$, i.e. x moves according to the mean transformation.*

Proof. Proof of left-invariance: let A be an affine transformations, and let us replace the T_i by $A.T_i$. By construction, $\bar{T}$ is bi-invariant and is replaced by $A.\bar{T}$ and thus Eq. (7.22) remains unchanged:

$$\dot{x} = \sum_i w_i(x) \log\left(\bar{T}^{-1}.A^{-1}.A.T_i\right).x = \sum_i w_i(x) \log\left(\bar{T}^{-1}.T_i\right).x.$$

Since the value of Φ is obtained by left-multiplying by $\bar{T}$ in a second step, Φ is replaced by $A.\Phi$, which means that our novel polyaffine fusion is left-invariant.

Affine-invariance: let us change the current coordinate system by transforming x into $y \stackrel{def}{=} A.x$ in homogeneous coordinates. This results in the following changes:

- a weight function $x \mapsto w_i(x)$ becomes $y \mapsto w_i\left(A^{-1}.y\right)$.
- an affine transformation T_i becomes $A.T_i.A^{-1}$.
- the bi-invariant mean becomes $A.\bar{T}.A^{-1}$.

In the new coordinate systems, the left-invariant polyaffine ODE becomes:

$$\dot{y} = \sum_i w_i\left(A^{-1}.y\right) \log\left(A.\bar{T}^{-1}.A^{-1}.A.T_i.A^{-1}\right).y = A.\left(\sum_i w_i\left(A^{-1}.y\right) \log\left(\bar{T}^{-1}.T_i\right)\right).A^{-1}.y,$$

which yields:

$$\frac{d}{dt}\left(A^{-1}.y\right) = \sum_i w_i\left(A^{-1}.y\right) \log\left(\bar{T}^{-1}.T_i\right).\left(A^{-1}.y\right). \tag{7.23}$$

This means that $x(t)$ is a solution of Eq. (7.22) if and only if $y(t) = A.x(t)$ is a solution of (7.23), i.e. of the left-invariant polyaffine ODE in the novel coordinate system. As a consequence, a change of coordinate system does not affect the left-invariant polyaffine fusion, i.e. this fusion is affine-invariant.

Finally, at a point x such that $w_i(x) = \alpha_i$, we have:

$$\sum_i w_i(x).\log(\bar{T}^{-1}.T_i).x = \left(\sum_i \alpha_i \log(\bar{T}^{-1}.T_i)\right).x = 0.x = 0,$$

by construction of the bi-invariant mean $\bar{T}$. This implies that X is a fixed point of Ψ. Therefore, we have $\Phi(x) = \bar{T}.x$ at this point. $\square$

Log-Euclidean vs. Left-Invariant Polyaffine Transformations. The price paid for the infinitesimal fusion of local rigid or affine components regardless of their distance to the identity is the following: the fusion is not *inversion-invariant*, i.e. the inverse of a left-invariant polyaffine transormation is not left-invariant polyaffine (but *right-invariant* polyaffine in fact). However, affine-invariance is preserved, i.e. independence with respect to the choice of coordinate system.

Computing Left-Invariant Polyaffine Transformations. The first step of the left-invariant polyaffine fusion, the ODE (7.22) is a simple Log-Euclidean polyaffine ODE, which can be integrated very efficiently using the Fast Polyaffine Transform described in [Arsigny 06a]. The second step is very simple to compute: it only consists in applying an affine transformation, which is the bi-invariant mean $\bar{T}$. We recall that this mean can be very efficiently numerically computed in homogeneous coordinates using the efficient iterative scheme described previously in this Chapter.

Right-Invariant Polyaffine Transformations (RIPTs). We have just defined LIPTs. What about right-invariant polyaffine transformations? With the same notations as before, one can indeed define such transformations, by slightly modifying the left-invariant polyaffine ODE:

1. Starting from a position x_0, the following ODE (in homogeneous coordinates) is integrated during one unit of time, which yields a final position $\Psi_R(x_0)$:

$$\dot{x} = \sum_i w_i \left(\bar{T}.x\right) \log\left(\bar{T}^{-1}.T_i\right).x. \tag{7.24}$$

2. We obtain the value at x_0 of the right-invariant polyaffine transformation Φ_R by computing: $\Phi_R(x_0) = \bar{T}.\Psi_R(x_0)$.

With exactly the same type of techniques as in the proof of Proposition 7.14, one can show that type of fusion is right-invariant and also affine-invariant. It is much less intuitive, since the weight functions w_i are geometrically deformed by $\bar{T}^{-1}$ before being used in the fusion. With have the interesting following relationship between LIPTs and RIPTs:

Proposition 7.15. *The inverse of a LIPT is the RIPT with inverted components, and vice versa.*

Proof. Inverting the left-invariant polyaffine fusion results in the following two steps:

1. multiplying by $\bar{T}^{-1}$

2. Integrating the following ODE during one unit of time:

$$\dot{x} = -\sum_i w_i(x) \log\left(\bar{T}^{-1}.T_i\right).x = \sum_i w_i(x) \log\left({T_i}^{-1}.\bar{T}\right).x \tag{7.25}$$

Let us use the change of variable $y = \bar{T}.x$ in (7.25). This yields:

$$\dot{x} = \bar{T}^{-1}.\dot{y} = \sum_i w_i \left(\bar{T}^{-1}.y\right) \log\left({T_i}^{-1}.\bar{T}\right).\bar{T}^{-1}.y = \bar{T}^{-1}\left(\sum_i w_i\left(\bar{T}^{-1}.y\right).\log\left(\bar{T}.T_i^{-1}\right)\right).y,$$

which yields the simpler equation:

$$\dot{y} = \sum_i w_i \left(\bar{T}^{-1}.y\right) \log\left(\bar{T}.T_i^{-1}\right).y, \tag{7.26}$$

which is the ODE associated to the first step of the right-invariant fusion of the inverses of the T_i with the same weights as originally.

Thus, $x(t)$ is the solution of (7.25), second step of the inversion of our LIPT if and only if $\bar{T}.x(t)$ is a solution of the first step of the RIPT with inverted components. As consequence, the two steps of the inversion of our LIPT are equivalent to the two steps the RIPT with inverted components. The inverse of a LIPT is therefore the RIPT with inverted components and *vice versa*.

□

And Bi-Invariant Polyaffine Transformations? So far, we have not been able to define a *bi-invariant* polyaffine fusion, i.e. an infinitesimal fusion of local affine transformations which would be simultanously left- and right- and inversion-invariant. Is such a fusion possible? We do not know yet, and this will be the subject of future work.

7.8 Conclusions and Perspectives

In this Chapter, we have presented a general framework to define rigorously a novel type of mean in Lie groups, called the *bi-invariant mean*. This mean enjoys many desirable *invariance properties*, which generalize to the non-linear case the properties of the arithmetic mean: it is invariant with respect to left- and right-multiplication, as well as inversion. Previously, this type of mean was only defined in Lie groups endowed with a bi-invariant Riemannian metric, like compact Lie groups such as the group of rotations [Pennec 06a, Moakher 02]. But Riemannian bi-invariant metrics do not always exist. In particular, we have proved in this work that such metrics do not exist in any dimension for *rigid transformations*, which form but the most simple Lie group involved in bio-medical image registration.

To overcome the lack of existence of bi-invariant Riemannian metrics for many Lie groups, we have proposed in this Chapter to define bi-invariant means in *any finite-dimensional real Lie group* via a general *barycentric equation*, whose solution is by definition the bi-invariant mean. We have shown the existence and uniqueness of this novel type of mean, provided the dispersion of the data is small enough, and the convergence of an efficient iterative algorithm for computing this mean has also been shown. The intuition of the existence of such a mean was first given in [Woods 03] (without any precise definition), along with an efficient algorithm for computing it (without proof of convergence), in the case of *matrix* groups. Moreover, we have briefly presented how empirical higher order moments can be computed in a simple way based on this novel notion of mean.

In the case of rigid transformations, we have been able to determine a simple criterion for the general existence and uniqueness of the bi-invariant mean, which happens to be the same as for rotations. We have also given closed forms for the bi-invariant mean in a number of simple but instructive cases, including 2D rigid transformations. Interestingly, for general linear transformations, we have shown that similarly to the Log-Euclidean mean, that we presented in Chapter 6, the bi-invariant mean is a generalization of the (scalar) geometric mean, since the determinant of the bi-invariant mean is exactly equal to the geometric mean of the determinants of the data.

Last but not least, we have used this new type of mean to define a novel class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components *arbitrarily far away from the identity*, contrary to Log-Euclidean polyaffine fusion, which we described in Chapter 6.

In future work, we are planning to compare the statistics obtained via the bi-invariant mean and the Log-Euclidean framework to other types of statistics on rigid or affine transformations such as the Log-Euclidean ones, which were presented in the previous Chapter, or left-invariant Riemannian statistics [Boisvert 06]. These statistics could prove very useful for example to constraint locally rigid or affine registration algorithms such as the one described in [Commowick 06a] and used in the previous Chapter. On this subject, one should note that both Log-Euclidean and bi-invariant frameworks can be used in practice in a simple way to compute statistics in *any* matrix groups (provided of course that the dispersion of the data is not too large, but this restriction also applies to Riemannian frameworks), whereas in the Riemannian left- and right-invariant cases, this depends very much on the form taken by geodesics in these cases. For rigid transformations, left-invariant geodesics are linked to the matrix exponential in a straightforward way and thus can be simply computed. However, to our knowledge, this is not the case for left-invariant Riemannian metrics on *affine transformations*, which makes the use of such metrics quite difficult in this case. No such problem occurs with our novel frameworks, which only rely on computations based on the *matrix* exponential and logarithm.

Chapter 8

Statistics on Diffeomorphisms in a Log-Euclidean Framework

We briefly present in this Chapter the generalization to diffeomorphisms of our Log-Euclidean framework, presented in Chapter 3 in the tensor case and generalized to linear transformations and finite-dimensional Lie groups in Chapter 6. Such a framework could prove very useful in computational anatomy, since it allows usual statistics to be performed on diffeomorphisms in a simple *vectorial* way on the logarithms of transformations, with excellent theoretical properties such as inversion-invariance. This yields in particular a closed form for the computation of mean diffeomorphisms, contrary to the iterative and possibly unstable computation suggested in [Vaillant 04]. Without relying on Riemannian geometry, such a framework provides *simple* processing algorithms *compatible* with a number of algebraic properties of diffeomorphisms.

As explained in this short Chapter, our non-linear generalization of the Log-Euclidean framework to general invertible deformations still poses some theoretical problems that are yet to be completely solved, due to the infinite-dimensional nature of diffeomorphisms. However, the encouraging partial theoretical evidence of the well-foundedness of our approach we present here, along with the promissing preliminary experimental results that we have already obtained have led us to present this work-in-progress 'as is' at the end of this thesis.

Contents

8.1 Introduction **168**
8.2 A Log-Euclidean Framework for Diffeomorphisms **169**
8.3 Computation of Exponentials and Logarithms **171**
8.4 Statistics on 3D Diffeomorphisms **173**
8.5 Conclusions and Perspectives **175**

Abstract. In this Chapter, we focus on the computation of statistics of invertible geometrical deformations (i.e., diffeomorphisms), based on the generalization to this type of data of the notion of *principal logarithm*. Remarkably, this logarithm is a simple 3D vector field, and can be used for diffeomorphisms close enough to the identity. This allows to perform *vectorial* statistics on diffeomorphisms, while preserving the invertibility constraint, contrary to Euclidean statistics on displacement fields. We also present here two efficient algorithms to

compute numerically logarithms of diffeomorphisms and exponentials of vector fields, whose accuracy is studied on synthetic data. Finally, we apply these tools to compute the mean of a set of diffeomorphisms, in the context of a registration experiment between an atlas an a database of 9 T1 MR images of the human brain.

Related Publications. This work was presented at the international conference MICCAI'2006 [Arsigny 06b], as well as at the International Workshop on the Mathematical Fundations of Computational Anatomy (MFCA-2006) [Arsigny 06d], which was one of the numerous MICCAI'2006 satellite workshops.

8.1 Introduction

In this Chapter, we focus on the computation of statistics of general *diffeomorphisms*, i.e. of geometrical deformations (non-linear in general) which are both one-to-one and regular (as well as their inverse). To quantitatively compare non-linear registration algorithms, or in order to constrain them, computing statistics on *global* deformations would be very useful as was done in [Commowick 05] with *local* statistics.

As we have seen in the Introduction of this thesis in Section 1.1.4, the computation of statistics is closely linked to the issue of the *parameterization* of diffeomorphisms. Statistics on *displacement fields* are not fully satisfactory, since the Euclidean means of such parameter do not always provide an average transformation which is invertible. In [Marsland 04], it was proposed to parameterize arbitrary diffeomorphisms with Geodesic Interpolating Spline control points [Camion 01], but although this guarantees the invertibility of the results, this low-dimensional parameterization may not be adequate for the whole variety of invertible transformations used in medical imaging.

To fully take into account the group structure of diffeomorphisms, it has been proposed to parameterize dense deformations with Hilbert spaces of *time-varying* velocity vector fields, which can be given an infinite-dimensional Lie group structure [Trouvé 98, Beg 05]. In [Vaillant 04], it is suggested that the linear space of intial momenta of the right-invariant geodesics in these spaces could provide an appropriate setting for statistics on diffeomorphisms. However, this is illustrated in [Vaillant 04] only in the *finite-dimensional* case of landmark matching. To our knowledge, this statistical framework has not been used yet in the *general case*, certainly because of the *iterative nature* of the computation of the mean in this setting, which requires very stable numerical algorithms to converge.

In this Chapter, we introduce a novel parameterization of diffeomorphisms, based on the generalization of the *principal logarithm* to non-linear geometrical deformations. Interestingly, this corresponds to parameterizing diffeomorphisms with *stationnary* velocity vectors fields. As for matrices, this logarithm can be used only for transformations close enough to the identity. However, our preliminary numerical experiments on 3D non-rigid registration suggest that this limitation affects only very large deformations, and may not be problematic for image registration results. This novel setting is the infinite-dimensional analogous of the Log-Euclidean framework proposed in Chapter 3 for tensors and in Chapter 6 for linear transformations. In this framework, usual Euclidean statistics can be performed on diffeomorphisms via their logarithms, whith excellent mathematical properties like inversion-invariance.

The rest of this Chapter is organized as follows. In the next Section, we present the Log-Euclidean framework for diffeomorphisms, which is closely linked to the notion of *one-parameter subgroups*. Then, we present two efficient algorithms to compute the exponential of a vector field and the logarithm of a diffeomorphism, which are exemplified on synthetic data.

Finally, we successfully apply our framework to non-linear registration results to compute a Log-Euclidean mean deformation between a 3D atlas and a dataset of 9 T_1 MR images of human brains.

8.2 A Log-Euclidean Framework for Diffeomorphisms

To generalize the notion of *principal logarithm* to diffeomorphisms, we will rely on its close link with *one-parameter subgroups*. Remarkably, one-parameter subgroups of diffeomorphisms take a simple form, and by analogy with the finite-dimensional case, we will define logarithms of diffeomorphisms as being the *infinitesimal generators* of these subgroups.

Exponential and One-Parameter Subgroups. Let us briefly recall the link between one-parameter subgroups and the group exponential detailed in Chapter 2. Let $(\mathcal{G}, .)$ be a group, which can typically be a matrix multiplicative group. Then a family of elements $(g(s))_{s\in\mathbb{R}}$ of $\mathcal{G}$ is a one-parameter subgroup of $\mathcal{G}$ if and only if $g(0)$ is the neutral element e of $\mathcal{G}$ and for all s, t: $g(s).g(t) = g(s+t)$. When such a subgroup belongs to a (finite-dimensional) Lie group and is continuous, it is also *differentiable*, and its derivative at 0, $\frac{dg}{ds}(0)$, is called its *infinitesimal generator*.

Examples of such subgroups are given by the exponential in the following way : $(\exp(s.M))_{s\in\mathbb{R}}$ is a one-parameter subgroup, where M belongs to the Lie algebra of $\mathcal{G}$ (i.e., is an element of the tangent space $T_e\mathcal{G}$). Conversely, we have the remarkable result that *all* continuous one-parameter subgroups are precisely of the form $(\exp(s.M))_{s\in\mathbb{R}}$. As a consequence, once one-parameter subgroups are identified in a Lie group, its exponential mapping is immediately known. This also yields the form taken by the logarithm, which is the (local) inverse of the exponential.

One-Parameter Subgroups of Diffeomorphisms. In the case of diffeomorphisms, what do continuous one-parameter subgroups look like? Quite intuitively, they are all obtained via the integration of *stationary* ODEs (also called *autonomous*), i.e ODE whose velocity vector does not depend on time [Tenenbaum 85].

Let $\dot{x} = V(x)$ be a stationary ODE, where the vector field $V(x)$ is smooth enough. The *flow* associated to this ODE is the family of mappings $\Phi_V(., s) : \mathbb{R}^n \to \mathbb{R}^n$ parameterized by a time parameter s, such that for a fixed x_0, $s \mapsto \Phi_V(x_0, s)$ is the unique solution of $\dot{x} = V(x)$ with initial condition x_0 at time 0.

Intuitively, for a fixed s, the flow $x \mapsto \Phi_V(x, s)$ gives the way the ambient space is deformed by the integration of the ODE during s units of time. Remarkably, the flow is a *one-parameter subgroup of diffeomorphisms*. This implies in particular that the deformations of space given at time 1 by $\Phi_V(., 1)$ are twice that observed at time 0.5 via $\Phi_V(., 0.5)$. The *infinitesimal generator* of this subgroup is $V(x)$. Conversely, all continuous one-parameter subgroups of diffeomorphisms are flows of some stationary ODE. See [Tenenbaum 85] for more details.

Exponentials of Vector Fields and Logarithms of Diffeomorphisms. Here, we propose to define the *exponential* $\exp(V)$ of a (smooth) vector field $V(x)$ as the flow at time 1 of the stationary ODE $\dot{x} = V(x)$. This is the only possible definition generalizing to vector fields the equivalence between one-parameter subgroups and exponential that exists in the finite-dimensional case. Intuitively, the *logarithm* $\log(\Phi)$ of a diffeomorphism Φ "close enough" to the identity is the unique vector field "near" zero such that $\exp(V) = \Phi$.

In the finite-dimensional case, the existence and uniqueness of such a logarithm can be proved by showing that the exponential is continuously differentiable and that its differential mapping at zero is the identity. This implies that the exponential is *diffeomorphic* near zero, i.e. is a smooth one-to-one mapping between an open neighborhood of zero and an open neighborhood of the identity. In this context, “near” zero and “close to” the identity mean belonging to one of these open neighborhoods.

In the infinite-dimensional case of diffeomorphisms, let V be a regular vector field, let t be a non-zero scalar, and let $\Phi_{t.V}(x, 1)$ be the flow at time 1 associated to $t.V(x)$. Then, a simple change of variable ($s \mapsto t.s$) shows that:

$$\operatorname*{limit}_{t \mapsto 0} \frac{\Phi_{t.V}(x,1) - x}{t} = \operatorname*{limit}_{t \mapsto 0} \frac{\exp(t.V)(x) - \exp(0)(x)}{t} = V(x),$$

which means intuitively that $D_0 \exp .V = V$, which suggests that the differential of the exponential is the identity, and that we have existence and uniqueness of the logarithm locally around the identity. To be entirely rigorous and to complete the proof, it is necessary to define precisely the smoothness required of V, i.e. to define properly the space of diffeomorphisms we are considering.

Can one just consider the very general spaces of $\mathcal{C}^r$ or $\mathcal{C}^\infty$ diffeomorphisms of $\mathbb{R}^n$? Well, the answer is negative in these cases [Milnor 83]: the group exponential “fails to be one-to-one or surjective near the identity” ([Banyaga 97], page 8). How come the properties of the exponential are so different in our infinite-dimensional case than in the finite-dimensional one? Essentially, the reason is that these spaces are *too large*.

In [Trouvé 95], Trouvé proposed a novel family of *smaller* infinite-dimensional groups of diffeomorphisms, partly motivated by practical reasons: “at the very end [it is important] to have an appropriate numerical scheme to solve various pattern recognition problems in this framework [i.e. with diffeomorphisms]”. This family of groups has remarkable properties and can be effectively used in practive to deal with diffeomorphisms. In recent years, this group has been increasingly relied on in the medical imaging community and can be now used to process medical images in a variety of contexts [Trouvé 98, Trouvé 00, Camion 01, Beg 03, Miller 03, Guo 04, Joshi 00, Beg 05, Trouvé 05b, Glaunes 04, Allassonnière 05].

Are these groups *locally exponential* [Glöckner 06], i.e. is their exponential locally diffeomorphic around zero? We do not have a definitive proof yet, but the following result suggests this is the case: “many (but not all) infinite-dimensional Lie groups $\mathcal{G}$ are locally exponential” ([Glöckner 06], page 5) and this is the case in particular for infinite-dimensional Lie groups whose tangent spaces are *Banach spaces* (the so-called *Banach-Lie groups*), see [Glöckner 06], page 3, or [Milnor 83]. This shows for that infinite-dimensional groups which are not exceedingly large, the usual properties of the exponential in finite dimensions still hold.

The groups of diffeomorphisms proposed by Trouvé are not exactly a Banach-Lie groups (otherwise we would be able to conclude once and for all). But they are very close to be so: Banach spaces “play the role of tangent spaces” in these groups. The technical difficulty here comes from the fact that no proper differential structure has been identified yet for these groups, only a “weak differentiable structure”. Furthermore, these groups are not exactly Lie groups, since they have no *Lie bracket*. But they are not far from being ones, since their group exponentials are well-defined and play a central role in the definition of their weak differentiable structures [Trouvé 95].

Alain Trouvé’s groups of diffeomorphisms have therefore a good chance of providing a fully rigorous infinite-dimensional setting in which our framework for diffeomorphisms is well-founded (locally around the identity). Even if for some rather technical reason it turned out otherwise, this could simply mean that we have not identified yet the adequate spaces of diffeomorphisms in which our Log-Euclidean framework can be used in completely rigorous way. Finding the answers to these technical questions will be the object of future work.

Log-Euclidean Statistics on Diffeomorphisms. On diffeomorphisms whose logarithm is well-defined, one can perform Euclidean operations, since these logarithms are simple vector fields. In particular, one can define a distance between these diffeomorphisms via a norm $\|.\|$ on vector fields: $\mathrm{dist}(\Phi_1, \Phi_2) = \|\log(\Phi_1) - \log(\Phi_2)\|$. Remarkably, this type of distance is *inversion-invariant*, since $\log(\Phi^{-1}) = -\log(\Phi)$. In the case of Hilbert norms, the point that minimizing the weighted sum of squared distances to the data is the *Log-Euclidean mean*, given by $\exp(\sum_i w_i \log(\Phi_i))$. This mean is inversion-invariant, and is also invariant with respect to the taking of square roots, since $\log(\Phi^{\frac{1}{2}}) = \frac{1}{2}\log(\Phi)$. More generally, in this setting, one can perform any kind of statistics on *diffeomorphisms* via *vectorial* statistics on their logarithms, which allows a straightforward generalization of classical analysis tools like Principal Component Analysis (PCA) on diffeomorphisms.

8.3 Computation of Exponentials and Logarithms

Fast Computation of Exponentials. Here, we present an efficient algorithm to compute exponentials of velocity vectors on regular grids, which generalizes to any initial vector field the Fast Polyaffine Transform we detail in Section 6.3 for Log-Euclidean polyaffine transformations. Exactly, as in Chapter 6, our approach is based on a non-linear generalization of the popular 'Scaling and Squaring' method, widely used to compute *matrix exponentials*. The basic idea is that the matrix exponential is much simpler to compute for matrices *close to zero*, for example using Padé approximants. In particular, one can compute very accurately $\exp\left(\frac{M}{2^N}\right)$ and obtain $\exp(M) = \exp(\frac{M}{2^N})^{2^N}$ by squaring recursively N times the result [Higham 05].

In the non-linear case, one can follow exactly the same steps as for matrices and generalize the 'Scaling and Squaring' to vector fields in the following way:

1. **Scaling step**: divide $V(x)$ by a factor 2^N, so that $V(x)/2^N$ is close enough to zero (according to the level of accuracy desired).

2. **Exponentiation step**: $\Phi_V\left(\frac{1}{2^N}, .\right)$ can be computed with a first-order explicit numerical scheme, i.e.: $\Phi_V\left(\frac{1}{2^N}, x\right) = x + \frac{V(x)}{2^N}$. Alternatively, more accurate numerical schemes such as Runge-Kutta's can be used.

3. **Squaring step**: N recursive squarings of $\Phi_V\left(\frac{1}{2^N}, .\right)$ yield an accurate estimation of $\Phi_V(1, .)$ (only N *compositions of mappings* are used).

Intuitively, this means that we obtain the deformations at time 1 as a result of 2^N times the composition of the (very) small deformations observed at time $\frac{1}{2^N}$. This allows a fast integration of stationary ODEs on regular grids, compared to classical integrations based on fixed time-steps. To perform the composition of (sampled) deformations on regular grids, we use bi- or tri-linear interpolation, which guarantees the continuity of the interpolation in an simple way.

Fast Computation of Logarithms. Exactly as for exponentials, we use here a non-linear generalization of the popular 'Inverse Scaling and Squaring' method (ISS), widely used to compute *matrix logarithms*. Anew, the idea is that logarithms are much simpler to compute for matrices *close to the identity*, for instance with Padé approximants. To transform a matrix

M so that it is closer to the identity, the ISS algorithm performs recursive computations of *square roots*. Then the identity $\log(M) = 2^N.\log(M^{2^{-N}})$ is used to compute $\log(M)$.

Let Φ be a diffeomorphism. To compute its logarithm V, we use the following non-linear generalization of the ISS algorithm:

1. **Scaling step**: chose a scaling factor 2^N (according to the level of accuracy desired).
2. **Square rooting step**: $\Phi^{2^{-N}}$ is computed by N successive recursive takings of square roots (see below).
3. **Computation of logarithm step**: $\log(\Phi)$ is given by $2^N.\log(\Phi^{2^{-N}})$, where $\log(\Phi^{2^{-N}})$ is simply estimated by $\Phi^{2^{-N}} - Id$ (where Id is the identity).

In order to compute square roots, we perform here a gradient descent on the functional $E_{\text{SQRT}}(T) = \frac{1}{2}.\int_\Omega \|T \circ T - \Phi\|^2(x)dx$, with $\frac{1}{2}.(\Phi - Id)$ as initialization. The (L^2) gradient of this energy is the following:

$$\nabla E_{\text{SQRT}}(T) = (DT^t) \circ T.(T \circ T - \Phi) + |\det(D(T^{-1}))|(T - \Phi \circ T^{-1}),$$

where DT is the Jacobian of T and where '.' and 'M^t' are matrix multiplication and transposition. Exactly as in the matrix case, this requires the inversion of T, which we also perform by gradient descent on $E_{\text{INV}}(T) = \frac{1}{2}.\int_\Omega \|\Phi \circ T - Id\|^2(x)dx$, with an initialization of $-(\Phi - Id)$. The (L^2) gradient is given by:

$$\nabla E_{\text{INV}}(T) = (D\Phi^t) \circ T.(\Phi \circ T - Id).$$

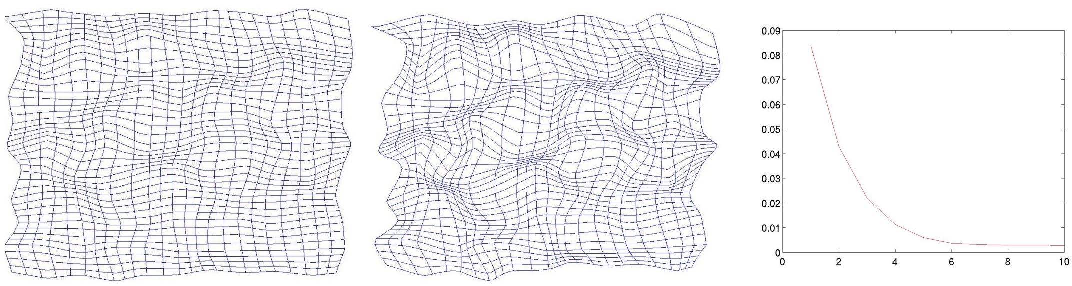

Figure 8.1: **Fast computation of the exponential of a random vector field. From left to right:** the two last iterations (scaling of 2^8) of our fast computation of the exponential and the evolution of the average relative accuracy with N. Note how deformations increase exponentially from one iteration to another. The relative accuracy obtained on average converges toward of 0.3%, which is approximately obtained for $N = 7$.

Experiments on 2D Synthetic Data. In these experiments, we evaluate the accuracy of our algorithms on 2D random deformations, sampled on a 40×40 regular grid. To generate a random vector field, random displacements are computed on a 11×11 regular grid (Gaussian white noise of standard deviation 0.2, coordinates of grid points in $[-5, 5]$) , which are smoothly interpolated on the finer grid using bilinear interpolation.

We measure the accuracy of the fast computation of the exponential by computing the relative difference with respect to a very accurate estimation of the continuous transformation, obtained by a classical integration (i.e., with fixed time step, here 2^{-8}) of the stationary

ODE associated to the random vector field, for each voxel of the grid. Fig. 8.1 presents the results, which show that a typical accuracy on average of 0.3% percent can be obtained with 7 squarings.

To evaluate the accuracy of the fast computation of the logarithm, we first estimate the exponential of the random vector field with our fast algorithm (with 8 squarings). Then, the logarithm of this diffeomorphism is computed and is compared to the original vector field (10 iterations are used for each gradient descent). Fig. 8.2 presents the results, which show that an accuracy of 3% percent can be obtained on average with 6 recursive computations of square roots.

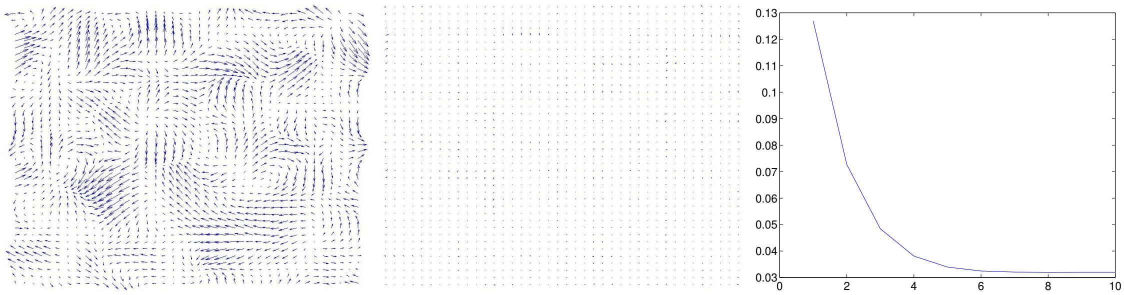

Figure 8.2: **Fast computation of the logarithm of a random deformation. From left to right:** random vector field (logarithm), then difference between original and estimated logarithms (for $N = 8$), evolution of accuracy with N. The relative accuracy obtained on average converges toward of 3%.

8.4 Statistics on 3D Diffeomorphisms

In this Section, we compute Log-Euclidean statistics on the registration results obtained between a $256\times256\times60$ artificial T_1 MR image of a human brain (coming from the BrainWeb[1]), referred to here as the 'atlas', and a dataset of 9 T_1 images. The images of the dataset are first globally aligned with the atlas using a robust block-matching affine registration algorithm. Then, a fine registration is performed using a dense transformation registration algorithm guaranteeing the invertibility of the global deformations [Stefanescu 04]. Statistics are performed on the deformations from the atlas to each of the subjects' geometry obtained in the non-linear registration step.

To compute logarithms of diffeomorphisms and exponential of vector fields, we used the fast algorithms presented in the former section. A scaling of 2^8 is used in both cases, and a maximal number of 10 iterations is used during each gradient descent.On the $256 \times 256 \times 60$ grid, using a Intel M processor 2.13GHz with 1 Go of RAM, an exponential was computed in typically $30s$ and a logarithm in 60 minutes.

Fig. 8.3 presents the results obtained on mean global deformations. We see in particular that the Log-Euclidean mean deformation tends to increase the size of ventricles, which are smaller in the atlas than in the dataset. We also compared the Log-Euclidean mean to the Euclidean mean of displacements, which is not guaranteed to be invertible. In this example, both means are quite close to each other, although locally, one can observe in the region of large mean deformations relative differences of the order of 30%, for example in the ventricles.

[1]Web site: http://www.bic.mni.mcgill.ca/brainweb/

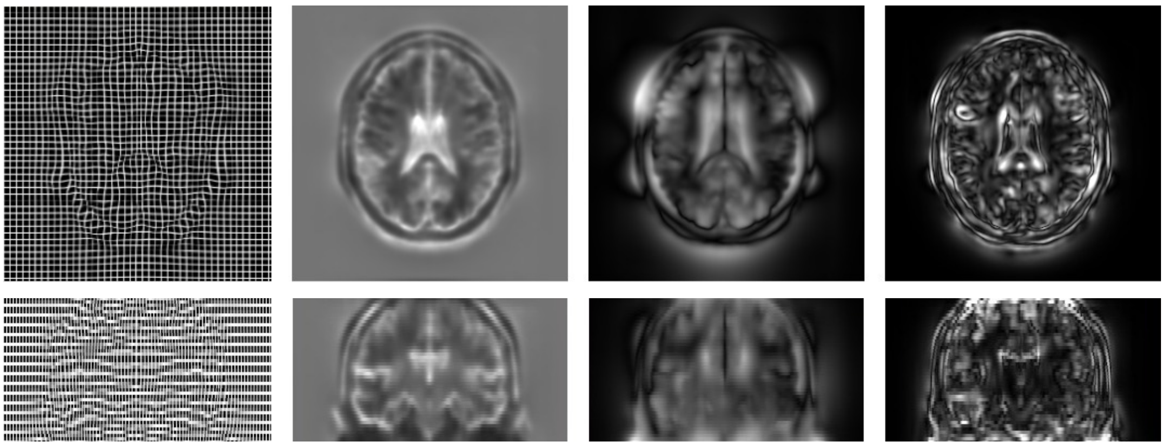

Figure 8.3: **Deformation of atlas with Log-Euclidean mean diffeomorphism. From left to right, images of:** regular grid deformed by Log-Euclidean mean deformations, Jacobian of Log-Euclidean mean deformations, norm of Log-Euclidean mean deformations, and finally norm of difference between mean Euclidean and Log-Euclidean deformations of atlas. The largest mean deformations are observed in the ventricles, on the cortex and the skull; this is due to the anatomical differences between the atlas and the population. On the Jacobian map, we see in particular that high values are obtained in the ventricles: this comes the fact that the atlas has on average smaller ventricles than in the population. In this example, both means are quite close to each other, although locally, one can observe in the region of large mean deformations relative differences of the order of 30%, for example in the ventricles.

8.5 Conclusions and Perspectives

In this Chapter, we have presented the generalization to diffeomorphisms of our Log-Euclidean framework, based on the generalization of the notion of *principal logarithm* to invertible deformations. This logarithm is a simple 3D vector field, and is well-defined for diffeomorphisms close enough to the identity. In the set of diffeomorphisms whose logarithm is well-defined, one can perform Euclidean operations, since these logarithms are simple vector fields. This yields the infinite-dimensional analogous of the Log-Euclidean framework proposed in [Arsigny 05a] for tensors. As a consequence, this framework can be used to perform *vectorial* statistics on diffeomorphisms which always preserve the invertibility constraint, contrary to Euclidean statistics on displacement fields.

From a theoretical point of view, our non-linear generalization of the Log-Euclidean framework still poses some problems due to the infinite-dimensional nature of diffeomorphsims that are yet to be completely solved. Addressing these technical problems will be one of our priorities in future work. In particular, it would be desirable to have a simple characterization of the conditions under which the principal logarithm of diffeomorphism exists, similarly to the simple criterion we have on the eigenvalues of invertible matrices.

However, from a practical point of view, we have presented in this work two efficient algorithms which successfully generalize to the non-linear case two popular algorithms used to compute the matrix exponential and logarithm. In practice, this allows for example the computation of Log-Euclidean means of 3D global deformations, as we have shown in the context of a registration experiment between an atlas an a database of 9 subjects. This type of statistics could prove very useful to quantitatively compare registration algorithms, or to constrain them. This opens the way to a consistent integrative framework for statistics in computational anatomy. In the domain of image and shape statistics, the Log-Euclidean framework for diffeomorphisms could provide an interesting setting to build models with a constant topology.

Chapter 9

Conclusions and Perspectives

Contents

9.1 Contributions and Publications 177
9.1.1 General Log-Euclidean Framework for Tensors 178
9.1.2 Diffusion Tensor Processing 179
9.1.3 Structure Tensor Processing 179
9.1.4 Statistics on Anatomical Variability via Tensors 180
9.1.5 Statistics on Deformation Tensors in Non-Linear Registration 181
9.1.6 Polyrigid and Polyaffine Transformations 182
9.1.7 Statistics in Finite-Dimensional Lie Groups 184
9.1.8 Statistics on Diffeomorphisms 185
9.2 Short Term Perspectives 185
9.2.1 Tensor Processing 185
9.2.2 Locally Rigid or Affine Transformations 186
9.2.3 General Statistics in Lie Groups 187
9.2.4 Statistics on Diffeomorphisms 187
9.3 More Global Perspectives 187
9.3.1 Natural Extensions of this Work 187
9.3.2 Bi-Invariant Framework vs. Riemannian Geometry 190
9.3.3 Medical Image Registration 191
9.4 Epilogue 191

9.1 Contributions and Publications

In this Ph.D. thesis, we have mainly concentrated on developing novel and rigorous mathematical frameworks to process various types of data belonging to Lie groups. We have not relied in all cases on Riemannian geometry, which can be computationally expensive or lacks desirable properties. Instead, we have generally used *algebraic-oriented* approaches, i.e. approaches rooted in the algebraic properties of the non-linear spaces we have considered. Relying on the algebraic properties of these groups, we have endeavored to find an adequate balance between desirable theoretical properties and simplicity, which is of paramount importance in practice.

In this manuscript, we have presented in detail our methodological contributions, along with some of their applications. The content of Chapters 3 to 8 is essentially novel and and gave rise to seven publications as first author during the three years of the Ph.D.

We have also collaborated with others as co-author, mainly to apply our novel tools in different contexts, which has led to a number of publications whose content has been but briefly described if at all in the previous chapters of this document. This work is put into light in this Section.

9.1.1 General Log-Euclidean Framework for Tensors

We have presented in Chapter 3 the theoretical properties of a novel and general Riemannian processing framework for tensors, called *Log-Euclidean*. Our approach is based on a *novel Lie group structure* for tensors, which can smoothly be extended into a vector space structure. Remarkably, the novel *algebraic structure* we have proposed for symmetric and positive-definite matrices is *compatible* with the usual algebraic properties of this set: the inverse of a tensor is its usual inverse and the matrix exponential is the group exponential of our Lie group structure.

Our novel framework does not introduce any superfluous complexity, since Log-Euclidean Riemannian computations can be converted into Euclidean ones once tensors have been transformed into their matrix logarithms, which makes classical Euclidean processing algorithms particularly simple to recycle.

We have also analyzed theoretically the similarities and differences between the affine-invariant mean of [Pennec 06b] and the Log-Euclidean mean. They are quite similar, since they have the same determinant, which is the classical geometric mean of the determinants of the averaged SPD matrices. They even coincide when there is enough commutativity (in the sense of matrix multiplication) in the data, and yet are different in general. We have proved that when the data are close enough to a multiple of the identity, Euclidean means are strictly more anisotropic than affine-invariant means. Also, we have presented the general statistical Log-Euclidean framework for tensors, which is now used on a regular basis by several teams [Peyrat 06b, Goodlett 06, Lepore 06].

The theoretical aspects of this work will be published in the SIAM Journal for Matrix Analysis and Applications. An INRIA research report on this topic is also available. Moreover, a French patent is pending on a general image processing device based on our Log-Euclidean framework:

- Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM Journal on Matrix Analysis and Applications, 2006.* Note: in press.

- Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Fast and Simple Computations on Tensors with Log-Euclidean Metrics. Research Report RR-5584, INRIA, Sophia-Antipolis, France, May 2005.

- Vincent Arsigny, Xavier Pennec, Pierre Fillard, and Nicholas Ayache. Dispositif perfectionné de traitement ou de production d'images de tenseurs. French patent filing number 0503483, April 2005.

9.1.2 Diffusion Tensor Processing

In Chapter 4, we have focused on the application of the Log-Euclidean framework for tensors to the processing of a specific type of tensors: *diffusion tensors*. An experimental comparison has been carried out between our framework and the Euclidean and affine-invariant frameworks [Pennec 06b] on interpolation and regularization tasks on synthetic and clinical 3D DTI data. Both Riemannian results are very close and are substantially better than the Euclidean ones. Log-Euclidean results are obtained in a much faster and simpler way than in the affine-invariance case, which makes this framework particularly appealing for the processing of this kind of tensor.

This work was presented during the peer-reviewed international conference MICCAI'05 and was published in the international journal Magnetic Resonance in Medicine:

- Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors. *Magnetic Resonance in Medicine, 56(2):411-421, August 2006.*

- Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Fast and Simple Calculus on Tensors in the Log-Euclidean Framework. *In J. Duncan and G. Gerig, editors, Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part I, volume 3749 of LNCS, Palm Springs, CA, USA, October 26-29, pages 115-122, 2005. Springer Verlag.*

With Pierre Fillard and others, inspired by [Wang 04], I have worked as co-author on the joint estimation and smoothing of diffusion tensors. This type of procedure is largely facilitated by the use of our Log-Euclidean framework, which allows to process tensors in a vectorial way. For the first time to our knowledge, we have proposed a rigorous way of taking into account the Rician noise of MR scans during the estimation of diffusion tensors. The positive impact on the tracking of white matter fibers in the brain and the spine has also been demonstrated in this work.

Our work was presented at the peer-reviewed Third IEEE Symposium on Biomedical Imaging and has been submitted to NeuroImage. A preprint of this work also appeared as an INRIA research report:

- Pierre Fillard, Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Clinical DT-MRI Estimation, Smoothing and Fiber Tracking with Log-Euclidean Metrics. *NeuroImage, 2006.* Note: Submitted.

- Pierre Fillard, Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Clinical DT-MRI estimation, smoothing and fiber tracking with log-Euclidean metrics. *In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), Crystal Gateway Marriott, Arlington, Virginia, USA, pages 786-789, April 2006.*

- Pierre Fillard, Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Joint Estimation and Smoothing of Clinical DT-MRI with a Log-Euclidean Metric. Research Report RR-5607, INRIA, Sophia-Antipolis, France, June 2005.

9.1.3 Structure Tensor Processing

With Pierre Fillard and others, I have investigated as co-author the use of affine-invariant metrics for the processing of *structure tensors*, which are a powerful tool which can be used in such image processing tasks as edge or corner detection and anisotropic smoothing. In an article presented at the peer-reviewed international workshop DSSCV'05 [Fillard 05a], we have in particular compared the images of gradient of structure tensors provided by affine-invariant and Euclidean metrics.

In the presence of a substantial amount of noise, we have found that affine-invariant metrics are less adapted for the processing of structure tensors than Euclidean ones, essentially because they give the same 'weight' to large and small tensors. This is problematic, because small tensors are mainly generated by the noise, whereas large tensors reflect in most cases significant structures in the image. On the contrary, in the case of a very high signal-to-noise ratio, affine-invariant metrics provide a magnification of low-contrast structures present in the image, which could prove valuable depending on the application considered.

- Pierre Fillard, Vincent Arsigny, Nicholas Ayache, and Xavier Pennec. A Riemannian Framework for the Processing of Tensor-Valued Images. *In Ole Fogh Olsen, Luc Florak, and Arjan Kuijper, editors, Deep Structure, Singularities, and Computer Vision (DSSCV), number 3753 of LNCS, pages 112-123, June 2005. Springer Verlag.*

9.1.4 Statistics on Anatomical Variability via Tensors

With Pierre Fillard and other co-authors (including Paul M. Thompson and Kiralee M. Hayashi from our associated team LONI at UCLA), we have developed a new mathematical model of normal brain variation based on a large set of cortical sulcal landmarks (72 per brain) delineated in each of 98 healthy human subjects scanned with 3D MRI. We have proposed an original method to compute an average representation of the sulcal curves, which constitutes the mean anatomy in this context. After global (affine) alignment of the individual data across subjects, the second order moment distribution of the sulcal position is modeled as a sparse field of covariance tensors, called here *variability tensors*.

To extrapolate this information to the full brain, we first used affine-invariant Riemannian metrics. We have generalized radial basis function interpolation and harmonic diffusion partial differential equations to tensor fields. As a result, we have obtained a dense 3D variability map which agrees well with prior results on smaller samples of subjects. "Leave one (sulcus) out" tests have shown that our model is globally able to recover the missing information on brain variation when there is a consistent neighboring pattern of variability. Finally, we have proposed an innovative method to analyze the asymmetry of brain variability. As expected, the greatest asymmetries have been found in regions that include the primary language areas. Interestingly, any such asymmetries in anatomical variance, if it remains after anatomical normalization, could explain why there may be greater power to detect group activation in one hemisphere versus the other in fMRI studies. Future applications of this work include the detection of genetic and demographic factors that contribute to brain structure variance, abnormality detection in individuals and groups, and improved nonlinear registration techniques that draw on tensor-valued statistical information regarding brain variation.

This work was presented in 2005 at the peer-reviewed international conference IPMI'05 and will be published in the international journal NeuroImage. An INRIA research report of this work was also recently published:

- Pierre Fillard, Vincent Arsigny, Xavier Pennec, Kiralee M. Hayashi, Paul M. Thompson, and Nicholas Ayache. Measuring Brain Variability by Extrapolating Sparse Tensor Fields Measured on Sulcal Lines. *NeuroImage, 2006.* Note: In press.

- Pierre Fillard, Vincent Arsigny, Xavier Pennec, Paul Thompson, and Nicholas Ayache. Extrapolation of Sparse Tensor Fields: Application to the Modeling of Brain Variability. *In Gary Christensen and Milan Sonka, editors, Proc. of Information Processing in Medical Imaging 2005 (IPMI'05), volume 3565 of LNCS, Glenwood springs, Colorado, USA, pages 27-38, July 2005. Springer.*

- Pierre Fillard, Vincent Arsigny, Xavier Pennec, Kiralee M. Hayashi, Paul M. Thompson, and Nicholas Ayache. Measuring Brain Variability by Extrapolating Sparse Tensor Fields Measured on Sulcal Lines. Research Report 5887, INRIA, Avril 2006.

9.1.5 Statistics on Deformation Tensors in Non-Linear Registration

I have also participated as co-author in the development of two novel frameworks to inject statistics in non-linear registration. Both approaches are based on first computing *local* statistics on *deformation tensors* resulting from prior registration experiments and then imbedding these statistics as prior knowledge in a non-linear registration algorithm. Both of these frameworks were presented during the peer-reviewed international conference MICCAI'05.

In Pennec *et al.* [Pennec 05], statistics are performed on deformation tensors in our Log-Euclidean framework (i.e., a mean and a covariance matrix are computed for each voxel). Then, these statistics are injected in a Riemannian and anisotropic generalization of the classical St Venant-Kirchoff elastic energy, called *Riemannian elastic energy*. The idea is simply to replace the usual isotropic Euclidean distance between the identity and the Cauchy-Green deformation tensor by an anisotropic Riemannian distance (here a Log-Euclidean Mahalanobis distance). Preliminary results (without any actual statistics yet) have shown that this framework can be quite easily implemented in a non-rigid registration algorithms.

In Commowick *et al.* [Commowick 05], simple statistics on the natural logarithms of the determinant of deformation tensors and on the matrix logarithm of Cauchy-Green deformation tensors are proposed to quantify in an isotropic and anisotropic way the local *deformability* of tissues. Then, these simple measures (a single scalar or tensor per voxel, vs. a mean tensor plus a 6x6 covariance matrix in Pennec *et al.*) are used in a registration algorithm with an inhomogeneous regularization [Stefanescu 04], to replace heuristic scalar maps of deformability. Our experiments, carried out on an image database of 36 patients with brain tumors, have shown quantitatively better segmentations with the proposed method, and also qualitatively more consistent from an anatomical point of view.

- Xavier Pennec, Radu Stefanescu, Vincent Arsigny, Pierre Fillard, and Nicholas Ayache. Riemannian Elasticity: A statistical regularization framework for non-linear registration. *In J. Duncan and G. Gerig, editors, Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part II, volume 3750 of LNCS, Palm Springs, CA, USA, October 26-29, pages 943-950, 2005. Springer Verlag.*

- Olivier Commowick, Radu Stefanescu, Pierre Fillard, Vincent Arsigny, Nicholas Ayache, Xavier Pennec, and Grégoire Malandain. Incorporating Statistical Measures of Anatomical Variability in Atlas-to-Subject Registration for Conformal Brain Radiotherapy. *In J. Duncan and G. Gerig, editors, Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part II, volume 3750 of LNCS, Palm Springs, CA, USA, October 26-29, pages 927-934, 2005. Springer Verlag.*

9.1.6 Polyrigid and Polyaffine Transformations

In Chapter 5, we have presented our original framework for polyrigid and polyaffine transformations. These transformations efficiently code for locally rigid or affine deformations with a small number of flexible and intuitive parameters. There exists very few classes of transformations in the literature that are *intrinsically* invertible, and guaranteeing the invertibility of these novel transformations is certainly our key contribution here.

From a mathematical point of view, we have shown that our novel transformations are diffeomorphisms, and smooth with respect to their parameters. For the numerical implementation of these transformations, we have devised a new and flexible numerical scheme to allow a trade-off between computational efficiency and closeness to the ideal diffeomorphism. In this context, we have also derived an effective optimization strategy of the transformations which demonstrates that this new tool is highly suitable for inference. The whole framework is exemplified successfully with the registration of histological slices, providing results comparable to [Pitiot 06], obtained with a completely different approach.

This work has been published in the international journal Medical Image Analysis, and was presented at the peer-reviewed international conference MICCAI'03, where this work was awarded the 'Best Student Presentation in Medical Image Processing and Visualization'. An INRIA research report on this topic was also published to ensure a timely dissemination of this innovative approach.

- Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Polyrigid and Polyaffine Transformations: a Novel Geometrical Tool to Deal with Non-Rigid Deformations - Application to the registration of histological slices. *Medical Image Analysis, 9(6):507-523, December 2005.*

- Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Polyrigid and Polyaffine Transformations: A New Class of Diffeomorphisms for Locally Rigid or Affine Registration. *In Randy E. Ellis and Terry M. Peters, editors, Proc. of MICCAI'03, Part II, volume 2879 of LNCS, Montreal, pages 829-837, November 2003. Springer Verlag.*

- Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. A novel family of geometrical transformations: Polyrigid transformations. Application to the registration of histological slices. Research report 4837, INRIA, 2003.

In Chapter 6, we have presented a novel framework to fuse rigid or affine components into a global transformation, called *Log-Euclidean polyaffine*. Similarly to our original polyaffine framework, it guarantees the *invertibility* of the result. However, contrary to our original framework, this is achieved with very intuitive properties: for example the inverse of a Log-Euclidean polyaffine transformation (LEPT) is a LEPT with identical weights and inverted affine components. Moreover, this novel fusion is *affine-invariant*, i.e. does not depend on the choice of coordinate system. We have also shown that contrary to previous polyaffine transformations, the specific properties of LEPTs allow their fast computations on regular grids, with an algorithm called the 'Fast Polyaffine Transform' (FPT), whose efficiency is somehow comparable to that of the Fast Fourier Transform.

We have also detailed in Chapter 6 a Log-Euclidean framework for rigid and affine transformations, as well as its extension to any finite-dimensional real Lie group. This framework generalizes to linear transformations the Log-Euclidean framework described in Chapter 3 for tensors.

We recently found out that in 2002, Alexa already proposed in [Alexa 02] to process linear geometrical transformations via their logarithms, in the context of the interpolation of transformations for computer graphics, and had also suggested to perform statistics on these transformations via their logs. One should note that our approach, which we developed independently, goes considerably deeper into the analysis of the properties of this framework. In particular, we were the first to our knowledge to put into light the geometric interpolation of determinants provided by the Log-Euclidean mean, as well as the *invariance properties* of this framework (i.e., inversion-invariance and affine-invariance). The extension of the Log-Euclidean framework to abstract Lie groups presented in this Thesis is also entirely novel to our knowledge.

Our Log-Euclidean polyaffine framework has been presented at the peer-reviewed international workshop WBIR'06, and the most of the results obtained in Chapter 6 were published in an INRIA research report. A journal publication of this work is currently being prepared.

- Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A Log-Euclidean Polyaffine Framework for Locally Rigid or Affine Registration. *In J.P.W. Pluim, B. Likar, and F.A. Gerritsen, editors, Proceedings of the Third International Workshop on Biomedical Image Registration (WBIR'06), volume 4057 of LNCS, Utrecht, the Netherlands, pages 120-127, 9 - 11 July 2006. Springer Verlag.*

- Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A Fast and Log-Euclidean Polyaffine Framework for Locally Affine Registration. Research Report RR-5865, INRIA Sophia-Antipolis, March 2006.

As mentioned in Chapter 6, we have worked as a co-author with Olivier Commowick (Ph.D. student in the Asclepios team and also working for DOSISoft SA, Cachan, France), toward drastically reducing the cost of using polyaffine transformations in 3D medical image registration procedures. To obtain short computation times (typically 10 minutes for whole 3D volumes in the locally affine case), the locally affine registration algorithm presented in [Commowick 06a, Commowick 06b] estimates affine components using the *direct fusion*, and relies on a Log-Euclidean regularization scheme of local affine components. The FPT is used in a *final step* to ensure the invertibility of the final transformation, as well as to compute its inverse.

In the journal version of this work [Commowick 06b], two applications of our fast and robust 3D polyaffine registration algorithm are considered. First, bone registration in the lower abdomen area. Second, the segmentation of critical brain structures in the context of conformal radiotherapy planning. In both cases, it is shown that our locally affine registration approach yields the same accuracy in the structures of interest as the dense transformation algorithm of [Stefanescu 04], but with *much smoother deformations*, which is more satisfactory from an anatomical point of view. This work has also been presented during the peer-reviewed Third IEEE Symposium on Biomedical Imaging (ISBI'06) [Commowick 06a]:

- Olivier Commowick, Vincent Arsigny, Jimena Costa, Nicholas Ayache, and Grégoire Malandain. An Efficient Locally Affine Framework for the Registration of Anatomical Structures. *Submitted to Medical Image Analysis, 2006.*

- Olivier Commowick, Vincent Arsigny, Jimena Costa, Nicholas Ayache, and Grégoire Malandain. An Efficient Locally Affine Framework for the Registration of Anatomical Structures. *In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), Crystal Gateway Marriott, Arlington, Virginia, USA, pages 478-481, April 2006.*

9.1.7 Statistics in Finite-Dimensional Lie Groups

In Chapter 7, we have presented a general framework to define rigorously a novel type of mean in Lie groups, called the *bi-invariant mean*. This mean enjoys many desirable *invariance properties*, which generalize to the non-linear case the properties of the arithmetic mean: it is invariant with respect to left- and right-multiplication, as well as inversion. Previously, this type of mean was only defined in Lie groups endowed with a bi-invariant Riemannian metric, like compact Lie groups such as the group of rotations. But Riemannian bi-invariant metrics do not always exist. In particular, we have proved in this work that such metrics do not exist in any dimension for *rigid transformations*, which form but the most simple Lie group involved in medical image registration.

To overcome the lack of existence of bi-invariant Riemannian metrics for many Lie groups, we have proposed in this thesis to define bi-invariant means in *finite-dimensional real Lie groups* via a general *barycentric equation*, whose solution is by definition the bi-invariant mean. The intuition of the existence of such a mean was first given by R.P.Woods (without precise definition) in [Woods 03], along with an efficient algorithm for computing it (without proof of convergence), in the case of *matrix* groups.

We have shown the existence and uniqueness of this novel type of mean provided the dispersion of the data is small enough, as well as the convergence of the efficient iterative algorithm of Woods. We have also briefly investigated how centered empirical higher order moments can be computed based on this novel notion of mean.

In the case of rigid transformations, we have given a simple a precise criterion for the general existence and uniqueness of the bi-invariant mean, which happens to be the same as for rotations. We have also given closed forms for the bi-invariant mean in a number of simple cases, including 2D rigid transformations. For general linear transformations, we have shown that the bi-invariant mean is a generalization of the (scalar) geometric mean like the Log-Euclidean mean.

In practice, one should note that both Log-Euclidean and bi-invariant frameworks can *always* be efficiently used to compute statistics in matrix Lie groups (provided of course that the dispersion of the data is not too large, but this restriction also applies to Riemannian frameworks), whereas in the Riemannian case, the tractability of such an approach depends very much on the form taken by geodesics. For rigid transformations, left-invariant geodesics are linked to the matrix exponential in a straightforward way and thus can be simply computed and used in practice. However, to our knowledge, this is not the case for left-invariant Riemannian metrics on *affine transformations*, which makes the use of such metrics quite difficult in this case. No such problem occurs with either the Log-Euclidean or the bi-invariant framework, which only rely on computations based on the *matrix* exponential and logarithm.

We have used this new type of mean to define a novel class of polyaffine transformations, called *left-invariant polyaffine*, which allows to fuse local rigid or affine components arbitrarily far away from the identity, contrary to Log-Euclidean polyaffine fusion, which we are presented in Chapter 6.

Most of this work was recently published in an INRIA research report:

- Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Bi-invariant Means in Lie Groups. Application to Left-invariant Polyaffine Transformations. Research Report RR-5885, INRIA Sophia-Antipolis, April 2006.

9.1.8 Statistics on Diffeomorphisms

In Chapter 8, we have generalized our Log-Euclidean framework to invertible geometrical deformations (i.e., diffeomorphisms) in order to provide simple tools to compute statistics on this special type of data. This yields in particular a closed form for the computation of mean diffeomorphisms, contrary to the iterative computation suggested in [Vaillant 04].

Our novel framework is based on the generalization to this type of data of the notion of *principal logarithm*. This logarithm is a simple 3D vector field, and is well-defined for diffeomorphisms close enough to the identity. This allows to perform *vectorial* statistics on diffeomorphisms, while preserving the invertibility constraint, contrary to Euclidean statistics on displacement fields.

Due to the infinite-dimensional nature of diffeomorphisms, our non-linear generalization of the Log-Euclidean framework to general invertible deformations still poses some theoretical problems that are yet to be completely solved. However, from a practical point of view, we have proposed two efficient algorithms to compute numerically logarithms of diffeomorphisms and exponentials of vector fields, whose accuracy has been studied on synthetic data. With these tools, for the first time to our knowledge, we have been able to compute a mean which takes into account the invertible nature of a set of high-dimensional 3D diffeomorphisms obtained with the algorithm of [Stefanescu 04], in the context of a registration experiment between an atlas an a database of 9 T1 MR images of the human brain.

This work was presented at the peer-reviewed international conference MICCAI'06 as well as in its satellite International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006):

- Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A Log-Euclidean Framework for Statistics on Diffeomorphisms. *In Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), LNCS, vol 4190, part I, pages 924-931, 2-4 October 2006, Springer-Verlag.*
- Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. Statistics on Diffeomorphisms in a Log-Euclidean Framework. *In Proc. of International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006), pages 16-17, 1st of October, 2006.*

9.2 Short Term Perspectives

However thorough we tried to be during this Ph.D., we did not have the time to cover all of the questions raised and all of the possibilities opened up by our work. In the near future, it would be interesting to go further in a number of directions, which are presented below.

9.2.1 Tensor Processing

Which Metrics for which Tensors? We have shown in this work that there are indeed several generalizations of the geometric mean to SPD matrices. Other variants may exist, and we will investigate other possible generalizations in future work. This is important, since situations in applied mathematics, mechanics, medical imaging, etc. where SPD matrices need to be processed are highly varied. As a consequence, the relevance of each generalization of the geometric mean and of the associated metric framework may depend on the application considered. We have already begun the assess the respective relevances of the Log-Euclidean and affine-invariant frameworks in DT-MRI [Arsigny 05a] and for structure tensors [Fillard 05a].

Study of Anatomical Variability. We have recently begun to use the Log-Euclidean framework to simplify the approach proposed in [Fillard 05c,Fillard 06d] to model and analyze the anatomical variability of the human brain cortex. Working in our vector space structure for tensors could allow for simple strategies to reduce the effect of the *aperture problem*, which induces in our current approach a substantial uncertainty in the estimation of the anatomical variability in the direction tangential to the sulcal lines of our database.

9.2.2 Locally Rigid or Affine Transformations

Methodological Issues

Updating Weight Functions with Simple Strategies. In Chapter 5, we have presented a general way of optimizing the parameters of polyrigid transformation for medical image registration. However, this approach is quite computationally expensive, and this has incited us to devise alternative strategies to drastically reduce this cost. This has led to the fast approach presented in [Commowick 06b], which we have briefly mentioned in Chapter 6. But in the latter setting, with weight functions, i.e. the geometry of the fused rigid or affine components, remain unchanged during the registration procedure. In future work, it would be desirable to devise a simple and adapted way of also optimizing these parameters, which are of a very different nature than those of the local linear transformations. To this end, it could be very interesting to adapt to our novel framework the ideas presented in [Pitiot 06] in the specific context of the piecewise affine registration of histological slices.

Numerical Accuracy of the FPT. In Chapter 6, we have presented a very efficient numerical algorithm, called the 'Fast Polyaffine Transform', to compute Log-Euclidean polyaffine transformations on regular grids. We have also experimentally verified and studied its convergence in a number of cases. However, it would be desirable to analyze the theoretical properties of this algorithm, in order to clarify the relationships between its numerical accuracy and the characteristics of the regular grid, the sampling step and the parameters of the transformation considered.

Applications

Anatomical Atlases. An application of polyrigid and polyaffine transformations is the building of new anatomical atlases, for example in the case of dataset presenting articulated structures. Using adapted transformations to establish correspondences between the various instances would surely lead to more accurate results. This approach was recently used and compared to other non-linear registration techniques in [Commowick 06c] in the context of radiotherapy planning.

Representing Arbitrary Diffeomorphisms. As in [Marsland 04] in the case of Geodesic Interpolating Splines (GIS), polyrigid and polyaffine transformations could be used to parameters arbitrary diffeomorphisms with a limited number of intuitive degrees of freedom.

Exactly like GIS, our novel transformations guarantee the invertibility of the resulting transformations. But they naturally code with very few parameters for a large panel of local deformations, whereas GIS are limited to local translations. This could prove a substantial advantage for polyaffine transformations in this specific task.

Shape Statistics. Similarly to the parameterization of shapes via 'M-reps' [Pizer 03], one could model the variability of a shape around its mean via the statistical analysis of these variations in a certain space of transformations. By choosing as adequately as possible this space of deformations, a model with a limited number of parameters could be derived.

Polyaffine transformations are a good candidate for doing so in a simple way, because they can code for a large variety of local deformations with few degrees of freedom: local rotations, translations, swellings, shearing, etc., while guaranteeing the invertibility of the resulting global transformation. This offers a larger panel of local deformations than M-reps, which require for example the modification of the parameters of *several* of its 'medial atoms' to code for local rotations or shearing, whereas only a *single* affine component component can be involved to do so in the polyaffine case.

9.2.3 General Statistics in Lie Groups

Quantitative Comparison of Existing Frameworks. In future work, we are planning to compare the statistics obtained via the bi-invariant mean and the Log-Euclidean framework to other types of statistics on rigid or affine transformations such as left-invariant Riemannian statistics, which are used in [Boisvert 06] in the context of the statistical analysis of the human spine scoliosis. Are there any significant differences between these various alternatives? Do they lead to substantially different results when used to constraint non-linear registration algorithms? This topic will be very interesting to investigate.

9.2.4 Statistics on Diffeomorphisms

Theoretical Issues. From a theoretical point of view, our non-linear generalization of the Log-Euclidean framework to general invertible deformations still poses some problems due to the infinite-dimensional nature of diffeomorphisms that are yet to be completely solved. Addressing these technical problems will be one of our priorities in future work. In particular, it would be desirable to have a simple characterization of the conditions under which the principal logarithm of diffeomorphism exists, similarly to the simple criterion we have on the eigenvalues of invertible matrices (e.g., could this linear criterion be directly transferred to the field of Jacobian matrices associated by differentiation to a diffeomorphism?).

Numerical Issues. Similarly to the FPT case, it would be highly desirable to analyze the convergence properties of our numerical algorithms for computing the exponential of a vector field and the logarithm of a diffeomorphism.

9.3 More Global Perspectives

In this thesis, we have presented quite a large number of approaches and frameworks, which are closely intertwined. We have gone much into fine details and endeavored to develop our ideas as far as we could, in different directions. We adopt here a more global perspective, to better visualize what are the interesting opportunities and directions of research opened up by this work, on a larger scale.

9.3.1 Natural Extensions of this Work

Global Picture

At the end of the Introduction, we had presented a simple way of representing our contributions, within a two-column table. In Fig. 9.1, we have added to this table the natural extensions of our work. They are presented below.

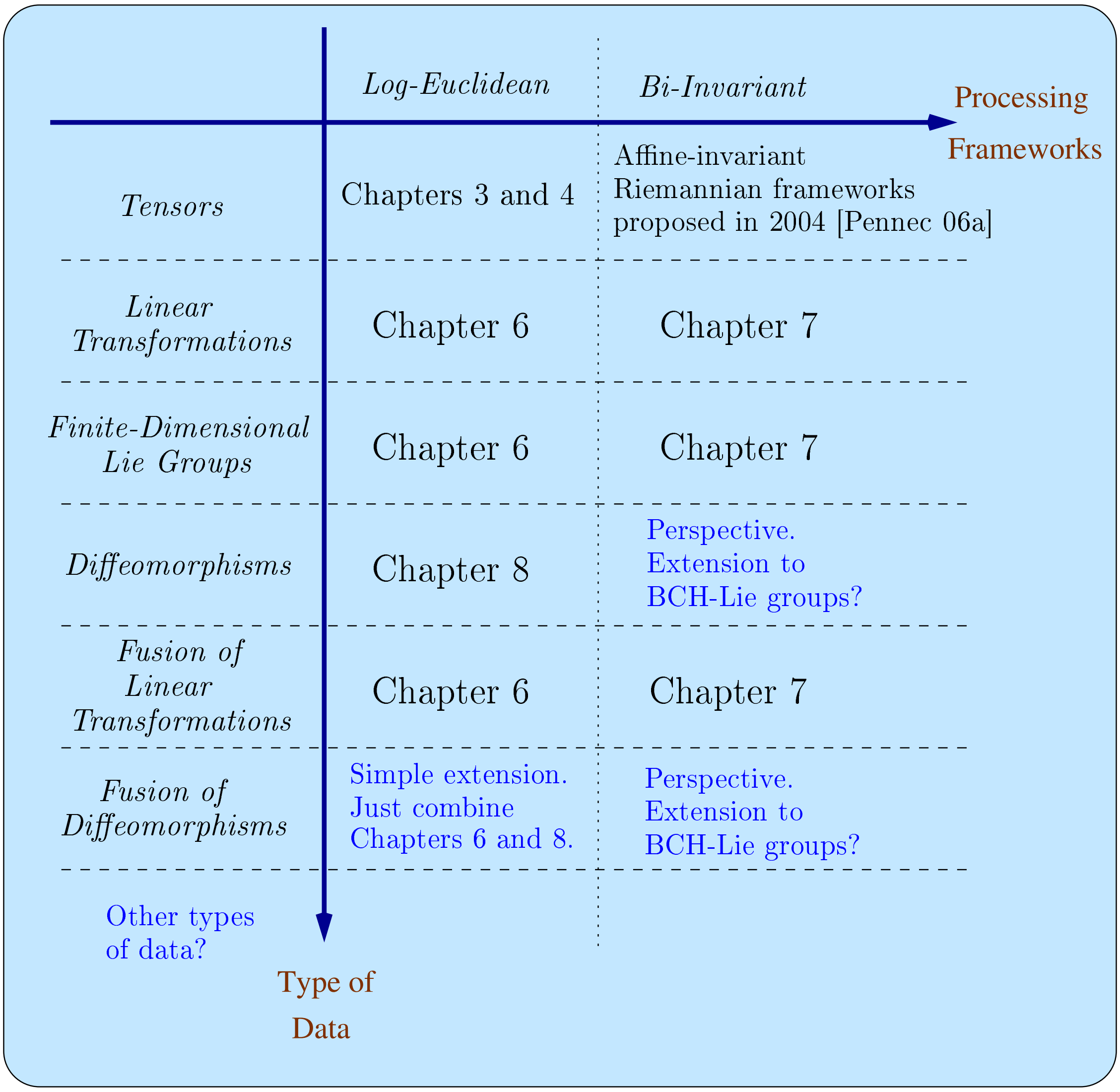

Figure 9.1: **Global Picture of this thesis**. **Lines: various types of data**, and **columns: processing frameworks** (either the Log-Euclidean one or the bi-invariant one). Chapter 5 does not appear in this table, because the polyaffine fusion of linear transformations presented in this chapter belongs neither to the Log-Euclidean nor the bi-invariant framework. The natural perspectives of this work are displayed in blue.

Bi-Invariant Means with Infinite Dimensions

In Chapter 7, we have presented a theory of bi-invariant means in finite-dimensional real Lie groups. To what extent could this type of mean be extended in the infinite-dimensional case, and in particular to diffeomorphisms? We saw in Chapter 8 that the logarithm could be defined without ambiguity in infinite-dimensional Lie groups which are *locally exponential*, i.e. where the exponential is a local diffeomorphism near zero.

To prove the existence and uniqueness of the bi-invariant mean, we have not only used the fact that finite-dimensional Lie groups are locally exponential, but also the fact that the group multiplication is *analytic* in exponential coordinates. This is essentially what the Baker-Campbell-Hausdorff formula presented in Section 2.3.6 is all about. Does there exist any infinite-dimensional Lie groups with this property? The answer is positive, as explained in [Glöckner 06], and locally exponential Lie groups with this property are called *BCH-Lie groups*. This property is stronger than that of being locally exponential: all locally exponential Lie groups are *not* BCH.

As a consequence, it would be very interesting to investigate whether one can generalize our construction of the bi-invariant mean to general BCH-Lie groups. Provided there exist some interesting groups of diffeomorphisms which are BCH, this could provide a statistical framework for diffeomorphisms with particularly natural invariance properties.

Fusion of Local Diffeomorphic Transformations

In this thesis, we have presented several frameworks, called *polyaffine*, to fuse local linear transformations into a global invertible transformation. Our Log-Euclidean, left-invariant and right-invariant polyaffine frameworks are mainly based on the fact that one can construct of fusion PDE from the *logarithms* of the local linear components. Consequently, one can generalize these approaches to local *diffeomorphic* transformations, provided their logarithms are well-defined.

Thus, one can perform of *Log-Euclidean* fusion of a finite number of local diffeomorphic components $(\Phi_i(x), w_i(x))$ via the following ODE:

$$\dot{x} = \sum_i w_i(x). \log(\Phi_i)(x),$$

which generalizes to local diffeomorphisms the Log-Euclidean polyaffine ODE of Eq. (6.7). To generalize LIPTs and RIPTs, it would be necessary to rely on a generalization to diffeomorphisms of the bi-invariant mean. As explained above, this could prove possible for BCH-Lie groups of diffeomorphisms.

This type of fusion could prove very interesting to take into account the *respective deformabilities* of anatomical structures. Some of them could be deformed using very few degrees of freedom (e.g., in a linear way), some a little more (e.g., diffeomorphic B-Splines, locally affine deformations) and others could be deformed much more. This type of fusion allows to 'glue' all of these local transformations into a global invertible deformation.

Processing other Types of Data

We briefly present here two types of data for which a general and rigorous processing framework with completely satisfying invariance properties is still to be developed.

More Sophisticated Diffusion Models. A generalization of the Log-Euclidean or bi-invariant (i.e., affine-invariant) frameworks for tensors would be very valuable to process to the more sophisticated diffusion models proposed in the recent years in the diffusion imaging community. These models include generalized diffusion tensors [Özarslan 03] and Q-balls [Tuch 04].

To what extent would it possible to rely on a Lie group approach to process these types of data? To do so in a direct way, it would be necessary to identify either a Lie group structure for these spaces or a larger Lie group in which they can be embedded. Is it possible to do so in a relevant way, i.e. in a manner which is compatible with the physical properties of these data? This remains to be seen.

Alternatively, to process this type of data, one could rely on other powerful approaches such as constructing Riemannian metrics which are *invariant* with respect to an *homogeneous manifold* structure, as done in [Pennec 06a] to define affine-invariant metrics on tensors. Another method consists in using *Fisher-Rao* metrics on these sophisticated diffusion models, which can be looked upon as parametric probability distributions. This was done in [Lenglet 06a] in the case of diffusion tensors, which yielded (again) affine-invariant metrics.

Toeplitz Hermitian Positive-Definite Matrices. As suggested recently to us by F. Barbaresco (Thales group, France), our Log-Euclidean framework can be extended in a straightforward way to positive-definite Hermitian matrices, i.e. the complex equivalent of tensors. This comes from the fact that the (complex) matrix exponential provides a one-to-one correspondence between Hermitian matrices and positive-definite Hermitian matrices, exactly as in the real case.

In Doppler radar technology however, *Toeplitz* positive-definite Hermitian matrices play a very important role, and performing rigorous statistics on these matrices with natural invariance properties would be very valuable. Unfortunately, Log-Euclidean means of such matrices are not Toeplitz in general, and it would be very interesting to see how our ideas could be adapted to the specific properties of this matrix subspace.

9.3.2 Bi-Invariant Framework vs. Riemannian Geometry

In Chapter 7, in the general Lie group case, we have shown the existence and uniqueness of the bi-invariant mean in finite-dimensional Lie groups, provided the dispersion of the data is not too large. In specific cases, in particular rigid transformations, we have been able to determine more precisely when such means are well-defined. Furthermore, we have shown that it is possible to compute centered higher order moments from this notion of mean, which opens the way for a general bi-invariant statistical framework.

Our construction of the bi-invariant mean essentially relies on the idea that their is a direct analogy between translated version of one-parameter subgroups and the geodesics of Riemannian geometry. Indeed, we have seen in Chapters 2 and 7 that the differential properties of the group and Riemannian exponential are very much alike. To what extent are these two approaches similar?

In [Woods 03], Woods suggested that one-parameter subgroups could be interpreted as geodesics for some bi-invariant *semi*-Riemannian metrics. Furthermore, he claimed that many of the results of Riemannian geometry could be generalized without modification to the semi-Riemannian case. Unfortunately, this point of view was only sketched in [Woods 03], and to our knowledge did not give rise to any mathematical or technical publication clarifying the intuitions of Woods.

As a consequence, it would be highly desirable to go further than the results of Chapter 7, and in particular, to systematically compare the properties of the bi-invariant (algebraic) framework to that of classical Riemannian geometry. For instance, is the measure of the (algebraic) *cut locus* of null measure (i.e., is the set of elements for which the logarithm is not well-defined negligible in the connected component of the identity)? Does there exist a general and precise criteria for the well-definedness of the bi-invariant mean in terms of *geodesic balls* as in the Riemannian case? Only then will we know whether our bi-invariant framework is as powerful as that of Riemannian geometry to perform statistics in Lie groups.

9.3.3 Medical Image Registration

In the medical imaging community, a vast number of registration techniques have been proposed during the last fifteen years to address the difficult task of establishing correspondences between different anatomies. With rigorous frameworks to perform a statistical analysis on the results of registration algorithms, i.e. on geometrical transformations, one can hope to better understand in the near future the differences between these algorithms from a *quantitative* point of view. A better understanding of the characteristics of these tools will certainly pave the way for more clever and relevant registration techniques, better taking into account the specificity of each imaging modality, the deformability of the different types of biological tissues considered and the interdependence in the variation in size and shape of organs from one individual to another, from one gender to another, from one age in life to another.

More modestly, with the tools presented in this work, one can perform statistics on linear transformations and even on diffeomorphisms whose logarithm is well-defined. We hope that this will help develop more comparison techniques between linear and non-linear registration algorithms, such as the one presented in [Nicolau 03, Glatard 06] in the rigid case. These statistics could also help to better constraint non-rigid registration algorithms (in particular constraint them more *globally*), with respect to what we begun to do in [Pennec 05] and [Commowick 05] with (*local*) statistics on Cauchy-Green deformation tensors. With global statistics, one can hope to capture the correlations in deformability between organs, which is of paramount importance to ensure that the deformations induced during the registration procedure are truly anatomically consistent.

9.4 Epilogue

During this work, we have striven to propose simple and rigorous frameworks for the processing of various types of data which arise in the growing discipline of *computational anatomy*. The results we have achieved in this thesis were made possible on the one hand by the multiplicity of situations and opportunities encountered in this young research field, and on the other hand by the inexhaustible abundance of the abstract mathematical world, which provide us with incredibly efficient ways of modeling and dealing with practical problems.

We hope that this double profusion and the many challenges still present in this domain will continue to attract the bright minds it needs to keep on progressing and continue to provide the medical community with some of the novel tools and insights it still lacks today.

Bibliography

[Alauzet 03] Frédéric Alauzet. *Adaptation de maillage anisotrope en trois dimensions : application aux simulations instationnaires en mécanique des fluides.* PhD thesis, Montpellier 2, october 2003. in French.

[Alexa 02] Marc Alexa. *Linear combination of transformations.* In Tom Appolloni, editor, SIGGRAPH, pages 380–387. ACM, 2002.

[Allassonnière 05] Stéphanie Allassonnière, Alain Trouvé & Laurent Younes. *Geodesic Shooting and Diffeomorphic Matching Via Textured Meshes.* In EMM-CVPR, pages 365–381, 2005.

[Arsigny 03a] Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *A novel family of geometrical transformations: Polyrigid transformations. Application to the registration of histological slices.* Research report 4837, INRIA, 2003.

[Arsigny 03b] Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *Polyrigid and Polyaffine Transformations: A New Class of Diffeomorphisms for Locally Rigid or Affine Registration.* In Randy E. Ellis & Terry M. Peters, editors, Proc. of MICCAI'03, Part II, volume 2879 of *LNCS*, pages 829–837, Montreal, November 2003. Springer Verlag. MICCAI 2003 Best Student Award in Image Processing and Visualization.

[Arsigny 05a] Vincent Arsigny, Pierre Fillard, Xavier Pennec & Nicholas Ayache. *Fast and Simple Calculus on Tensors in the Log-Euclidean Framework.* In J. Duncan & G. Gerig, editors, Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part I, volume 3749 of *LNCS*, pages 115–122, Palm Springs, CA, USA, October 26-29, 2005. Springer Verlag.

[Arsigny 05b] Vincent Arsigny, Pierre Fillard, Xavier Pennec & Nicholas Ayache. *Fast and Simple Computations on Tensors with Log-Euclidean Metrics.* Research Report RR-5584, INRIA, Sophia-Antipolis, France, May 2005.

[Arsigny 05c] Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *Polyrigid and Polyaffine Transformations: a Novel Geometrical Tool to Deal with Non-Rigid Deformations - Application to the registration of histological slices.* Medical Image Analysis, vol. 9, no. 6, pages 507–523, December 2005.

[Arsigny 05d] Vincent Arsigny, Xavier Pennec, Pierre Fillard & Nicholas Ayache. *Dispositif perfectionné de traitement ou de production d'images de tenseurs.* French patent filing number 0503483, April 2005.

[Arsigny 06a] Vincent Arsigny, Olivier Commowick, Xavier Pennec & Nicholas Ayache. *A Fast and Log-Euclidean Polyaffine Framework for Locally Affine Registration.* Research report RR-5865, INRIA Sophia-Antipolis, March 2006.

[Arsigny 06b] Vincent Arsigny, Olivier Commowick, Xavier Pennec & Nicholas Ayache. *A Log-Euclidean Framework for Statistics on Diffeomorphisms.* In Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), Part I, no. 4190 in LNCS, pages 924–931, 2-4 October 2006.

[Arsigny 06c] Vincent Arsigny, Olivier Commowick, Xavier Pennec & Nicholas Ayache. *A Log-Euclidean Polyaffine Framework for Locally Rigid or Affine Registration.* In J.P.W. Pluim, B. Likar & F.A. Gerritsen, editors, Proceedings of the Third International Workshop on Biomedical Image Registration (WBIR'06), volume 4057 of *LNCS*, pages 120–127, Utrecht, The Netherlands, 9 - 11 July 2006. Springer Verlag. To appear.

[Arsigny 06d] Vincent Arsigny, Olivier Commowick, Xavier Pennec & Nicholas Ayache. *Statistics on Diffeomorphisms in A Log-Euclidean Framework.* In X. Pennec & S. Joshi, editors, Proc. of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006), pages 16–17, 1st of October 2006.

[Arsigny 06e] Vincent Arsigny, Pierre Fillard, Xavier Pennec & Nicholas Ayache. *Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices.* SIAM Journal on Matrix Analysis and Applications, 2006. In press.

[Arsigny 06f] Vincent Arsigny, Pierre Fillard, Xavier Pennec & Nicholas Ayache. *Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors.* Magnetic Resonance in Medicine, vol. 56, no. 2, pages 411–421, August 2006.

[Arsigny 06g] Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *Bi-invariant Means in Lie Groups. Application to Left-invariant Polyaffine Transformations.* Research report RR-5885, INRIA Sophia-Antipolis, April 2006.

[Ashburner 99] J. Ashburner, J. Andersson & K. Friston. *High-dimensional image registration using symmetric priors.* NeuroImage, vol. 9, pages 619–628, 1999.

[Banyaga 97] Augustin Banyaga. The structure of classical diffeomorphism groups, volume 400 of *Mathematics and its applications.* Kluwer Academic Publishers, 1997.

[Bardinet 02] Eric Bardinet, Sébastien Ourselin, Grégoire Malandain, Dominique Tandé, Karine Parain, Nicholas Ayache & Jérôme Yelnik. *Three dimensional functional cartography of the human basal ganglia by registration of optical and histological serial sections.* In IEEE International Symposium on Biomedical Imaging, pages 329–332, Washington, USA, 2002.

[Basser 94] P. J. Basser, J. Mattiello & D. Le Bihan. *MR diffusion tensor spectroscopy and imaging.* Biophysical Journal, vol. 66, pages 259–267, 1994.

[Basser 98] P. J. Basser & C. Pierpaoli. *A Simplified method to measure the diffusion tensor from seven MR images.* Magnetique Resonance in Medecine, vol. 39, pages 928–934, 1998.

[Basser 00] P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda & A. Aldroubi. *In Vivo Fiber Tractography Using DT-MRI Data.* Magnetic Resonance in Medicine, vol. 44, pages 625–632, 2000.

[Basser 02] Peter J. Basser & Derek K. Jones. *Diffusion-tensor MRI: theory, experimental design and data analysis - a technical review.* NMR in Biomedicine, vol. 15, no. 7-8, pages 456–467, 2002.

[Basser 03] Peter J. Basser & Sinisa Pajevic. *A Normal Distribution for Tensor-Valued Random Variables: Applications to Diffusion Tensor MRI.* IEEE Transactions on Medical Imaging, vol. 22, no. 7, pages 785–794, July 2003.

[Batchelor 05] P. G. Batchelor, M. Moakher, D. Atkinson, F. Calamante & A. Connelly. *A Rigorous Framework for Diffusion Tensor Calculus.* Magnetic Resonance in Medicine, vol. 53, pages 221–225, 2005.

[Bazaraa 93] Mokhtar S. Bazaraa, Hanif D. Sherali & C .M. Shetty. Non linear programming, theory and algorithms, 2nd edition. John Wiley & Sons, Inc, 1993.

[Beg 03] Mirza Faisal Beg, Michael I. Miller, Alain Trouvé & Laurent Younes. *The Euler-Lagrange Equation for Interpolating Sequence of Landmark Datasets.* In MICCAI (2), pages 918–925, 2003.

[Beg 05] Mirza Faisal Beg, Michael I. Miller, Alain Trouvé & Laurent Younes. *Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms.* International Journal of Computer Vision, vol. 61, no. 2, pages 139–157, 2005.

[Billingsley 95] Patrick Billingsley. Probability and measure. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, third edition edition, 1995.

[Boisvert 06] Jonathan Boisvert, Xavier Pennec, Nicholas Ayache, Hubert Labelle & Farida Cheriet. *3D Anatomic Variability Assesment of the Scoliotic Spine Using Statistics on Lie Groups.* In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), pages 750–753, Crystal Gateway Marriott, Arlington, Virginia, USA, April 2006. IEEE.

[Bookstein 99] F. L. Bookstein. *Linear methods for nonlinear maps: Procrustes fits, thin-plate splines, and the biometric analysis of shape variability.* In A. Toga, editor, Brain Warping, pages 157–181. Academic Press, 1999.

[Bourbaki 89] Nicolas Bourbaki. Elements of mathematics: Lie groups and lie algebra, chapters 1-3. Springer-Verlag, 2nd printing edition, 1989.

[Broxand 03] T. Broxand, M. Roussonand, R. Deriche & J. Weickert. *Unsupervised segmentation incorporating colour, texture, and motion.* In Lecture Notes in Computer Science, volume 2756, pages 353–360. Springer, Berlin, 2003.

[Cachier 02] Pascal Cachier. *Recalage non-rigide d'images médicales volumiques, contributions aux approches iconiques et géométriques.* PhD thesis, École Centrale Paris, 2002.

[Cachier 03] Pascal Cachier, Eric Bardinet, Didier Dormont, Xavier Pennec & Nicholas Ayache. *Iconic Feature Based Nonrigid Registration: The PASHA Algorithm.* Comp. Vision and Image Understanding, vol. 89, no. 2-3, pages 272–298, Feb.-march 2003. Special Issue on Nonrigid Registration.

[Camion 01] Vincent Camion & Laurent Younes. *Geodesic Interpolating Splines.* In M. Figuiredo, J. Zerubia & A.K. Jain, editors, Proc. of Energy Minimization Methods in Comp. Vis. and Pat. Rec. (EMMCVPR;01), LNCS 2134, pages 513–527, 2001.

[Charpiat 05a] Guillaume Charpiat, Olivier D. Faugeras & Renaud Keriven. *Approximations of Shape Metrics and Application to Shape Warping and Empirical Shape Statistics.* Foundations of Computational Mathematics, vol. 5, no. 1, pages 1–58, 2005.

[Charpiat 05b] Guillaume Charpiat, Olivier D. Faugeras & Renaud Keriven. *Image Statistics Based on Diffeomorphic Matching.* In ICCV, pages 852–857. IEEE Computer Society, 2005.

[Chefd'hotel 02] C. Chefd'hotel, G Hermosillo & O. Faugeras. *Flows of diffeomorphisms for multimodal image registration.* In Proc. of IEEE Int. Symp. on Biomedical Ima., Washington D.C, july 8-11 2002.

[Chefd'hotel 04] C. Chefd'hotel, D. Tschumperlé, R. Deriche & O. Faugeras. *Regularizing Flows For Constrained Matrix-Valued Images.* Journal of Mathematical Imaging and Vision, vol. 20, no. 1-2, pages 147–162, 2004.

[Cheng 01] S. Hun Cheng, N. J. Higham, C. S. Kenney & A. J. Laub. *Approximating the Logarithm of a Matrix to Specified Accuracy.* SIAM J. Matrix Anal. Appl., vol. 22, no. 4, pages 1112–1125, 2001.

[Choi 00] Yongchoel Choi & Seungyong Lee. *Injectivity Conditions of 2D and 3D Uniform Cubic B-Spline Functions.* Graphical Models, vol. 62, no. 6, pages 411–427, 2000.

[Christensen 96] Gary E. Christensen, Richard D. Rabbitt & Michael I. Miller. *Deformable templates using large deformation kinematics.* IEEE Transactions on Image Processing, vol. 5, no. 10, pages 1435–1447, 1996.

[Clatz 05] Olivier Clatz, Hervé Delingette, Ion-Florin Talos, Alexandra J. Golby, Ron Kikinis, Ferenc Jolesz, Nicholas Ayache & Simon Warfield. *Robust Non-Rigid Registration to Capture Brain Shift from Intra-Operative MRI.* IEEE Transactions on Medical Imaging, vol. 24, no. 11, pages 1417–1427, Nov. 2005.

[Commowick 05] Olivier Commowick, Radu Stefanescu, Pierre Fillard, Vincent Arsigny, Nicholas Ayache, Xavier Pennec & Grégoire Malandain. *Incorporating Statistical Measures of Anatomical Variability in Atlas-to-Subject Registration for Conformal Brain Radiotherapy.* In J. Duncan & G. Gerig, editors, Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part II, volume 3750 of *LNCS*, pages 927–934, Palm Springs, CA, USA, October 26-29, 2005. Springer Verlag.

[Commowick 06a] Olivier Commowick, Vincent Arsigny, Jimena Costa, Nicholas Ayache & Grégoire Malandain. *An Efficient Locally Affine Framework for the Registration of Anatomical Structures.* In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), pages 478–481, Crystal Gateway Marriott, Arlington, Virginia, USA, April 2006.

[Commowick 06b] Olivier Commowick, Vincent Arsigny, Jimena Costa, Nicholas Ayache & Grégoire Malandain. *An Efficient Locally Affine Framework for the Registration of Anatomical Structures.* Medical Image Analysis, 2006. Submitted.

[Commowick 06c] Olivier Commowick & Grégoire Malandain. *Evaluation of Atlas Construction Strategies in the Context of Radiotherapy Planning.* In Proceedings of the SA2PM Workshop (From Statistical Atlases to Personalized Models), Copenhagen, October 2006. held in conjunction with MICCAI 2006.

[Coulon 04] O. Coulon, D.C. Alexander & S. Arridge. *Diffusion tensor magnetic resonance image regularization.* Medical Image Analysis, vol. 8, no. 1, pages 47–67, 2004.

[Culver 66] W. J. Culver. *On the existence and uniqueness of the real logarithm of a matrix.* Proceedings of the American Mathematical Society, vol. 17, no. 5, pages 1146–1151, October 1966.

[Cuzol 05] A. Cuzol, P. Hellier & E. Mémin. *A novel parametric method for non-rigid image registration.* In G. Christensen & M. Sonka, editors, Proc. of IPMI'05, no. 3565 in LNCS, pages 456–467, Glenwood Springes, Colorado, USA, July 2005.

[Davies 02] Rhodri H. Davies, Carole J. Twining, Tim F. Cootes, John C. Waterton & Chris J. Taylor. *3D Statistical Shape Models Using Direct Opimisation of Description Length.* ECCV 2002, LNCS 2352, Springer-Verlag, pages 3–20, 2002.

[Davis 04] Brad Davis, Peter Lorenzen & Sarang C. Joshi. *Large Deformation Minimum Mean Squared Error Template Estimation for Computational Anatomy.* In ISBI, pages 173–176. IEEE, 2004.

[do Carmo 92] Manfredo P. do Carmo. Riemannian geometry. Birkhäuser, Boston, MA, 1992.

[du Bois d'Aische 05] A. du Bois d'Aische, M. de Craene, B. Macq & S.K. Warfield. *An articulated registration method.* In Proc. of the IEEE International Conference on Image Processing (ICIP 2005), volume I, pages 21–24, 2005.

[Feddern 04] C. Feddern, J. Weickert, B. Burgeth & M. Welk. *Curvature-driven PDE methods for matrix-valued images.* Technical report 104, Department of Mathematics, Saarland University, Saarbrücken, Germany, April 2004.

[Ferrant 99] M. Ferrant, S.K. Warfield, C.R.G. Guttmann, R.V. Mulkern, F.A. Jolesz & R. Kikinis. *3D Image Matching Using a Finite Element Based Elastic Deformation Model.* In Proc. of MICCAI'99, LNCS 1679, pages 202–209, 1999.

[Fillard 03] Pierre Fillard, John Gilmore, Joseph Piven, Weili Lin & Guido Gerig. *Quantitative Analysis of White Matter Fiber Properties along Geodesic Paths.* volume 2879 of *LNCS*, pages 16–23. Springer, November 2003.

[Fillard 05a] Pierre Fillard, Vincent Arsigny, Nicholas Ayache & Xavier Pennec. *A Riemannian Framework for the Processing of Tensor-Valued Images.* In Ole Fogh Olsen, Luc Florak & Arjan Kuijper, editors, Deep Structure, Singularities, and Computer Vision (DSSCV), no. 3753 in LNCS, pages 112–123. Springer Verlag, June 2005.

[Fillard 05b] Pierre Fillard, Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *Joint Estimation and Smoothing of Clinical DT-MRI with a Log-Euclidean Metric.* Research Report RR-5607, INRIA, Sophia-Antipolis, France, June 2005.

[Fillard 05c] Pierre Fillard, Vincent Arsigny, Xavier Pennec, Paul M. Thompson & Nicholas Ayache. *Extrapolation of Sparse Tensor Fields: Application to the Modeling of Brain Variability.* In Gary Christensen & Milan Sonka, editors, Proc. of Information Processing in Medical Imaging 2005 (IPMI'05), volume 3565 of *LNCS*, pages 27–38, Glenwood springs, Colorado, USA, July 2005. Springer.

[Fillard 06a] Pierre Fillard, Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *Clinical DT-MRI estimation, smoothing and fiber tracking with log-Euclidean metrics.* In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), LNCS, pages 786–789, Crystal Gateway Marriott, Arlington, Virginia, USA, April 2006.

[Fillard 06b] Pierre Fillard, Vincent Arsigny, Xavier Pennec & Nicholas Ayache. *Clinical DT-MRI Estimation, Smoothing and Fiber Tracking with Log-Euclidean Metrics.* NeuroImage, 2006. Submitted.

[Fillard 06c] Pierre Fillard, Vincent Arsigny, Xavier Pennec, Kiralee M. Hayashi, Paul M. Thompson & Nicholas Ayache. *Measuring Brain Variability by Extrapolating Sparse Tensor Fields Measured on Sulcal Lines*. Research Report 5887, INRIA, April 2006.

[Fillard 06d] Pierre Fillard, Vincent Arsigny, Xavier Pennec, Kiralee M. Hayashi, Paul M. Thompson & Nicholas Ayache. *Measuring Brain Variability by Extrapolating Sparse Tensor Fields Measured on Sulcal Lines*. Neuroimage, vol. 34, no. 2, pages 639–650, January 2006. Also as INRIA Research Report 5887, April 2006.

[Fletcher 04a] P. T. Fletcher & S. C. Joshi. *Principal Geodesic Analysisee on Symmetric Spaces: Statistics of Diffusion Tensors*. In Proc. of CVAMIA and MMBIA Workshops, Prague, Czech Republic, May 15, 2004, LNCS 3117, pages 87–98. Springer, 2004.

[Fletcher 04b] P. Thomas Fletcher, Conglin Lu, Stephen M. Pizer & Sarang C. Joshi. *Principal geodesic analysis for the study of nonlinear statistics of shape*. IEEE Trans. Med. Imaging, vol. 23, no. 8, pages 995–1005, 2004.

[Gallot 93] Sylvestre Gallot, Dominique Hulin & Jacques Lafontaine. Riemannian geometry. Springer-Verlag, 2nd edition edition, 1993.

[Glatard 06] Tristan Glatard, Xavier Pennec & Johan Montagnat. *Performance evaluation of grid-enabled registration algorithms using bronze-standards*. In Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), Part II, no. 4191 in LNCS, pages 152–160. Springer, 2-4 October 2006.

[Glaunes 04] Joan Glaunes, Alain Trouvé & Laurent Younes. *Diffeomorphic Matching of Distributions: A New Approach for Unlabelled Point-Sets and Sub-Manifolds Matching*. In CVPR (2), pages 712–718, 2004.

[Glöckner 06] Helge Glöckner. *Fundamental problems in the theory of infinite-dimensional Lie groups*, 2006.

[Godement 82] Roger Godement. Introduction à la théorie des groupes de Lie. Publications Mathématiques de l'Université Paris VII, 1982.

[Goodlett 06] Casey Goodlett, Brad Davis, Remi Jean, John Gilmore & Guido Gerig. *Improved Correspondence for DTI Population Studies via Unbiased Atlas Building*. In Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), LNCS, 2-4 October 2006. To appear.

[Grenander 06] Ulf Grenander, Anuj Srivastava & Sanjay Saini. *Characterization of Biological Growth Using Iterated Diffeomorphisms*. In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), Crystal Gateway Marriott, Arlington, Virginia, USA, April 2006.

[Guo 04] Hongyu Guo, Anand Rangarajan, S. Joshi & Laurent Younes. *Non-Rigid Registration of Shapes Via Diffeomorphic Point Matching*. In ISBI, pages 924–927, 2004.

[Hall 03] B. C. Hall. Lie groups, lie algebras, and representations: an elementary introduction, volume 222 of *Graduate Texts in Mathematics*. Springer Verlag, 2003.

[Haselgrove 96] J. C. Haselgrove & J. R. Moore. *Correction of distortion of echo-planar images used to calculate the apparent diffusion coefficient.* Magnetic Resonance in Medecine, vol. 36, pages 960–964, 1996.

[Hellier 01] Pierre Hellier, Christian Barillot, Étienne Mémin & P. Perex. *Hierarchical Estimation of a Dense Deformation Field for 3D Robust Registration.* IEEE Trans. Med. Imaging, vol. 20, no. 5, pages 388–402, 2001.

[Hermosillo 02] G. Hermosillo, C. Chefd'Hotel & O.D. Faugeras. *Variational Methods for Multimodal Image Matching.* IJCV, vol. 50, no. 3, pages 329–343, December 2002.

[Higham 89] N. J. Higham. *Matrix Nearness Problems and Applications.* In M. J. C. Gover & S. Barnett, editors, Applications of Matrix Theory, pages 1–27. Oxford University Press, Oxford, 1989.

[Higham 05] N. J. Higham. *The Scaling and Squaring Method for the Matrix Exponential Revisited.* SIAM J. Matrix Anal. Appl., vol. 26, no. 4, pages 1179–1193, 2005.

[Hill 01] Derek L G Hill, Philipp G Batchelor, Mark Holden & David J Hawkes. *Medical image registration.* Phys. Med. Biol., vol. 46, pages R1–R45, 2001.

[Holden 00] M. Holden, D. L. G. Hill, E. R. E. Denton, J. M. Jarosz, T. C. S. Cox, T. Rohlfing, J. Goodey & London; Hawkes D. J. King's Coll. *Voxel similarity measures for 3-D serial MR brain image registration.* IEEE Trans. Med. Imag., vol. 19, no. 2, pages 94–102, 2000.

[Holm 04a] D. D. Holm & J. E. Marsden. *Momentum maps and measure valued solutions of the Euler-Poincaré equations for the diffeomorphism group.* Prog. Math., vol. 232, pages 203–235, 2004.

[Holm 04b] Darryl D. Holm, J. Tilak Ratnanather, Alain Trouve & Laurent Younes. *Soliton Dynamics in Computational Anatomy.* NeuroImage, vol. 23, page S170, 2004.

[Jones 02] Derek K. Jones, Lewis D. Griffin, Daniel C. Alexander, Marco Catani, Mark A. Horsfield, Robert Howard & Steve C. R. Williams. *Spatial Normalization and Averaging of Diffusion Tensor MRI Data Sets.* NeuroImage, vol. 17, pages 592–617, 2002.

[Joshi 00] Sarang C. Joshi & Michael I. Miller. *Landmark matching via large deformation diffeomorphisms.* IEEE Transactions on Image Processing, vol. 9, no. 8, pages 1357–1370, 2000.

[Joshi 05] Shantanu H. Joshi, Anuj Srivastava & Washington Mio. *Elastic Shape Models for Interpolations of Curves in Image Sequences.* In IPMI, pages 541–552, 2005.

[Kendall 84] D. G. Kendall. *Shape manifolds, Procrustean metrics and complex projective spaces.* Bull. London Math. Soc., vol. 16, pages 81–121, 1984.

[Kendall 89] David G. Kendall. *A Survey a of the Statistical Theory of Shape.* Statistical Science, vol. 4, no. 2, pages 87–120, 1989.

[Klassen 03] Eric Klassen, Anuj Srivastava & Washington Mio. *Analysis of Planar Shapes Using Geodesic Paths on Shape Spaces.* 2003.

[Klassen 06] E. Klassen & A. Srivastava. *Geodesic Between 3D Closed Curves Using Path Straightening.* In Proc. of ECCV'2006, 2006. To appear.

[Lambert 91] J. D. Lambert. Numerical methods for ordinary differential systems: the initial value problem. John Wiley & Sons, Inc., New York, NY, USA, 1991.

[Lang 04] S. Lang. Algebra. Graduate Texts in Mathematics. Springer, 3rd rev. ed. 2002. corr. 4th printing edition, 2004.

[Le Bihan 91] Denis Le Bihan. *Diffusion MNR imaging.* Magn Reson Q, vol. 7, pages 1–30, 1991.

[Le Bihan 01] Denis Le Bihan, Jean-Francois Mangin, Cyril Poupon, Chris A. Clark, Sabina Pappata, Nicolas Molko & Hughes Chabriat. *Diffusion Tensor Imaging: Concepts and Applications.* Journal of Magnetic Resonance Imaging, vol. 13, pages 534–546, 2001.

[Le 93] H. Le & D. G. Kendall. *The Riemannian structure of Euclidean shape spaces: a novel environment for statistics.* The Annals of Statistics, vol. 21, no. 3, pages 1225–1271, 1993.

[Lenglet 04] C. Lenglet, R. Deriche & O. Faugeras. *Inferring White Matter Geometry from Diffusion Tensor MRI: Application to Connectivity Mapping.* In T. Pajdla & J. Matas, editors, Proc. of the 8th European Conference on Computer Vision, LNCS, pages 127–140. Springer, May 11-14 2004.

[Lenglet 06a] C. Lenglet, M. Rousson, R. Deriche & O. Faugeras. *Statistics on the Manifold of Multivariate Normal Distributions: Theory and Application to Diffusion Tensor MRI Processing.* Journal of Mathematical Imaging and Vision, 2006. In press.

[Lenglet 06b] Christophe Lenglet. *Geometric and Variational Methods for Diffusion Tensor MRI Processing.* Thèse de sciences (PhD thesis), Université de Nice Sophia-Antipolis, Nice (France), December 2006.

[Lepore 06] Natasha Lepore, Caroline A. Brun, Ming-Chang Chiang, Yi-Yu Chou, Oscar L. Lopez, Howard J. Aizenstein, Arthur W. Toga, James T. Becker & Paul M. Thompson. *Multivariate Statistics of the Jacobian Matrices in Tensor Based Morphometry and their application to HIV/AIDS.* In Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), LNCS, 2-4 October 2006. To appear.

[Little 96] J. A. Little, D. L. G. Hill & D. J. Hawkes. *Deformations incorpotationg rigid structures.* Computer Vision and Image Understanding, vol. 66, no. 2, pages 223–232, May 1996.

[Lorenzen 05] Peter Lorenzen, Brad Davis & Sarang C. Joshi. *Unbiased Atlas Formation Via Large Deformations Metric Mapping.* In James S. Duncan & Guido Gerig, editors, MICCAI (2), volume 3750 of *Lecture Notes in Computer Science*, pages 411–418. Springer, 2005.

[Maes 97] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal & P. Suetens. *Multimodality image registration by maximization of mutualinformation.* IEEE Trans. Med. Imag., vol. 16, no. 2, pages 187–198, 1997.

[Maintz 98] J.B.A. Maintz & M.A Viergever. *A survey of medical registration.* Medical image analysis, vol. 2, no. 1, pages 1–36, 1998.

[Marsland 03] Stephen Marsland, Carole J. Twining & Christopher J. Taylor. *Groupwise Non-rigid Registration Using Polyharmonic Clamped-Plate Splines.* In MICCAI (2), pages 771–779, 2003.

[Marsland 04] Stephen Marsland & Carole J. Twining. *Constructing diffeomorphic representations for the groupwise analysis of nonrigid registrations of medical images.* IEEE Trans. Med. Imaging, vol. 23, no. 8, pages 1006–1020, 2004.

[Michor 06] Peter W. Michor & David Mumford. *Riemannian geometries on spaces of plane curves.* Jour. Eur. Math. Soc., vol. 8, pages 1–48, 2006.

[Miller 03] Michael I. Miller, Alain Trouvé & Laurent Younes. *The metric spaces, Euler equations, and normal geodesic image motions of computational anatomy.* In ICIP (2), pages 635–638, 2003.

[Miller 06] M. I. Miller, A. Trouvé & L. Younes. *Geodesic Shooting for Computational Anatomy.* Journal of Mathematical Imaging and Vision, 2006. In press.

[Milnor 83] J. Milnor. *Remarks on infinite dimensional Lie groups in Relativity, groups and Topology II.* In Proc. of the Summer School on Quantum Gravity, Les Houches, 1983. North Holland, Amsterdam-New-York (1984).

[Moakher 02] Maher Moakher. *Means and Averaging in the Group of Rotations.* SIAM J. Matrix Anal. Appl., vol. 24, no. 1, pages 1–16, 2002.

[Moakher 05] M. Moakher. *A differential geometry approach to the geometric mean of symmetric positive-definite matrices.* SIAM Journal on Matrix Analysis and Applications, vol. 26, pages 735–747, 2005.

[Mohammadi 97] B. Mohammadi, H. Borouchaki & P.L.George. *Delaunay Mesh Generation Governed by Metric Specifications. Part II: Applications.* Finite Element in Analysis and Design, no. 25, pages 85–109, 1997.

[Moler 78] Cleve Moler & Charles Van Loan. *Nineteen dubious ways to compute the exponential of a matrix.* SIAM jour. of Matr. Anal. and Appl., vol. 20, no. 4, pages 801–836, October 1978.

[Mori 02] S. Mori, W. E. Kaufmann, C. Davatzikos, B. Stieltjes, L. Amodei, K. Fredericksen, G. D. Pearlson, E. R. Mehlem, M. Solaiyappan, G. V. Raymond, H. W. Moser & P. C. van Zijl. *Imaging cortical association tracts in the human brain using diffusion-tensor-based axonal tracking.* Magnetique Resonance in Medecine, vol. 47, pages 215–223, 2002.

[Narayanan 05] R. Narayanan, J. A. Fessler, H. Park & C. R. Meyer. *Diffeomorphic Nonlinear Transformations: A Local Parametric Approach for Image Registration.* In Proceedings of IPMI'05, volume 3565 of LNCS, pages 174–185, 2005.

[Nicolau 03] Stéphane Nicolau, Xavier Pennec, Luc Soler & Nicholas Ayache. *Evaluation of a New 3D/2D Registration Criterion for Liver Radio-Frequencies Guided by Augmented Reality.* In N. Ayache & H. Delingette, editors, International Symposium on Surgery Simulation and Soft Tissue Modeling (IS4TM'03), volume 2673 of *Lecture Notes in Computer Science*, pages 270–283, Juan-les-Pins, France, 2003. INRIA Sophia Antipolis, Springer-Verlag.

[Noblet 05] Vincent Noblet, Christian Heinrich, Fabrice Heitz & Jean-Paul Armspach. *3-D deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization.* IEEE Transactions on Image Processing, vol. 14, no. 5, pages 553–566, 2005.

[Norris 06] A. N. Norris. *The isotropic material closest to a given anisotropic material.* J. Mech. Materials Struct., vol. 1, no. 2, pages 231–246, 2006.

[Ourselin 00] S. Ourselin, A. Roche, S. Prima & N. Ayache. *Block Matching: A General Framework to Improve Robustness of Rigid Registration of Medical Images.* In A.M. DiGioia & S. Delp, editors, Third International Conference on Medical Robotics, Imaging And Computer Assisted Surgery (MICCAI 2000), volume 1935 of *Lectures Notes in Computer Science*, pages 557–566, Pittsburgh, Penn, USA, octobre 11-14 2000. Springer.

[Ourselin 01a] S. Ourselin, E. Bardinet, D. Dormont, G. Malandain, A. Roche, N. Ayache, D. Tande, K. Parain & J. Yelnik. *Fusion of histological sections and MR images: towards the construction of an atlas of the human basal ganglia.* In W.J. Niessen & M.A. Viergever, editors, 4th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'01), volume 2208 of *LNCS*, pages 743–751, Utrecht, The Netherlands, October 2001.

[Ourselin 01b] Sébastien Ourselin, Alexis Roche, Gérard Subsol, Xavier Pennec & Nicholas Ayache. *Reconstructing a 3D Structure from Serial Histological Sections.* Image and Vision Computing, vol. 19, no. 1-2, pages 25–31, January 2001.

[Özarslan 03] E. Özarslan & T. H. Mareci. *Generalized Diffusion Tensor Imaging and Analytical Relationships Between Diffusion Tensor Imaging and High Angular Resolution Diffusion Imaging.* Magnetique Resonance in Medecine, vol. 50, pages 955–965, 2003.

[Papademetris 05] X. Papademetris, D. P. Dione, L. W. Dobrucki, L. H. Staib & A. J. Sinusas. *Articulated Rigid Registration for Serial Lower-Limb Mouse Imaging.* In MICCAI'05 (2), pages 919–926, 2005.

[Park 03] Hyunjin Park & Charles R. Meyer. *Grid Refinement in Adaptive Non-rigid Registration.* In MICCAI (2), pages 796–803, 2003.

[Paulsen 04] R. R. Paulsen. *Statistical Shape Analysis of the Human Ear Canal with Application to In-the-Ear Hearing Aid Design.* PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004. Supervised by Rasmus Larsen (IMM), Søren Laugesen (Oticon), Herve Delingette (INRIA), and Knut Conradsen (IMM).

[Pennec 96] Xavier Pennec. *L'incertitude dans les problèmes de reconnaissance et de recalage – Applications en imagerie médicale et biologie moléculaire.* Thèse de sciences (phd thesis), Ecole Polytechnique, Palaiseau (France), December 1996.

[Pennec 98a] Xavier Pennec. *Computing the mean of geometric features - Application to the mean rotation.* Research Report RR-3371, INRIA, March 1998.

[Pennec 98b] Xavier Pennec & Nicholas Ayache. *Uniform distribution, distance and expectation problems for geometric features processing.* Journal of Mathematical Imaging and Vision, vol. 9, no. 1, pages 49–67, July 1998. A preliminary version appeared as INRIA Research Report 2820, March 1996.

[Pennec 99] Xavier Pennec. *Probabilities and Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements.* In A.E. Cetin, L. Akarun, A. Ertuzun, M.N. Gurcan & Y. Yardimci, editors, Proc. of Nonlinear Signal and Image Processing (NSIP'99), volume 1, pages 194–198, June 20-23, Antalya, Turkey, 1999. IEEE-EURASIP.

[Pennec 05] Xavier Pennec, Radu Stefanescu, Vincent Arsigny, Pierre Fillard & Nicholas Ayache. *Riemannian Elasticity: A statistical regularization framework for non-linear registration.* In J. Duncan & G. Gerig, editors, Proceedings of the 8th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, Part II, volume 3750 of *LNCS*, pages 943–950, Palm Springs, CA, USA, October 26-29, 2005. Springer Verlag.

[Pennec 06a] Xavier Pennec. *Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements.* Journal of Mathematical Imaging and Vision, vol. 25, no. 1, pages 127–154, July 2006. A preliminary appeared as INRIA RR-5093, January 2004.

[Pennec 06b] Xavier Pennec, Pierre Fillard & Nicholas Ayache. *A Riemannian Framework for Tensor Computing.* International Journal of Computer Vision, vol. 66, no. 1, pages 41–66, January 2006. A preliminary version appeared as INRIA Research Report 5255, July 2004.

[Peter 06] Adrian Peter & Anand Rangarajan. *Shape Analysis Using the Fisher-Rao Riemannian Metric: Unifying Shape Representation and Deformation*. In Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), Crystal Gateway Marriott, Arlington, Virginia, USA, April 2006.

[Peyrat 06a] Jean-Marc Peyrat, Maxime Sermesant, Hervé Delingette, Xavier Pennec, Chenyang Xu, Elliot McVeigh & Nicholas Ayache. *Towards a Statistical Atlas of Cardiac Fiber Architecture*. Research Report 5906, INRIA, May 2006.

[Peyrat 06b] Jean-Marc Peyrat, Maxime Sermesant, Hervé Delingette, Xavier Pennec, Chenyang Xu, Elliot McVeigh & Nicholas Ayache. *Towards a Statistical Atlas of Cardiac Fiber Structure*. In Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), Part I, no. 4190 in LNCS, pages 297–304, 2-4 October 2006.

[Pitiot 03] A. Pitiot, G. Malandain, E. Bardinet & P. M. Thompson. *Piecewise Affine Registration of Biological Images*. In J.C. Gee, J.B. A. Maintz & M. W. Vannier, editors, Second International Workshop on Biomedical Image Registration WBIR'03, volume 2717 of *Lecture Notes in Computer Science*, pages 91–101, Philadelphia, PA, USA, 2003. Springer-Verlag. Also research report INRIA RR-4866.

[Pitiot 06] Alain Pitiot, Eric Bardinet, Paul M. Thompson & Grégoire Malandain. *Piecewise Affine Registration of Biological Images for Volume Reconstruction*. Medical Image Analysis, vol. 10, no. 3, pages 465–483, June 2006.

[Pizer 03] Stephen M. Pizer, P. Thomas Fletcher, Andrew Thall, Martin Styner, Guido Gerig & Sarang C. Joshi. *Object models in multiscale intrinsic coordinates via m-reps*. Image Vision Comput., vol. 21, no. 1, pages 5–15, 2003.

[Poupon 00] C. Poupon, C. A. Clark, V. Frouin, J. Regis, I. Bloch, D. LeBihan & J.-F. Mangin. *Regularization of diffusion-based direction maps for the tracking of brain white matter fascicles*. Neuroimage, vol. 12, no. 2, pages 184–95, August 2000.

[Rivière 00] D. Rivière. *Apprentissage de la variabilité inter-individuelle de l'anatomie corticale cérébrale pour la reconnaissance automatique des sillons*. thèse de doctorat, Univ. d'Évry Val d'Essonne, Évry, sep. 2000.

[Rivière 02] D. Rivière, J.-F. Mangin, D. Papadopoulos-Orfanos, J.-M. Martinez, V. Frouin & J. Régis. *Automatic recognition of cortical sulci of the Human Brain using a congregation of neural networks*. Medical Image Analysis, vol. 6, no. 2, pages 77–92, 2002.

[Rohde 03] Gustavo K. Rohde, Akram Aldroubi & Benoit M. Dawant. *The Adaptive Bases Algorithm for Intensity Based Nonrigid Image Registration*. IEEE Trans. Med. Imaging, vol. 22, no. 11, pages 1470–1479, 2003.

[Rueckert 99] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach & D. J. Hawkes. *Non-rigid registration using free-form deformations: Application to breast MR images.* IEEE Trans. Medecal Imaging, vol. 18, no. 8, pages 712–721, 1999.

[Rueckert 03] D. Rueckert, A. F. Frangi & J. A. Schnabel. *Automatic Construction of 3D Statistical Deformation Models of the Brain using Non-Rigid Registration.* IEEE TMI, vol. 22, no. 8, pages 1014–1025, 2003.

[Salencon 01] J. Salencon. Handbook of continuum mechanics. Springer Verlag, Berlin, 2001.

[Samelson 63] H. Samelson. *On the Brouwer Fixed Point Theorem.* Portugal. Math., vol. 22, pages 264–268, 1963.

[Sattinger 86] D.H. Sattinger & O.L. Weaver. Lie groups and algebras with applications to physics, geometry, and mechanics, volume 61 of *AMS*. Springer Verlag, 1986.

[Schnabel 01] Julia A. Schnabel, Daniel Rueckert, Marcel Quist, Jane M. Blackall, Andy D. Castellano-Smith, Thomas Hartkens, Graeme P. Penney, Walter A. Hall, Haiying Liu, Charles L. Truwit, Frans A. Gerritsen, Derek L. G. Hill & David J. Hawkes. *A Generic Framework for Non-rigid Registration Based on Non-uniform Multi-level Free-Form Deformations.* In MICCAI, pages 573–581, 2001.

[Schwartz 97] L. Schwartz. Analyse tome 2 : Calcul differentiel. Hermann, octobre 1997.

[Sheppard 68] D. Sheppard. *A two-dimensionnal interpolation function for irregularly spaced data.* In 23rd National Conference of the ACM, pages 517–524. ACM Press, 1968.

[Small 96] Christopher G. Small. The statistical theory of shape. Springer, 1996.

[Stefanescu 04] Radu Stefanescu, Xavier Pennec & Nicholas Ayache. *Grid Powered Nonlinear Image Registration with Locally Adaptive Regularization.* Medical Image Analysis, vol. 8, no. 3, pages 325–342, September 2004. MICCAI 2003 Special Issue.

[Stefanescu 05] Radu Stefanescu. *Parallel nonlinear registration of medical images with a priori information on anatomy and pathology.* Thèse de sciences, Université de Nice – Sophia-Antipolis, March 2005.

[Sternberg 64] S. Sternberg. Lectures on differential geometry. Prentice Hall Mathematics Series. Prentice Hall Inc., 1964.

[Tarantola 06] Albert Tarantola. Elements for physics - quantities, qualities, and instrinsic theories. Springer Verlag, 2006.

[Tenenbaum 85] M. Tenenbaum & H. Pollard. Ordinary differential equations. Dover, 1985.

[Terriberry 05] Timothy B. Terriberry, Sarang C. Joshi & Guido Gerig. *Hypothesis Testing with Nonlinear Shape Models.* In IPMI, pages 15–26, 2005.

[Thirion 98] J.-P. Thirion. *Image matching as a diffusion process: an analogy with Maxwell's demons.* Medical Image Analysis, vol. 2, no. 3, pages 243–260, 1998.

[Trouvé 95] Alain Trouvé. *An Approach of Pattern Recognition through infinite dimensional group.* Research Report 95-9, Laboratoire de Mathématiques de l'École Normale Supérieure (LMENS), May 1995.

[Trouvé 98] Alain Trouvé. *Diffeomorphisms Groups and Pattern Matching in Image Analysis.* International Journal of Computer Vision, vol. 28, no. 3, pages 213–221, 1998.

[Trouvé 00] A. Trouvé & L. Younes. *Diffeomorphic Matching Problems in One Dimension: Designing and Minimizing Matching Functionals.* In ECCV00, pages I: 573–587, 2000.

[Trouvé 05a] A. Trouvé & L. Younes. *Local Geometry of Deformable Templates.* SIAM Journal on Mathematical Analysis (SIMA), vol. 37, no. 1, pages 17–59, 2005.

[Trouvé 05b] Alain Trouvé & Laurent Younes. *Metamorphoses Through Lie Group Action.* Foundations of Computational Mathematics, vol. 5, no. 2, pages 173–198, 2005.

[Tschumperlé 01] D. Tschumperlé & R. Deriche. *Diffusion Tensor Regularization with Constraints Preservation.* In Conference on Computer Vision and Pattern Recognition (CVPR), volume I, pages 948–953, Kauai, Hawaii, December 2001.

[Tschumperlé 05] David Tschumperlé & Rachid Deriche. *Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 4, pages 506–517, April 2005.

[Tuch 04] D. S. Tuch. *Q-Ball imaging.* Magnetic Resonance in Medecine, vol. 52, pages 1358–1372, 2004.

[Vaillant 04] M. Vaillant, M.I. Miller, L. Younes & A. Trouvé. *Statistics on diffeomorphisms via tangent space representations.* NeuroImage, vol. 23, pages S161–S169, 2004.

[Vemuri 01] B. C. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang & T. Mareci. *Fiber Tract Mapping from Diffusion Tensor MRI.* In Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01), pages 81–88. IEEE, 2001.

[Vercauteren 06] Tom Vercauteren, Aymeric Perchant, Grégoire Malandain, Xavier Pennec & Nicholas Ayache. *Robust Mosaicing with Correction of Motion Distortions and Tissue Deformation for In Vivo Fibered Microscopy.* Medical Image Analysis, vol. 10, no. 5, pages 673–692, October 2006. Annual Medical Image Analysis (MedIA) Best Paper Award 2006.

[Wang 04] Z. Wang, B. C. Vemuri, Y. Chen & T. H. Mareci. *A Constrained Variational Principle for Simultaneous Smoothing and Estimation of the Diffusion Tensors from Complex DWI data.* IEEE TMI, vol. 23, no. 8, pages 930–939, 2004.

[Westin 02] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz & R. Kikinis. *Processing and Visualization of Diffusion Tensor MRI.* Medical Image Analysis, vol. 6, pages 93–108, 2002.

[Woods 03] Roger P. Woods. *Characterizing volume and surface deformations in an atlas framework: theory, applications, and implementation.* Neuroimage, vol. 18, no. 3, pages 769–88, 2003.

[Wüstner 03] Michael Wüstner. *A connected Lie group equals the square of the exponential image.* Journal of Lie Theory, vol. 13, pages 307–309, 2003.

[Xu 06] Shun Xu, Martin Styner, Brad Davis, Sarang Joshi & Guido Gerig. *Group Mean Differences of Voxel and Surface Objects via Nonlinear Group Averaging.* In Proc. of the Third IEEE Intern. Symp. on Biomed. Imaging (ISBI'2006), pages 758–761, 2006.

[Zefran 99] Milos Zefran, Vijay Kumar & Croke Christopher. *Metrics and Connections for Rigid-Body Kinematics.* International Journal of Robotics Research,, vol. 18, no. 2, pages 243–258, 1999.

[Zhang 04] Hui Zhang, Paul A. Yushkevich & James C. Gee. *Towards Diffusion Profile Image Registration.* In ISBI, pages 324–327, 2004.