Registration of Uncertain Geometric Features: Estimating the Pose and its Accuracy

Xavier Pennec
INRIA - Projet Epidaure
MIT - Artificial intelligence Lab.
NE43-741, 545 Technology square,
Cambridge MA 02-139, USA
Email: xpennec@ai.mit.edu

Abstract

We provide in this article a generic framework for the pose estimation from geometric features. We propose more particularly three algorithms: a gradient descent on the Riemannian least square distance and on the Mahalanobis distance, and an Extended Kalman Filter. For all these methods, we provide a way to compute the uncertainty of the resulting transformation. The analysis and comparison of the algorithms show how to combine them in a powerful and high level statistical registration algorithm. An application in medical image analysis demonstrate the usefulness of the method.

1 Introduction

In computer vision, most registration algorithms rely on feature extracted from the images. In this framework, the problem can generally be separated into two steps: (1) finding the correspondences between features (matches) and (2) computing the geometric transformation that maps one set of features to the other. In this article, we do not discuss matching methods per se, but the estimation of the geometric transformation and its accuracy.

We first review the existing methods for computing the 3D rigid motion from two sets of matched points and how one can estimate its uncertainty. However, models of the real world often lead to consider more complex features like lines [Gri90], planes, oriented points [FA96] or frames [PA94] and no traditional method easily extends to such features. Moreover, we have shown in [PA98] that geometric features generally do not belong to a vector space but rather to a manifold and that it induces paradoxes if we try to use the standard techniques on points with them. Thus, we have developed in [Pen96] a rigorous theory of uncertainty on geometric features. The basic idea is that our measures on features should be invariant by the action of a given transformation group. In the geometric framework, this can be ensured by using an invariant Riemannian distance on the manifold (which means that the transformation group is a kind of "rigid" one for the features). We have developed the existence conditions for such an invariant metric, a coherent definition of the expectation and of the covariance matrix of random features. Then, other statistical operations can be defined, such as the Mahalanobis distance and the χ^2 test.

Using these basic operations, we develop in the second section three generic pose estimation algorithms. The first is the generalization of the Least Squares and corresponds to the minimization of the squared Riemannian distance between matched features. The minimum is obtained by gradient descent. The two others minimize the Mahalanobis distance (in the rigorous sense of above) either by gradient descent or Kalman filtering. For the three methods, we provide a way to compute an estimation of the result accuracy.

In the third section, we analyze and compare the three algorithms in terms of accuracy of the estimated transformation, accuracy of its uncertainty estimation and of computation time. This leads to a useful classification for the use of these algorithms and a way to combine them in order to build a general but practical pose estimation algorithm including the estimation of the noise on features and an outliers rejection stage for robustness.

In the last section, we apply this theory to a practical case in Medical Image Analysis: the estimation of the relative motion of bones in the pelvis from MRI images. In this study, the estimation of the pose uncertainty turns out to be of the highest importance since it prevents a possibly wrong conclusion.

1.1 Pose estimation from matched points

In order to introduce the pose estimation problem, we assume here that our objects of interest are represented by two sets of n matched points $\{x_i\}$ and $\{y_i\}$ in $\mathbb{R}^3$. We call them respectively the model and scene points. Thus, if the transformation between these two objects is rigid and if we have exact measurements, we have $y_i = R.x_i + t$. We could then determine the movement parameters with only 3 couples of matched points: this gives 9 scalar equations for 6 unknown (3 for the translation and 3 for the rotation) and 3 invariants (the distance between points in each triplet).

Unfortunately, we never have exact data and we have to cope with measurement errors. Because of the 3 invariants the system is over-constrained. The usual solution is to look for the parameters of the rotation R and the translation t that minimize the sum of square errors:

$$C(R, t) = \frac{1}{2} \sum_i \|y_i - R.x_i - t\|^2 \qquad (1)$$

X. Pennec, *Registration of Uncertain Geometric Features: Estimating the Pose and its Accuracy*, **In Proc. of the First Image Registration Workshop, Greenbelt, USA, J. Le Moigne ed, pages 263-272. CESDIS, November 20-21 1997.**

This problem is called the *orthogonal Procrustes problem* in statistics, the *absolute orientation problem* in photogrammetry and the *pose estimation problem* in computer vision. Several closed form solutions have been developed, using unit quaternions [Hor87, Aya91, Fau93], the singular value decomposition (SVD) [Sch66, AHB87, Ume91], the Polar decomposition [HHN88] or dual quaternions [WS91]. In our implementation, we use the unit quaternion method.

Since all these methods compute the global minimum of the same criterion, the solutions are obviously mathematically identical. A study of the numerical point of view [LEF95] concludes that the difference in accuracy between the four methods is not significant (compared to the numerical precision of the machine) and that the computation times are comparable (a maximum difference of 15% which can be due to variations in the optimization of the implementation). The only (very slight) difference is the sensibility of the methods when the set of points becomes close to a degenerated configuration (aligned points).

The second problem is that the rigid transformation we have computed is of course not the exact transformation between the exact points. It is important and even vital in some medical applications to estimate the accuracy of our estimation. If uncertainty handling is a central topic in several works, like [DW88a, DW88b] or [Aya91, ZF92], there are much fewer studies dealing with the accuracy of the estimated motion. Early experimental works can be found in [FH84], [Sny89], or [HJL$^+$89]. A theoretical evaluation of the errors on rotation estimation is introduced in [Kan93, Kan94]. We have described in [PT97] how to estimate the uncertainty on this least squares pose estimation. We just recall here the results. Let r be the rotation vector and $R(\mathrm{r})$ the corresponding rotation matrix: the action of the rotation vector on a point x is $\mathrm{r} \star x = R(\mathrm{r})$ and the action of a rigid transformation $\mathrm{f} = (\mathrm{r}, t)$ is $\mathrm{f} \star x = R(\mathrm{r}).x + t$. In the case of an isotropic, identical and independent noise with standard deviation σ_x on model points and σ_y on scene points (the least squares estimator is optimal only at this condition), the covariance on the estimated rigid transformation $\hat{\mathrm{f}} = (\hat{\mathrm{r}}, \hat{t})$ is the inverse of the criterion Hessian matrix scaled by the variance of the residuals $\sigma_z^2 = \sigma_x^2 + \sigma_y^2$:

$$\Sigma_{\hat{\mathrm{f}}\hat{\mathrm{f}}} = \sigma_z^2 . H_{\hat{\mathrm{f}}}^{(-1)} \tag{2}$$

with

$$H_{\mathrm{f}} = \sum_i \left(\frac{\partial(\mathrm{f} \star x_i)}{\partial \mathrm{f}} \right)^{\mathrm{T}} \cdot \left(\frac{\partial(\mathrm{f} \star x_i)}{\partial \mathrm{f}} \right)$$

Moreover, σ_z^2 can be directly estimated from the residuals with $\hat{\sigma}_z^2 = 2.C(\hat{\mathrm{f}})/(3.(n-2))$.

Now, if we have an uncertainty information on our points (i.e. covariance matrices $\Sigma_{\boldsymbol{x_i x_i}}$ and $\Sigma_{\boldsymbol{y_i y_i}}$), the least squares estimator is no longer optimal and we have to minimize the Mahalanobis distance. We have proposed in [PT97] a approximated solution based on the Extended Kalman Filter. We do not detail this method here as it will appear as a special case of section 2.3.

1.2 Geometric features

Models of the real world often lead to consider geometric features that are more complex than points, such as lines [Gri90], planes, oriented points [FA96] or frames [PA94]. No traditional registration method easily extends to such features. Moreover, we have shown in [PA98] that geometric features generally do not belong to a vector space but rather to a manifold and that it can induce paradoxes if we use directly the standard techniques. Thus, we have developed in [Pen96] a rigorous theory of uncertainty on geometric features. The basic idea is that our measures on features should be invariant by the action of a given transformation group. We summarize here the basic concepts of this theory.

In the geometric framework, one specifies the structure of a manifold $\mathcal{M}$ by a *Riemannian metric*. This is a continuous collection of dot products on the tangent space at each point x of the manifold. Thus, if we consider a curve on the manifold, we can compute at each point its instantaneous speed vector and its norm, the instantaneous speed. To compute the length of the curve, we can proceed as usual by integrating this value along the curve. The distance between two points of a connected Riemannian manifold is the minimum length among the curves joining these points. The curves realizing this minimum for any two points of the manifold are called geodesics. Let o be a point of the manifold that we call the origin. We can develop the manifold in the tangent space at o along the geodesics (think of rolling a sphere along its tangent plane at the north pole). The geodesics going through the origin are transformed into straight lines and the distance along these geodesics are conserved. This realizes a chart called *the principal chart* that covers all the manifold (except a set of null measure called the cut locus). Let $\vec{\mathrm{y}}$ be the representation of y in this chart. Then its distance to the origin is $\mathrm{dist}(o, \mathrm{y}) = \|\vec{\mathrm{y}}\|$. This chart is somehow the "most linear" chart of the manifold with respect to the origin.

Now, since we are working with features on which acts a transformation group $\mathcal{G}$ that models the possible image view points, it is natural to choose an invariant Riemannian metric (the existence condition is detailed in [Pen96]). This way, all the measurements based on distance are independent of the image reference frame or the transformation of the image: $\mathrm{dist}(\mathrm{x}, \mathrm{y}) = \mathrm{dist}(\mathrm{g} \star \mathrm{x}, \mathrm{g} \star \mathrm{y})$. Let f_{x} be a "placement function", i.e. a transformation of G such that $\mathrm{f}_{\mathrm{x}} \star o = \mathrm{x}$. The distance can be expressed as: $\mathrm{dist}(\mathrm{x}, \mathrm{y}) = \mathrm{dist}\left(\mathrm{f}_{\mathrm{x}}^{(-1)} \star \mathrm{y}, o\right) = \|\mathrm{f}_{\vec{\mathrm{x}}}^{(-1)} \star \vec{\mathrm{y}}\|$. One can show that this formula is independent of the choice of the placement function.

In fact, we have shown that all interesting operations on deterministic and uncertain geometric features and transformations can be expressed in the principal chart of the manifold and the group with only a few basic algorithmic blocs that we call atomic operations.

For the group, we have to implement

- *the composition* $\vec{\mathrm{f}} \circ \vec{\mathrm{g}}$ and its Jacobians $\frac{\partial(\vec{\mathrm{f}} \circ \vec{\mathrm{g}})}{\partial \vec{\mathrm{f}}}$, $\frac{\partial(\vec{\mathrm{f}} \circ \vec{\mathrm{g}})}{\partial \vec{\mathrm{g}}}$,
- *the inversion* $\vec{\mathrm{f}}^{(-1)}$ and its Jacobian $\frac{\partial(\vec{\mathrm{f}}^{(-1)})}{\partial \vec{\mathrm{f}}}$.

For the feature manifold, the basic operations are

- *the action* $\vec{\mathrm{f}} \star \vec{\mathrm{x}}$ and its Jacobians $\frac{\partial(\vec{\mathrm{f}} \star \vec{\mathrm{x}})}{\partial \vec{\mathrm{f}}}$ and $\frac{\partial(\vec{\mathrm{f}} \star \vec{\mathrm{f}})}{\partial \vec{\mathrm{x}}}$,
- and *the placement function* $\vec{\mathrm{f}}_{\vec{\mathrm{x}}}$ with its Jacobian $\frac{\partial(\vec{\mathrm{f}}_{\vec{\mathrm{x}}})}{\partial \vec{\mathrm{x}}}$.

To simplify further computations, we can add to these atomic operations the Jacobian of the

- *left translation of the identity* $J_L(\vec{\mathrm{f}}) = \left.\frac{\partial(\vec{\mathrm{f}} \circ \vec{e})}{\partial \vec{e}}\right|_{\vec{e}=Id}$.

From this small set of atomic operations, we can implement the composition and inversion of a probabilistic transformation, the action of such a transformation on a probabilistic (or deterministic) feature, the distance and Mahalanobis distance between transformations, etc. We have implemented this framework for rigid transformations acting on different kinds of features such as frames, semi- or non-oriented frames and points. Frames are composed of a point and an orthonormal trihedron. In semi-oriented frames, the trihedron is composed of two directions $(t_1, t_2) \equiv (-t_1, -t_2)$ and an oriented vector n and models the differential properties of a point on a surface. For non oriented frames, the normal is also un-oriented $(n \equiv -n)$.

2 Pose estimation from matched features

Since we now have a distance between two geometric features and a Mahalanobis distance between two probabilistic features, the generalization of the two criteria for pose estimation from points is straightforward. However, it is rather unlikely to find a closed form solution like in the point case. Thus, we focus on iterative but generic algorithms.

2.1 Least squares on Riemannian distance

Let $\{\mathrm{x_i}\}$ and $\{\mathrm{y_i}\}$ be two sets of matched features. The Least squares criterion is easily written using the Riemannian distance:

$$\begin{aligned} C(\mathrm{f}) &= \frac{1}{2}.\sum_i \operatorname{dist}(\mathrm{y_i}, \mathrm{f} \star \mathrm{x_i})^2 \\ &= \frac{1}{2}.\sum_i \operatorname{dist}\left(\mathrm{f}_{\mathrm{y_i}}^{(-1)} \circ (\mathrm{f} \star \mathrm{x_i})\ ,\ \mathrm{o}\right)^2 \end{aligned}$$

2.1.1 Error vector

Now, thanks to the good properties of the principal chart, it turns out that this criterion can be expressed as a classical sum of square of vector norms. Let $\vec{\mathrm{z}}_i = \vec{\mathrm{f}}_{\vec{\mathrm{y}}_i}^{(-1)} \circ (\vec{\mathrm{f}} \star \vec{\mathrm{x}}_i)$ be the "error vector" in the principal chart. The criterion becomes

$$C(\mathrm{f}) = \frac{1}{2}.\sum_i \|\vec{\mathrm{z}}_i\|^2$$

This expression is only true in the principal chart and is false for any other representation.

To minimize the criterion and determine the uncertainty at the minimum, we will need to compute its first and second derivatives. From the atomic operations, and using the composition rule for differentials, we can compute the error vector $\vec{\mathrm{z}}_i$ and its Jacobians $\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{x}}_i}$, $\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}_i}$, and $\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}$ (see [Pen96] for equations). If the uncertainty of the measurements $\vec{\mathrm{x}}_i$ and $\vec{\mathrm{y}}_i$ is represented by the covariance matrices $\Sigma_{\vec{\mathbf{x}}_i\vec{\mathbf{x}}_i}$ and $\Sigma_{\vec{\mathbf{y}}_i\vec{\mathbf{y}}_i}$, the uncertainty of the error vector is:

$$\Sigma_{\vec{\mathbf{z}}_i\vec{\mathbf{z}}_i} = \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{x}}_i}.\Sigma_{\vec{\mathbf{x}}_i\vec{\mathbf{x}}_i}.\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{x}}_i}^{\mathrm{T}} + \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}_i}.\Sigma_{\vec{\mathbf{y}}_i\vec{\mathbf{y}}_i}.\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}_i}^{\mathrm{T}}$$

2.1.2 Derivatives of the criterion

Since the criterion is $C = \frac{1}{2}\sum_i \vec{z}_i^{\mathrm{T}}.\vec{z}_i$, the first derivative is:

$$\Phi = \left(\frac{\partial C}{\partial \vec{\mathrm{f}}}\right)^{\mathrm{T}} = \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}.\vec{z}_i$$

Neglecting the term in $\ddot{z}.z$ with respect to the terms in $\dot{z}^2$ in the second derivatives (see [GMW81, section 4.7]), we obtain:

$$H = \frac{\partial \Phi}{\partial \vec{\mathrm{f}}} \simeq \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}.\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}} \qquad \text{and} \qquad \frac{\partial \Phi}{\partial \vec{\mathrm{z}}_i} \simeq \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}$$

2.1.3 A gradient descent algorithm

Let $\vec{\mathrm{f}}$ be an estimation of the sought transformation. We have in the principal chart the following Taylor expansion (see [Pen96]):

$$\begin{aligned} C(\vec{\mathrm{f}} \circ \overrightarrow{\delta\mathrm{f}}) = \;& C(\vec{\mathrm{f}}) + \Phi^{\mathrm{T}}.J_L(\vec{\mathrm{f}}).\overrightarrow{\delta\mathrm{f}} \\ & + \frac{1}{2}\overrightarrow{\delta\mathrm{f}}^{\mathrm{T}}.J_L(\vec{\mathrm{f}})^{\mathrm{T}}.H.J_L(\vec{\mathrm{f}}).\overrightarrow{\delta\mathrm{f}} \\ & + O\left(\|\overrightarrow{\delta\mathrm{f}}\|^3\right) \end{aligned} \qquad (3)$$

where $\Phi = \frac{\partial C}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}$ and $H = \frac{\partial^2 C}{\partial \vec{\mathrm{f}}^2}$. The minimum of this approximated criterion is given for:

$$\overrightarrow{\delta\mathrm{f}} = -J_L(\vec{\mathrm{f}})^{(-1)}.H^{(-1)}.\Phi$$

Now, we can obtain a much better estimation of the transformation by iterating the previous algorithm. As an initial estimate, we can choose the identity if nothing else is given. Let $\vec{\mathrm{f}}_t$ be the estimation of the transformation at step t. The estimation at step $t+1$ is

$$\vec{\mathrm{f}}_{t+1} = \vec{\mathrm{f}}_t \circ \overrightarrow{\delta\mathrm{f}}_t \qquad \text{with} \qquad \overrightarrow{\delta\mathrm{f}}_t = -J_L(\vec{\mathrm{f}}_t)^{(-1)}.H_t^{(-1)}.\Phi_t \qquad (4)$$

The process is stopped when the norm $\|\overrightarrow{\delta\mathrm{f}}_t\|$ of the adjustment transformation becomes too small (we use $\varepsilon = 10^{-10}$, which is almost the numeric precision of the machine) or when the number of iterations becomes too high. Practically, we observe that this algorithm always converges in about 10 iterations for more than 3 matches.

2.1.4 Estimation of the uncertainty at the minimum

We have obtained above the transformation minimizing the criterion for the observed data. Now, since the data are corrupted by noise (represented by their covariance matrix), the estimated transformation could be slightly different from the exact one. We are now trying to characterize the transformation uncertainty as a function of the data uncertainty.

We assume for the moment that both the data and the state are vectors. Let $\hat{\chi}^{\mathrm{T}} = (\hat{z}_1, \ldots \hat{z}_n)$ be the vector of observed data and $\hat{f}$ the corresponding state (we have added here the hats to stress the difference between actual and potential measurements). We assume that all our measurements are independent so that $\Sigma_{\boldsymbol{\chi\chi}} = \mathrm{DIAG}(\Sigma_{\boldsymbol{z_1 z_1}}, \ldots \Sigma_{\boldsymbol{z_n z_n}})$. A Taylor expansion of the criterion around the actual values $(\hat{\chi}, \hat{f})$ gives:

$$\begin{aligned} C(\chi, f) &= \hat{C} + \hat{\Phi}^{\mathrm{T}}.(f - \hat{f}) + \frac{\partial C}{\partial X}.(\chi - \hat{\chi}) \\ &\quad + \tfrac{1}{2}(f - \hat{f})^{\mathrm{T}}.\hat{H}.(f - \hat{f}) \\ &\quad + O(\|f - \hat{f}\|^3) + O(\|\chi - \hat{\chi}\|^2) \end{aligned}$$

In this formula, $\hat{\Phi}$ and $\hat{H}$ are the Jacobian and the Hessian matrix of the criterion (with respect to f) estimated at $(\hat{\chi}, \hat{f})$. The minimum of the criterion is characterized by a null derivative with respect to the state. Using the above Taylor expansion, this derivative is:

$$\begin{aligned} \Phi(\chi, f) &= \hat{\Phi} + \hat{J}_{\Phi}.(\chi - \hat{\chi}) + \hat{H}.(f - \hat{f}) \\ &\quad + O(\|f - \hat{f}\|^2) + O(\|\chi - \hat{\chi}\|^2) \end{aligned}$$

where $\hat{J}_\Phi = \frac{\partial^2 C}{\partial \chi \partial f}(\hat{\chi}, \hat{f})$. Equating to zero, we obtain the new state:

$$f = \hat{f} - \hat{H}^{(-1)}.\hat{J}_\Phi.(\chi - \hat{\chi}) + O(\|\chi - \hat{\chi}\|^2)$$

Thus, the covariance of $\hat{f}$ is:

$$\Sigma_{\hat{f}\hat{f}} = \mathbf{E}\left((f - \hat{f}).(f - \hat{f})^{\mathrm{T}}\right) = \hat{H}^{(-1)}.\hat{J}_\Phi.\Sigma_{\hat{\chi}\hat{\chi}}\hat{J}_\Phi^{\mathrm{T}}.\hat{H}^{(-1)}$$

Now, using the particular form of χ and $\Sigma_{\chi\chi}$, we can simplify $\hat{J}_\Phi.\Sigma_{\hat{\chi}\hat{\chi}}\hat{J}_\Phi^{\mathrm{T}}$ to obtain:

$$\Sigma_{\hat{\mathrm{f}}\hat{\mathrm{f}}} = \hat{H}^{(-1)}.\left(\sum_i \frac{\partial \hat{\Phi}}{\partial \vec{\mathrm{z}}_i}.\Sigma_{\mathbf{z_i z_i}}.\frac{\partial \hat{\Phi}}{\partial \vec{\mathrm{z}}_i}^{\mathrm{T}}\right).\hat{H}^{(-1)} \tag{5}$$

Last but not least, the data and the state are not vectors in our case, but points respectively in a manifold and in a transformation group. However, thanks to be properties of the principal chart, this derivation is still valid if we replace $(f - \hat{f})$ by $J_L(\hat{\vec{\mathrm{f}}}).(\hat{\vec{\mathrm{f}}}^{(-1)} \circ \vec{\mathrm{f}})$ and $(\chi - \hat{\chi})$ by a somehow similar expression. Now, it turns out that the definition of the covariance is changed accordingly and that finally nothing is changed in equation (5).

2.1.5 Uncertainty for the least-squares solution

As for points, the least-square estimation is optimal only if $\Sigma_{\mathbf{z_i z_i}} = \sigma_z^2.Id$. Then, taking into account that $\frac{\partial \Phi}{\partial \vec{\mathrm{z}}_i} \simeq \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}$, the uncertainty is simplified into

$$\Sigma_{\hat{\mathrm{f}}\hat{\mathrm{f}}} = \sigma_z^2.H^{(-1)}$$

Moreover, the variance σ_z^2 can be estimated from the data with

$$\hat{\sigma}_z^2 = \frac{\sum_i \|\hat{\vec{z}}_i\|^2}{3(n-3)} \text{ for points} \qquad \hat{\sigma}_z^2 = \frac{\sum_i \|\hat{\vec{z}}_i\|^2}{6(n-1)} \text{ for frames}$$

However, this hypothesis is quite strong for frame-based features: it means that the variance σ_θ^2 on the rotation vector representing the trihedron is equal to the variance σ_x^2 on the position of the frame. Thus the choice of the units (radian, degree, meter or millimeter) is crucial. Most of the time, this hypothesis does not hold and we have to use equation (5).

2.2 Gradient descent on the Mahalanobis distance

To allow different and non isotropic covariance matrices for different measure, we can minimize the sum of squared Mahalanobis distances after registration. It turns out that this Mahalanobis distance can be expressed with exactly the same error vector as before:

$$\begin{aligned} C(\mathrm{f}) &= \frac{1}{2}\sum_i \mu^2(\mathbf{y}_i, \mathrm{f} \star \mathbf{x}_i) \\ &= \frac{1}{2}\sum_i \mu^2(\mathbf{z}_i, 0) = \frac{1}{2}\sum_i \vec{\mathrm{z}}_i^{\mathrm{T}}.\Sigma_{\mathbf{z_i z_i}}^{(-1)}.\vec{z}_i \end{aligned}$$

Thus, the algorithm is the same than for the least squares, but the derivatives of the criterion are different. The first derivative is

$$\Phi = \left(\frac{\partial C}{\partial \vec{\mathrm{f}}}\right)^{\mathrm{T}} = \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}.\Sigma_{\mathbf{z_i z_i}}^{(-1)}.\vec{z}_i$$

Similarly, the second derivatives can be approximated by:

$$H = \frac{\partial \Phi}{\partial \vec{\mathrm{f}}} \simeq \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}.\Sigma_{\mathbf{z_i z_i}}^{(-1)}.\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}$$

and

$$\frac{\partial \Phi}{\partial \vec{\mathrm{z}}_i} \simeq \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{f}}}^{\mathrm{T}}.\Sigma_{\mathbf{z_i z_i}}^{(-1)}$$

Now, the Taylor expansion for the criterion is the same, and the evolution for the gradient descent is still given by equation (4). We can use the same starting value and stopping criterion as before. Practically, we have observe a convergence in about 15 iterations when starting from identity and in 5 to 10 iterations when starting from the least-squares solution.

For the uncertainty of the solution, we replace the values of H and $\frac{\partial \Phi}{\partial \vec{\mathrm{z}}_i}$ into equation (5) and obtain:

$$\Sigma_{\hat{\mathrm{f}}\hat{\mathrm{f}}} = \hat{H}^{(-1)}$$

2.3 Kalman filtering

Assume that we have an initial estimate $\vec{\mathrm{f}}_0$ with covariance $\Sigma_{\vec{\mathrm{f}}_0\vec{\mathrm{f}}_0}$. The natural criterion to minimize becomes

$$\begin{aligned} C(\mathrm{f}) &= \frac{1}{2}\mu^2(\mathrm{f}, \mathrm{f}_0) + \frac{1}{2}\sum_i \mu^2(\mathrm{y}_i, \mathrm{f} \star \mathrm{x}_i) \\ &= \frac{1}{2}(\vec{\mathrm{f}}_0^{(-1)} \circ \vec{\mathrm{f}})^{\mathrm{T}}.J_L(\vec{\mathrm{f}}_0)^{\mathrm{T}}.\Sigma_{\vec{\mathrm{f}}_0\vec{\mathrm{f}}_0}^{(-1)}.J_L(\vec{\mathrm{f}}_0).(\vec{\mathrm{f}}_0^{(-1)} \circ \vec{\mathrm{f}}) \\ &\quad + \frac{1}{2}\sum_i \vec{\mathrm{z}}_i^{\mathrm{T}}.\Sigma_{\mathbf{z_i z_i}}^{(-1)}.\vec{z}_i \end{aligned}$$

In the case of linear transformations of Gaussian random vectors, the maximum likelihood estimation and minimum error variance estimation corresponds, and could be expressed as a recursive filter known as the Kalman filter [Jaz70]. In the general case of non-linear transformations (which is indeed our case here), the filter could be extended by using linear approximations around the current state and measurements [Aya91]. At each step the input is

- an estimate $(\vec{\mathrm{f}}_{i-1}, \Sigma_{\vec{\mathbf{f}}_{i-1}\vec{\mathbf{f}}_{i-1}})$ of the state,
- the measurement $(\vec{z}_i, \Sigma_{\vec{\mathbf{z}}_i\vec{\mathbf{z}}_i})$
- and the linearized dependency between the state and the measurement $M_i = \left.\frac{\partial \vec{z}_i}{\partial \vec{\mathrm{f}}}\right|_{\vec{\mathrm{f}}=\vec{\mathrm{f}}_{i-1}}$.

The output is the updated estimation of the state using the following equations:

$$\begin{aligned} K_i &= \Sigma_{\vec{\mathbf{f}}_{i-1}\vec{\mathbf{f}}_{i-1}}.M_i^{\top}.(\Sigma_{\vec{\mathbf{z}}_i\vec{\mathbf{z}}_i} + M_i.\Sigma_{\vec{\mathbf{f}}_{i-1}\vec{\mathbf{f}}_{i-1}}.M_i^{\top})^{-1} \\ \vec{\mathrm{f}}_i &= \vec{\mathrm{f}}_{i-1} - K_i.\vec{z}_i \\ \Sigma_{\vec{\mathbf{f}}_i\vec{\mathbf{f}}_i} &= (Id - K_i.M_i).\Sigma_{\vec{\mathbf{f}}_{i-1}\vec{\mathbf{f}}_{i-1}} \end{aligned}$$

However, we must be careful that the state $\vec{\mathrm{f}}_i$ is constrained to belong to the principal chart and the above Kalman evolution equation could make it go out of its definition domain. Thus, we have to verify after each step that the state is still in the definition domain and correct if it is out (this is done using an additional atomic operation called "Domain" in [Pen96]).

3 Algorithm comparison

In the following, we denote by LSQ, MAHA and KAL the three pose estimation methods developed above. Since we can forget the trihedron in all types of frames to keep only a point, we can also apply the standard pose estimation techniques on points, such as the unit quaternion method, which we denote by QUAT. We analyze here the relative accuracy of the different methods, their computation time and the accuracy of the uncertainty on the transformation (which should predict the absolute accuracy).

Statistics are realized by choosing a random sample of N features in an image 512x256x128. This is the exact model. We choose a random rigid movement (the “exact” transformation), and transform the exact model into the exact scene. Then each exact feature is perturbed by a Gaussian noise with a given covariance Σ in order to constitute the observed model and scene.

3.1 Accuracy of the methods

With the same couple of (synthetic) model and scene, we compute the transformation using QUAT, LSQ, KAL and MAHA. For each of these estimations, we compute the residual transformation with respect to the exact transformation. To simplify the analysis, we only compare the residual angle of rotation θ and the length d of the residual translation. As these values are varying a lot with the considered set of matches and since we only want to study the relative accuracy of the three methods, we have chosen the error of MAHA as the reference. Thus, we compute the ratios $\varepsilon_{\theta_{\mathrm{LSQ}}} = \theta_{\mathrm{LSQ}}/\theta_{\mathrm{MAHA}}$ and $\varepsilon_{d_{\mathrm{LSQ}}} = d_{\mathrm{LSQ}}/d_{\mathrm{MAHA}}$ to characterize the relative accuracy of LSQ and similarly for KAL: $\varepsilon_{\mathrm{LSQ}} < 1$ means that LSQ is more accurate than MAHA and $\varepsilon_{\mathrm{LSQ}} > 1$ means less accurate.

To obtain reliable statistics, we have done 250 trials for each number of features and we present in fig. (1) the mean value and the standard deviation[1] of the relative accuracies of LSQ and KAL with respect to MAHA.

In these experiments, LSQ and MAHA are initialized with the identity, whereas KAL is initialized with the result of LSQ (the covariance matrix is multiplied by 50 to minimize its influence on the final uncertainty estimation). Indeed, initializing KAL with the identity and a fixed covariance matrix (or any transformation a little too far from the solution) leads to a pseudo-convergence toward a false solution. On the contrary, we did not see any influence of the initialization on the solutions of LSQ and MAHA (except for the computation time).

The next observation is that the absolute accuracy of each method is proportional to $1/\sqrt{n}$ (where n is the number of features) but the ratio of the accuracy of different methods does not depend on the number of matches. We found that KAL and MAHA perform very similarly. So do QUAT and LSQ. On the contrary, we found that KAL and MAHA are 1.2 times to 2 times more accurate than QUAT and LSQ depending on the covariance matrix on frames (more particularly the ratio of the orientation variance over the position variance).

[1] As the relative accuracy is multiplicative (2 times more accurate means $\varepsilon = 0.5$), we have computed the mean value and the standard deviation of $\log(\varepsilon)$, but we present the result with ratios for an easier visualization.

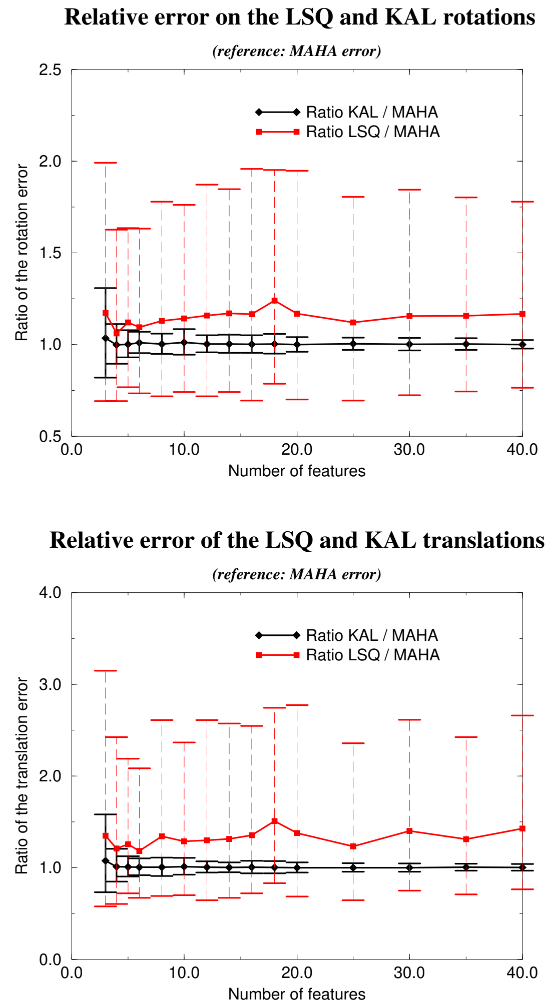

Figure 1: Relative accuracy of the rotation and translation estimated by LSQ and KAL with respect to MAHA.

3.2 Accuracy of the uncertainty estimation

3.2.1 The validation index

With synthetic data, we know the exact motion that we are looking for, and we can verify that it is compatible with our transformation estimation modulo its uncertainty. The basic idea of the validation method presented in [PT97] is normalize the residual transformation error with respect to the predicted uncertainty. In fact, the simplest way do do that is to compute the Mahalanobis distance between the computed and the exact transformation:

$$\mu^2(\mathrm{f}, \hat{\mathbf{f}}) = (\hat{\vec{\mathrm{f}}}^{(-1)} \circ \vec{\mathrm{f}})^{\mathrm{T}}.J_L(\hat{\vec{\mathrm{f}}})^{\mathrm{T}}.\Sigma^{(-1)}_{\hat{\mathbf{f}}\hat{\mathbf{f}}}.J_L(\hat{\vec{\mathrm{f}}}).(\hat{\vec{\mathrm{f}}}^{(-1)} \circ \vec{\mathrm{f}})$$

Under the Gaussian hypothesis (justified in [Pen96]), this value should be χ^2_6 distributed if the covariance matrix is exact.

We can now repeat this experiment on N pairs of images to obtain N independent values μ_i^2 and verify if it is really χ^2_6 distributed. The Kolmogorov-Smirnov (or simply K-S) test [PFTV91] is well adapted to do that but since it only gives a binary answer, we also use the fact that the mean value of a χ_6^2 distribution is 6 and its variance is 12. We call *validation index* the estimated mean value of μ_i^2:

$$I = \bar{\mu}^2 = \frac{1}{N} \sum \mu_i^2 \qquad \text{[ideal value: 6]}$$

and its variance is computed accordingly with

$$\sigma_I^2 = \frac{1}{N-1} \sum (\mu_i^2 - \bar{\mu}^2)^2 \qquad \text{[ideal value: 12]}$$

This index can be interpreted as an indications on how the estimation method under-estimates ($I > 6$) or over-estimates ($I < 6$) the error on the estimated motion. It is a kind of relative error on the error estimation.

This method can be generalized to the case of real data by splitting the set of matches in two approximatively equal sets on matches. Then one two independent estimates $\mathbf{f}_1$ and $\mathbf{f}_2$ of the same transformation f and their Mahalanobis distance $\mu^2(\mathbf{f}_1, \mathbf{f}_2)$ should once again be χ_6^2 distributed under the Gaussian assumption. The two transformations could then be fused to obtain a better estimation of the transformation and used for other purposes (figure 2).

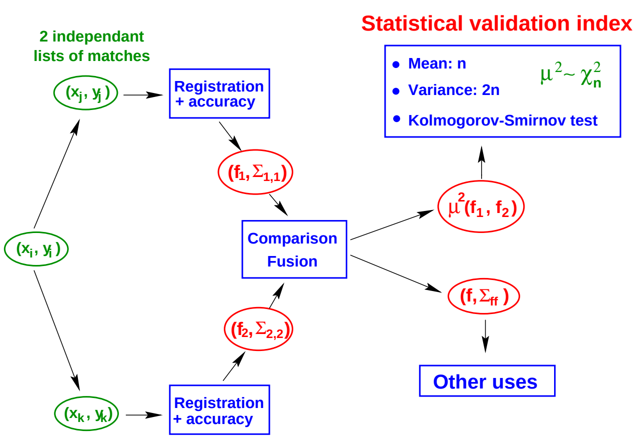

Figure 2: Validation of the registration uncertainty with real data.

3.2.2 Validation results

As expected, the best accuracy estimation is given by MAHA. We present in figure 3 the computed mean and standard deviation of the validation index with MAHA on semi-oriented frames with respect to the number of matches. The dotted lines represent the theoretical values for a χ_6^2 distribution (a mean of 6 and a standard deviation of $\sqrt{12} = 3.46$). The K-S test at 1% is always accepted. The mean validation index over the $25 * 500 = 12500$ registrations in this graphic is $I = 6.05$ for a variance of $\sigma_I^2 = 11.90$, which proves that the uncertainty estimation is very accurate. This indicates an underestimation of the covariance matrix of less than 1%. The same results are observed with other kind of feature.

The graphic for KAL is quite similar, except that the K-S test is one order of magnitude worse. Indeed, the mean validation index is now $I = 6.29$ for a variance of $\sigma_I^2 = 12.90$, indicating an underestimation of 4.6%.

In the case of LSQ, the hypothesis of a diagonal and isotropic covariance matrix on features does not hold (except for points). Thus, using formula $\Sigma_{\hat{\mathbf{f}}\hat{\mathbf{f}}} = \sigma_z^2.H^{(-1)}$ leads to a

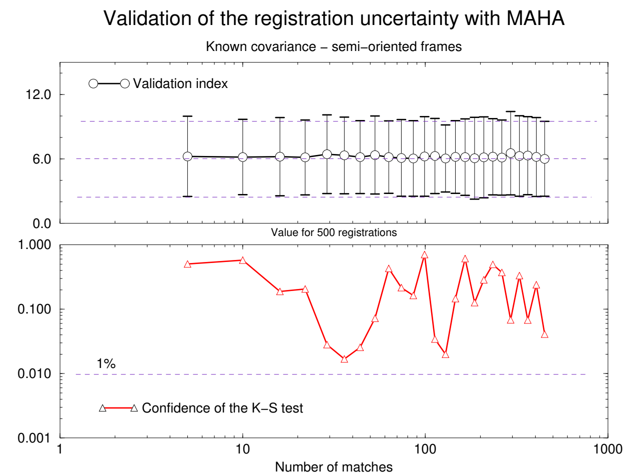

Figure 3: Validation of the registration uncertainty with a known covariance on semi-oriented frames (method MAHA). Top: mean and standard deviation of the validation index with respect to the number of matches used. Bottom: confidence value of the K-S test. It validates the uncertainty in all cases with a confidence superior to 1%.

very estimation of the uncertainty. However, We can still use formula (5) and in this case the uncertainty is much better estimated. We obtain a mean validation index of $I = 5.82$ for a variance of $\sigma_I^2 = 11.09$. This means a 3% overestimation of the uncertainty on the transformation.

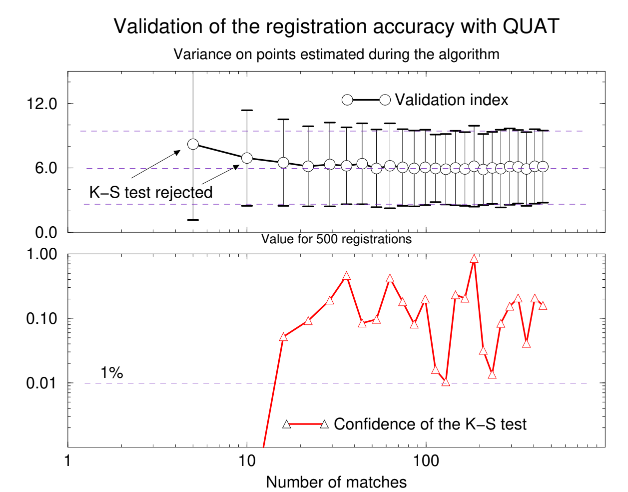

Figure 5: Validation of the registration uncertainty with QUAT. The variance on points is estimated during the algorithm. Top: mean and standard deviation of the validation index with respect to the number of matches used. Bottom: confidence value of the K-S test. It validates the uncertainty with a confidence superior to 1% for more than 15 matches.

The case of QUAT is more interesting: in most cases, the hypothesis of a diagonal and isotropic covariance matrix on points is justified (there is hardly ever an order a magnitude of difference between the 3 directions of the space). Hence, we can use the simplified formula and estimate the variance

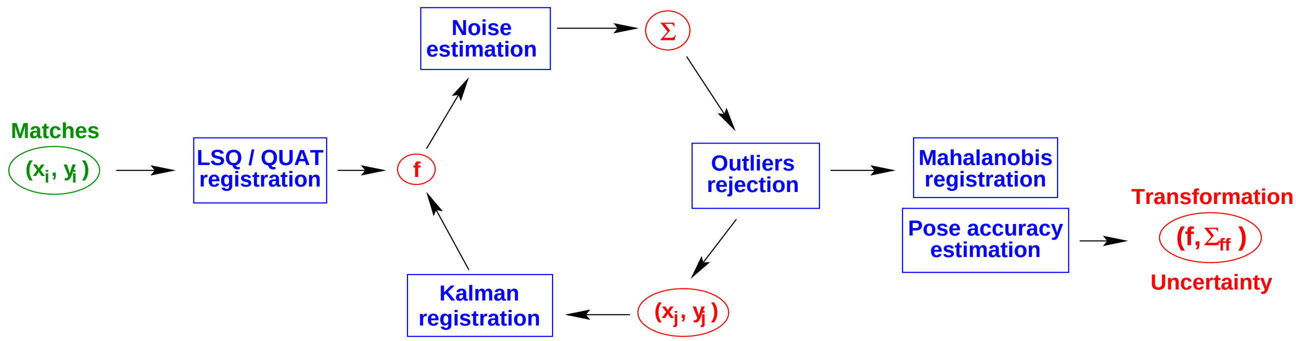

Figure 4: Modular registration algorithm.

σ_z^2 on points directly from the data. This make an important difference with the previous experiments where the noise model was known. Therefore, we do not expect the uncertainty to be very accurate with a small number of matches. Indeed, we observe in figure (5) that the uncertainty is completely underestimated for less than 15 matches, but is quite correctly estimated (and validated by the K-S test) for more than 15 matches. The mean validation index in this part is $I = 6.06$ for a variance of $\sigma_I^2 = 12.51$, which corresponds to an underestimation of 1%.

3.3 Computation times

The algorithm LSQ is initialized with the identity but LSQ and MAHA are initialized with the result of QUAT. The computation times are approximately linear in the number of matches. Thus, we only give below the computation times for 100 matches on a DEC alpha 3000 166 Mhz (average value on 100 to 500 trials)

	Frames	S.O. Frames	N.O. Frames	Points
QUAT				20 ms
LSQ	694 ms	785 ms	820 ms	275 ms
KAL	216 ms	245 ms	250 ms	87 ms
MAHA	792 ms	880 ms	870 ms	368 ms

It is clear that all the algorithms are much slower than QUAT but they still remain reasonable with respect to the increase in accuracy they provide.

3.4 Discussion: use of the algorithms

Due to the choice of the metric (the units for the rotation and the translation), LSQ and QUAT give very similar results. QUAT is much faster and should thus be used when possible, i.e. for more than 3 matches. However, LSQ is still usable for 1 or 2 frame matches (but could be unstable). For the transformation uncertainty, QUAT does not need the covariance matrix on features, but it is only reliable with more than 15 matches. LSQ needs to know the covariance matrix on features is order to produce a reliable uncertainty estimation.

KAL and MAHA are quite similar and give a better transformation accuracy. MAHA gives an accurate transformation and uncertainty in all cases, even with very few matches (it is more stable than LSQ). KAL is faster but requires a reliable initialization point and underestimates the uncertainty. However, it is incremental and we can forget the matches when they have updated the transformation, which could prove to be a real advantage for the memory complexity. Another combination is to compute the transformation with MAHA for the first (let say 10) matches and then update iteratively the transformation with the other using KAL.

	QUAT	LSQ	KAL	MAHA
Cov. Mat.	No	No	Yes	Yes
Comp. time	++	–	+	– –
Transfo. accuracy	–	–	+	+
Uncert. accuracy	–	–	– –	+
Sensitivity		Metric	Init. cond.	

3.5 A practical and generic algorithm

At the end of a matching algorithm, we usually have a list of matched features with possibly outliers and no or little information about the noise on features. We have proposed in [PT97] an algorithm that estimates the pose, its uncertainty, the outliers and the noise on features in the case of frames. This algorithm could be easily generalized to generic features and modified to make the best use of each characteristics of the previous algorithms (figure 4).

Since we have no information about the uncertainty of features, we begin by a global QUAT or LSQ registration. Then we can estimate the noise on features, and use a χ^2 test to rule out outliers. The transformation is then updated using KAL and the process it iterated. When the transformation becomes stable or after a maximum number of iterations (5 to 10), we finally use a MAHA estimation to ensure the accuracy of the uncertainty estimation.

4 Estimation of the relative motion of the bones in the pelvis

The goal of the experiment was to measure quantitatively the relative motion of bones in the pelvis during particular motions. Two MRI (T1) images were taken at the Lenval Foundation (Nice, France), one in "open pelvis" configuration and one in "close pelvis" configuration. Images are 256 x 256 x 28 with a voxel size of 1.33 x 1.33 x 5.6 mm (i.e. an anisotropic factor greater than 4 in the z direction!).

The pelvis is principally constituted of the sacrum and the left and right hipbones, forming a quasi-rigid structure. Hence their relative motion is very small and we need to estimate the motions of each bone and their uncertainty to be able to decide statistically if there is really a motion between them.

Figure 6: An axial slice of an MR image of the pelvis (T1 weighted). One can see the sacrum in the bottom center and the hipbones on the sides.

4.1 Segmentation

It is quite hard to segment bones in an MRI image. Indeed, the iso-surface method that we usually use for bones in CT-scans or the surface of the brain in MRI images does not work here: the bones appear as an intermediate signal delimited by a narrow band of weaker signal on the border (figure 6). This weak signal has a varying intensity and is moreover asymmetric due to the chemical shift.

Since the other automatic segmentation tools we tried also failed, each bone was manually segmented by a specialist (Pr. Dourthe) in both images. The result is presented in figure (7).

4.2 Registrations

To register the bones in the two images, we extracted the crest lines on the bones surface and some particular points on these lines called extremal points [TG93]. These points optimize a differential geometry criterion and can be seen as a generalization of corner points for smooth surfaces. As they are points on a surface, we have not only a point but also the normal to the surface and the two principal directions, i.e. a *semi-oriented frame*. Each bone is thus modeled as a set of semi-oriented frames.

The matches between the extremal points of a bone in one image and their counterpart in the second images were determined using a geometric hashing method described in [GPA97]. We found only 30 matches for the hipbones and 35 for the sacrum. Then we used the pose estimation algorithm described in the previous section to compute the transformations. In figure (8) we present the superimposition of the pelvis obtained using the sacrum transformation only. Considering that the sacrum is fixed, we can visualize the hipbones motion: the hipbones in the "open pelvis" configuration effectively seem to be slightly more open than in the "closed pelvis" configuration. The residual transformation between hipbones confirms that impression since it is a rotation of 1.2 deg around the vertical axis and a translation of 3 to 4 mm that tends to open the hipbones.

4.3 Statistical results

However, when it comes to the uncertainty, the results are much less clear. The uncertainty estimated on features is rather large: the standard deviation of the residual transformation is around 25 degrees for the rotation (trihedron) and 1 mm for the translation (position). This leads to a very high uncertainty on the estimated transformations: around 3.5 degrees for the rotation and 1.5 mm for the translation. Comparing the residual transformations (left and right hipbones with respect to the sacrum or left hipbone with respect to the right hipbone) with the identity (rigid hypothesis: no motion inside the pelvis), we found Mahalanobis distances of 16 and 18, which approximately corresponds to the threshold of χ^2 tests at 98%. Thus, we cannot conclude that there is a statistically significant relative motion of bones in the pelvis. In this example, the availability of higher order statistics on the registration prevented us from a possibly wrong conclusion. However, we cannot conclude either that there is no relative motion in the pelvis. In fact, we would need more precise data to be able to conclude statistically. This could be obtained by using images with a better resolution, a better (semi-automatic) segmentation of the bones, and more extremal points on each bone surface.

5 Conclusion

We show in this article how to generalize the classical least squares and least Mahalanobis distance pose estimation algorithms to generic geometric features, and how to estimate the uncertainty of the result. The analysis of the three new algorithms shows that each one has advantages and drawbacks. The uncertainty prediction on the transformation is validated within a bound of 5% for all algorithms, decreasing to less than 1% of inaccuracy for the algorithm MAHA.

Then, we show how to combine all the algorithms into a modular and high level registration algorithm to exploit all their particularities. The accuracy of the transformations uncertainties is not considered in this article but is perfectly validated in [Pen96, chap. 9]. We believe that these algorithms could be easily embedded in many applications and provide a thorough basis for computing many image statistics.

In the last part, we show an application in medical imaging where the availability of the transformation uncertainties prevents us from a possible wrong conclusion. This also stress that the quality of the registration directly depends on the quality and the number of features used. By the way, we could think to evaluate the quality of different feature extraction methods by the accuracy of the estimated pose.

Further improvements could include the generalization of this framework to non rigid transformations, the estimation of a multiple registration of n sets of matched features and the coupling with statistical matching algorithms to provide a complete registration system.

References

[AHB87] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.

[Aya91] N. Ayache. *Artificial Vision for Mobile robots - Stereo-vision and Multisensor Perception.* MIT-Press, 1991.

[DW88a] H.F. Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems.* Kluwer Academic Publishers, 1988.

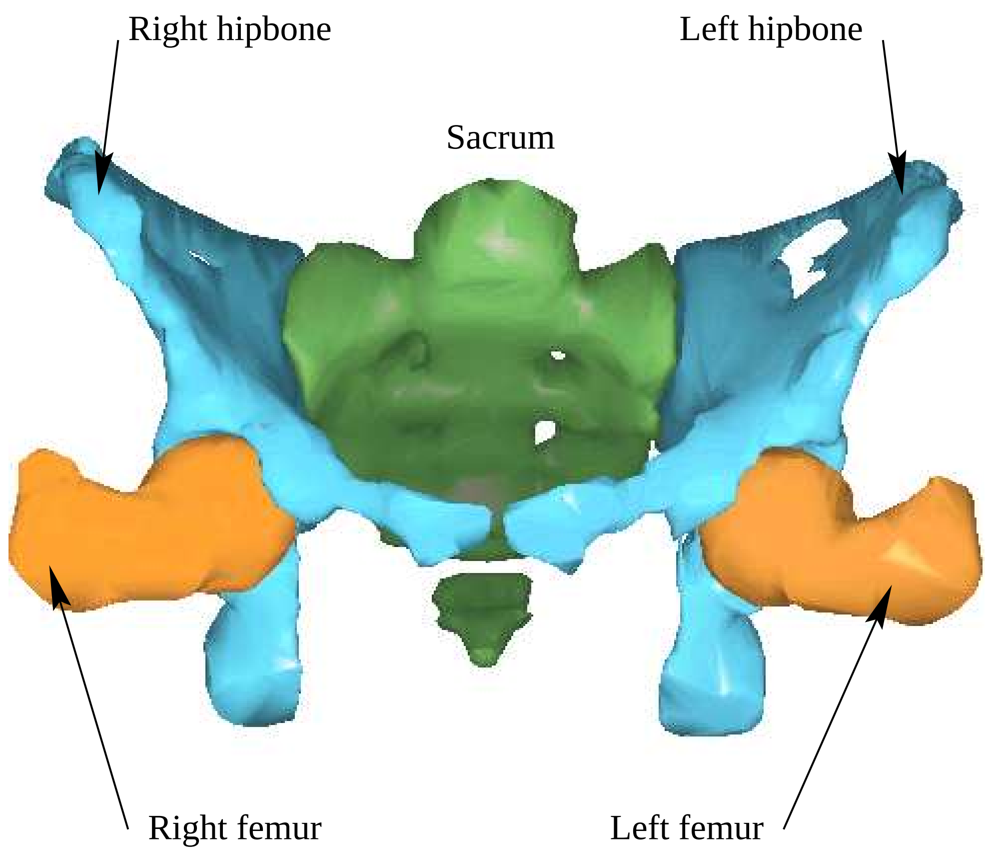

Figure 7: Result of the manual segmentation of one MRI image. The surface of each bone is available independently and can be labeled as in this view.

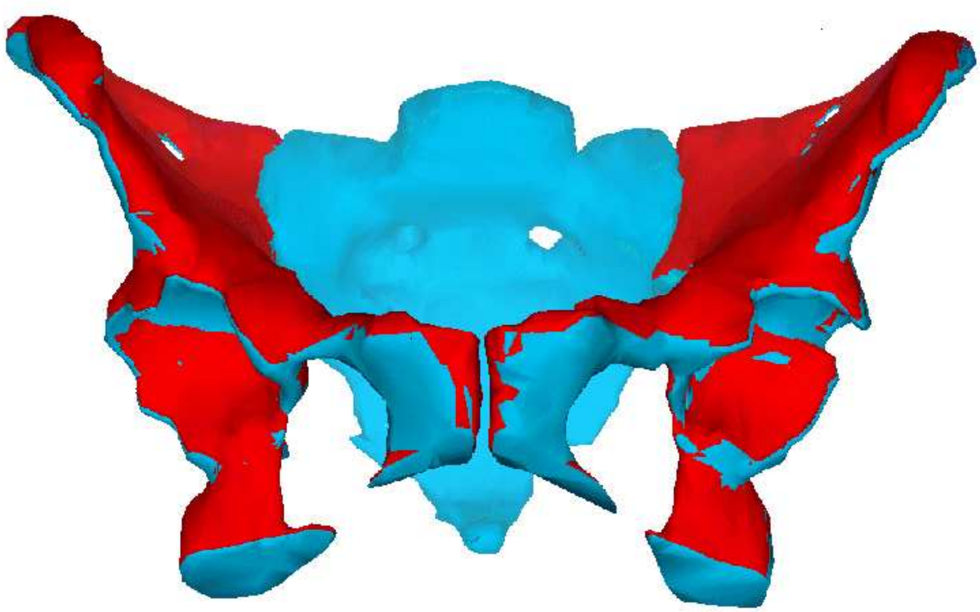

Figure 8: Superimposition of the "open" and "closed" pelvis using the sacrum registration. the hipbones in the "open pelvis" configuration (in bright) effectively seem to be slightly more open than in the "closed" configuration (in dark).

[DW88b] H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, February 1988.

[FA96] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *IJCV*, 18(2):99–119, May 1996.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT press, 1993.

[FH84] J. Fang and T.S. Huang. Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):545–554, September 1984.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. Academic Press, London and New York, 1981.

[GPA97] A. Guéziec, X. Pennec, and N. Ayache. Medical image registration using geometric hashing. *IEEE Computational Science & Engineering, special issue on Geometric Hashing*, 4(4):29–41, 1997. Oct-Dec.

[Gri90] W.E.L. Grimson. *Object Recognition by Computer - The role of Geometric Constraints*. MIT Press, 1990.

[HHN88] B.K.P. Horn, Hilden H.M., and S. Negahdaripour. Closed form solutions of absolute orientation using orthonormal matrices. *Journal of Optical Society of America*, A-5(7):1127–1135, 1988.

[HJL+89] R.M. Haralick, H. Joo, C.N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *IEEE transaction on Systems, Man and Cybernetics*, 19(6):1426–1446, November,December 1989.

[Hor87] B.K.P. Horn. Closed form solutions of absolute orientation using unit quaternions. *Journal of Optical Society of America*, A-4(4):629–642, April 1987.

[Jaz70] A.M. Jazwinsky. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[Kan93] K. Kanatani. *Geometric Computation for Machine Vision*. Number 37 in The Oxford Engineering science series. Clarendon Press - Oxford, 1993.

[Kan94] K. Kanatani. Analysis of 3-D rotation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):543–549, May 1994.

[LEF95] A. Lorusso, D.W. Eggert, and R.B. Fisher. A comparison of four algorithms for estimating 3-d rigid transformation. In *Proc. 6th British Machine Vision Conference (BMVC'95)*, pages 237–246, 1995.

[PA94] X. Pennec and N. Ayache. An $\mathcal{O}(n^2)$ algorithm for 3D substructure matching of proteins. In A. Califano, I. Rigoutsos, and H.J. Wolson, editors, *Shape and Pattern Matching in Computational Biology – Proc. First Int. Workshop, Seattle, Wash, June 20, 1994*, pages 25–40. Plenum Publishing, 1994. also as INRIA Research Report n^o 2274.

[PA98] X. Pennec and N. Ayache. Uniform distribution, distance and expectation problems for geometric features processing. *Journal of Mathematical Imaging and Vision*, 9(1):49–67, July 1998.

[Pen96] Xavier Pennec. *L'Incertitude dans les Problèmes de Reconnaissance et de Recalage – Applications en Imagerie Médicale et Biologie Moléculaire*. PhD thesis, Ecole Polytechnique, Palaiseau (France), december 1996.

[PFTV91] W.H. Press, B.P. Flannery, S.A Teukolsky, and W.T. Vetterling. *Numerical Recipices in C*. Cambridge Univ. Press, 1991.

[PT97] X. Pennec and J.P. Thirion. A framework for uncertainty and validation of 3D registration methods based on points and frames. *Int. Journal of Computer Vision*, 25(3):203–229, 1997.

[Sch66] P.H. Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31:1–10, 1966.

[Sny89] M.A. Snyder. The precision of 3-D parameters in correspondence-based techniques: The case of uniform translational motion in a rigid environment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):523–528, May 1989.

[TG93] J-P. Thirion and A. Gourdon. The 3D marching lines algorithm : new results and proofs. Technical Report 1881, INRIA, March 1993. published in CVIU in 1995.

[Ume91] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, April 1991.

[WS91] M.W. Walker and L. Shao. Estimating 3-D location paramters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, Nov. 1991.

[ZF92] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis: a stereo based approach*, volume 27 of *Springer series in information science*, chapter 2: Uncertainty Manipulation and Parameter Estimation, pages 9–27. Springer Verlag, 1992.