Auxiliary Variables and Two-step Iterative Algorithms in Computer Vision Problems

Laurent D. COHEN
CEREMADE, U.R.A. CNRS 749, Université Paris 9 - Dauphine,
Place du Maréchal de Lattre de Tassigny 75775 Paris cedex 16, France.
Email: Laurent.Cohen@Ceremade.dauphine.fr

To appear in Journal of Mathematical Imaging and Vision
Cahiers de Mathématiques de la Décision 9511

Abstract

We present a new mathematical formulation of some curve and surface reconstruction algorithms by the introduction of auxiliary variables. For deformable models and templates, the extraction of a shape is obtained through the minimization of an energy composed of an internal regularization term (not necessary in the case of parametric models) and an external attraction potential. Two-step iterative algorithms have been often used where, at each iteration, the model is first locally deformed according to the potential data attraction and then globally smoothed (or fitted in the parametric case).

We show how these approaches can be interpreted as the introduction of auxiliary variables and the minimization of a two-variables energy. The first variable corresponds to the original model we are looking for, while the second variable represents an auxiliary shape close to the first one. This permits to transform an implicit data constraint defined by a non convex potential into an explicit convex reconstruction problem. This approach is much simpler since each iteration is composed of two simple to solve steps. Our formulation permits a more precise setting of parameters in the iterative scheme to ensure convergence of the energy to a minimum.

We show some mathematical properties and results on this new auxiliary problem, in particular when the potential is a function of the distance to the closest feature point. We then illustrate our approach for some deformable models and templates.

Keywords: Deformable Models and Templates, Distance Map, Energy Minimization, Image Restoration, Feature Extraction, Pattern Matching, Registration, Shape Extraction and Regularization, Spline Functions, Surface and Curve Reconstruction.

Technical Biography of the Author

Laurent David Cohen was born in 1962. He was student at the **Ecole Normale Supérieure**, rue d'Ulm in Paris, France from 1981 to 1985. He received the Master's and Ph.D. degrees in Applied Mathematics from University of Paris 6, France, in 1983 and 1986, respectively.
From 1985 to 1987, he was member at the Computer Graphics and Image Processing group at Schlumberger Palo Alto Research, Palo Alto, California and Schlumberger Montrouge Research, Montrouge, France and remained consultant with Schlumberger afterwards. He began working with INRIA, France in 1988, mainly with the medical image understanding group Epidaure.
He obtained in 1990 a position of Research Scholar with the French National Center for Scientific Research (CNRS) in the Applied Mathematics and Image Processing group at CEREMADE, University Paris-Dauphine, Paris, France. His research interests and teaching at university are applications of variational methods to Image Processing and Computer Vision, like deformable models, surface reconstruction, Image segmentation and restoration.

1 Introduction

Many problems in Computer Vision are formulated by the minimization of an energy. Using deformable models and templates, the extraction of a shape is obtained through the minimization of an energy composed of an internal regularization term and an external attraction potential (data term), examples can be found in [1, 2, 3, 4, 5, 6, 7, 8] for applications in segmentation, surface reconstruction, image restoration and feature extraction.

The minimization is usually solved by gradient descent using an iterative scheme. Often, to make the task possible or easier, each iteration is divided in two stages which may be interpreted as a separation between a global transform and a local one. It consists in minimizing separately the two terms of the energy. While each step may make one part of the energy decrease, it may globally increase. Also, it may happen after this change, that while the method converges to sensible results, the problem solved differs from the initial one. Our work gives a good mathematical formulation to many of these data extraction and reconstruction algorithms by adding an auxiliary variable in the energy.

The original contribution of this work is twofold. First we introduce auxiliary variables to define a two-variable energy for shape extraction and reconstruction and show some mathematical results on its minimization. This permits at each iteration to transform a problem with implicit data constraints defined through the minimization of a potential into a reconstruction with regularization of explicit data.

We then apply these results to some already existing two-step iterative algorithms for deformable templates and deformable models to show how they can be interpreted as a minimization of a two-variable energy. We give by the way a uniform mathematical formulation of many computer vision problems. Using our formulation has the advantage to be sure we minimize, in these two steps, the good initial energy with one variable.

In general, the energy can be written:

$$E(v) = \int R(v) + \int P(v) \tag{1}$$

where v is the unknown shape, R is the regularization term and P is the potential. In case the model is parametric, the internal energy is not necessary and the potential alone is minimized.

The energy we introduce with the extra auxiliary variable w has this form:

$$E_{aux}(v,w) = \int R(v(s))ds + \frac{1}{2}\int \|v(s) - w(s)\|^2 ds + \int P_1(w(s))ds \tag{2}$$

For a given w, the minimization of this energy with respect to v is an explicit regularization convex problem. For a given estimate of v, the minimization of E_{aux} with respect to w is a problem that can be made convex even if P and P_1 are not. We show how this minimization is solved in a straightforward way and that there is a way to choose P_1 relatively to P to make the w minimization convex and to have:

$$\inf_w E_{aux}(v,w) = E(v) \tag{3}$$

The hypothesis to obtain this result is either that $\frac{1}{2}\|N\|^2 - P(N)$ is convex, or, in the case $P = f(d)$ (d is a distance map to data points), that $(\frac{x^2}{2} - f(x))$ is convex. The consequence is that this auxiliary problem has the same solution in v as the original initial problem (1):

$$arg_v \inf_{(v,w)} E_{aux}(v,w) = arg \inf_v E(v) \tag{4}$$

This proves that the alternate minimization of E_{aux} with respect to v and w solves the same initial problem with energy E.

The use of auxiliary variables permits to transform a non convex problem in a two-step minimization where each step is easy to solve and convex. Our formulation permits a more precise setting of parameters in the iterative scheme to obtain the good hypothesis on P to ensure a good minimization.

The paper is organized as follows. We begin by a mathematical formulation of many data reconstruction problems (Section 2) and then introduce our two-variable energy minimization (Section 3). In Sections 4 and 5, we give the main mathematical study and results. We then apply this approach to give a new formulation for deformable templates, pattern matching, snakes and "spline-snakes" (Sections 6 to 9).

2 Curve and Surface Reconstruction

2.1 General formulation of Regularization

A general formulation of the curve or surface reconstruction problem, as presented in [9] uses Tikhonov regularization [10] to approximate data u by a smooth function v minimizing the following type of energy:

$$E(v,u) = \int R(v(s))ds + \int V(v(s),u(s))ds \tag{5}$$

where $R(v)$ measures the smoothness of the reconstruction v, and V is a measure of the distance between the function v and the data u. Usually R is the norm of a derivative of v or a combination of norms of different derivatives. The minimization is made on the space of functions for which $\int R(v)$ is well defined adding some constraints on the domain boundary (see [11] for a definition of Sobolev spaces). The first term is void if we impose the smoothness through a restriction on the shape. This is the case for deformable templates [8] (see Sec. 6) or spline-snakes [5] (see Sec. 9).

2.2 Attraction Potential. Examples

In [4], we give a survey of some reconstruction approaches. In particular, we make a distinction between explicit and implicit attraction and give physical interpretations in terms of zero length springs.

By explicit attraction, we mean that the second term $V(v,u)$ gives an explicit constraint on v through data u. For example, if $V(v,u) = \|v-u\|^2$, each point of the curve $v(s)$ is linked explicitly to a data point $u(s)$. The minimization of the energy:

$$E_u(v) = \int R(v(s))ds + \int \|v(s)-u(s)\|^2 ds \tag{6}$$

corresponds to a "classic" least-square reconstruction with explicit constraints of data u ([12, 1, 2]), this is a convex problem if R is quadratic. The minimum of E_u is a smoothed version of u.This is a simple problem since we already know in which order the unknown curve v has to pass by the data points u.

Also, in this case, all the data is relevant and represents a same shape, while in the implicit case, two tasks are solved simultaneously. At the same time the shape is reconstructed, a segmentation is applied on the data set to choose which points of the data belong to this shape and also to determine the order in which the curve has to match the data points. This is done through the use of implicit constraints. By implicit, we mean that V is defined through an attraction potential: $V(v, u) = P_u(v)$. This potential is designed in such a way that the low values of P correspond to points and features of interest. The more frequent examples of implicit attractions are

- $P = -\|\nabla I\|^2$ used in the "snake" model [3], where even the data points are not explicitly known and the constraint on v is imposed through the minimization of the potential P.
- $P = f(d)$ is a function of the distance d to the closest data point (see [4]), where a set of data points is known, as a binary image resulting from edge detection . The constraint is implicit since a point of v is linked to the closest data point. For a given point on v, this data point changes along iterations through the minimization process. Hence only part of the data is used and this can be seen as a simultaneous segmentation of the data at each iteration.

The energy to minimize is then:

$$E_P(v) = \int R(v(s))ds + \int P(v(s))ds. \tag{7}$$

The data term of the energy $\int V(v, u)$ or $\int P(v)$ is either an integral of functions defined for a continuous variable as in (5) or a finite sum $\sum_i V(v(s_i), u_i)$, $\sum_i \|v(s_i) - u_i\|^2$ or $\sum_i P(v(s_i))$ by discretization of s. In practical implementation, a discrete version of the energy is usually used.

Also, the data and its reconstruction represent either a curve or a 3D surface. We usually give examples with curves since they have a shorter formulation but we often use the word shape to mean either a curve or a surface since all our results are valid in any dimension.

We will detail currently used two-step iterative algorithms and their interpretation as a minimization of an auxiliary variable problem in Sections 6 to 9. The reader interested in one of these precise problems or their two-stage algorithms may immediately refer to these sections before reading our general presentation and mathematical results on the introduction of an auxiliary variable. We first define and study the two-variable energy minimization.

3 Introducing Auxiliary Variables

3.1 Motivation

For many shape reconstruction problems, two-stage iterative algorithms have been proposed (see for example [13, 5, 14, 15]), but without a precise link to the original energy minimization problem. In these algorithms, each iteration is composed of a small local deformation followed by some global regularization or best fit. This permits to separate the two problems of data segmentation and data fitting as explained in [4]. This is useful when all the data is not significant and at the same time the fit is done, some segmentation task operates to choose only part of the given data. The introduction of auxiliary variables is a mathematical justification of these algorithms.

To give a better mathematical formulation to these two-stage algorithms, we modify the energy by introducing an auxiliary variable in a way similar to [7] (described in the next section). The two stages are then interpreted as an alternate minimization to each of the two variables, the variable of the initial energy and the auxiliary variable. However, in our case we introduce an auxiliary variable in the data term instead of the regularization term. The difference comes from the fact that in our energy the data potential is not convex while the smoothing term is convex since we are interested in smooth shapes and do not need to introduce discontinuities.

This permits to transform an implicit data constraint defined by a non convex potential in an explicit reconstruction convex problem. This approach is much simpler since each iteration is composed of two simple to solve steps. The added variable is in most cases a current estimate of the shape to find.

3.2 Half-quadratic Regularization [7]

In a different context, where the regularization term is not convex and in order to make the energy minimization more efficient, some authors [7, 16] have introduced an auxiliary variable in the smoothing term R. In [7], the energy used is like Blake and Zisserman's one [17] to solve an image restoration problem where H is the point spread function, u is the blurred image and v the restored image:

$$E_{BZ}(v) = \int \sum_i f(\|\frac{\partial v}{\partial x_i}(x)\|)dx + \int \|Hv(x) - u(x)\|^2 dx \tag{8}$$

It contains a non convex bounded function of the derivatives $R(v) = \int \sum_i f(\|\frac{\partial v}{\partial x_i}(x)\|)dx$ to permit the introduction of discontinuities. Instead of using *Graduated Non Convexity* as in [17], the energy is transformed to solve a convex minimization. The auxiliary variable w is introduced as an estimate of the derivative ∇v. The new regularization term writes:

$$R_1(v, w) = \int \sum_i f_1(\|w_i(x)\|)dx + \frac{1}{2} \int \|\nabla v(x) - w(x)\|^2 dx \tag{9}$$

where f and f_1 form a Legendre pair (this means that they satisfy equation (62)). The variables $w_i(x)$ are the components of $w(x)$. The advantage of this formulation is that the energy becomes convex (and quadratic) in ∇v and minimization is solved analytically in w, for a given v. This gives a two-stage simpler algorithm by successive minimization of E with respect to v and w.

In our approach, the goal is opposite to the previous one. Indeed in [7], the auxiliary variable comes to present a new simpler algorithm to the problem of image restoration using stochastic minimization, while our work gives a better mathematical formulation and parameters setting to many two-stage deterministic algorithms that have already been used for solving reconstruction problems but without justification. A consequence of our present work is that even the formulation in [7] can be simplified since we show in Section 5.3.1 that in fact function f_1 does not need to be computed once f satisfies the convexity hypothesis. Note that a different kind of duality was also used to transform an energy for image restoration and regularization in [18, 19].

3.3 Energy with an auxiliary variable

A general formulation of the energy E_{aux} with the extra auxiliary variable w has this form:

$$E_{aux}(v, w, u) = \int R(v(s))ds + \int V_{aux}(v(s), w(s))ds + \int V_1(w(s), u(s))ds \tag{10}$$

where u is the data, v is the reconstructed shape as before and the middle term is added. Remark that the data term $\int V_1(w, u)$ relates the data u only to the auxiliary variable w and this variable in turn determines the shape v. This means that the auxiliary variable acts as a an intermediary between data u and shape v. (In Section 3.4, we will see that we have to use a different potential V_1 instead of V). The introduction of w is particularly useful when V is a potential defining an implicit constraint (see Sec. 2.2). This approach transforms the implicit constraints in an explicit problem through the use of an "explicit" potential $V_{aux}(v, w)$, for example $V_{aux}(v, w) = \frac{1}{2}\|v - w\|^2$. In the following, we will be interested in the typical form of this energy for most examples:

$$E_{aux}(v, w) = \int R(v(s))ds + \frac{1}{2}\int \|v(s) - w(s)\|^2 ds + \int P_1(w(s))ds \tag{11}$$

where the influence of the data u (defined implicitly by a potential P_1) on the unknown v is indirect in the second term through the use of variable w. The introduction of auxiliary variables permits to solve the indetermination of the implicit constraint. We will keep the auxiliary term coefficient $\frac{1}{2}$ in the following for simplicity but of course it could be replaced by $\frac{1}{2\tau}$ and we will mention how results are changed with the energy:

$$E_{aux}(v, w) = \int R(v(s))ds + \frac{1}{2\tau}\int \|v(s) - w(s)\|^2 ds + \int P_1(w(s))ds \tag{12}$$

We can see a geometric interpretation of the iterative alternate minimization of E_{aux} as a deformation of the current shape followed by a regularization. This corresponds to a separation between local deformation and global processing. Recall we are looking for a shape v that fits best the data u which is located at the smaller values of the potential P_1. Given a same initial estimate v_0 for v^0 and w^0, we iterate a two-stage process:

- shape v^n being fixed, the minimization of E_{aux} with respect to w is a problem that is made convex even if P_1 is not. An auxiliary shape w is deformed to be attracted by small values of P, and at the same time has to remain close to v^n. So v^n being given, w^n solves a trade-off between localization of features and small deformation from v^n. This corresponds to local deformation.

- auxiliary shape w^n being fixed, the minimizer of E_{aux} with respect to v, v^{n+1} is a regularized reconstruction of w^n taken as explicit data. This is a global processing. This permits to transform the second kind of implicit problem (7) into an explicit classic problem (6).

This geometric interpretation will be more precise in the examples given in Sections 6 to 9. In the discrete case, the shape is defined by parameters or nodes and the auxiliary variables w_i locate the position of auxiliary nodes. An illustration of the two-step approach is given in Fig. 1 where the potential P is a distance to the data set S.

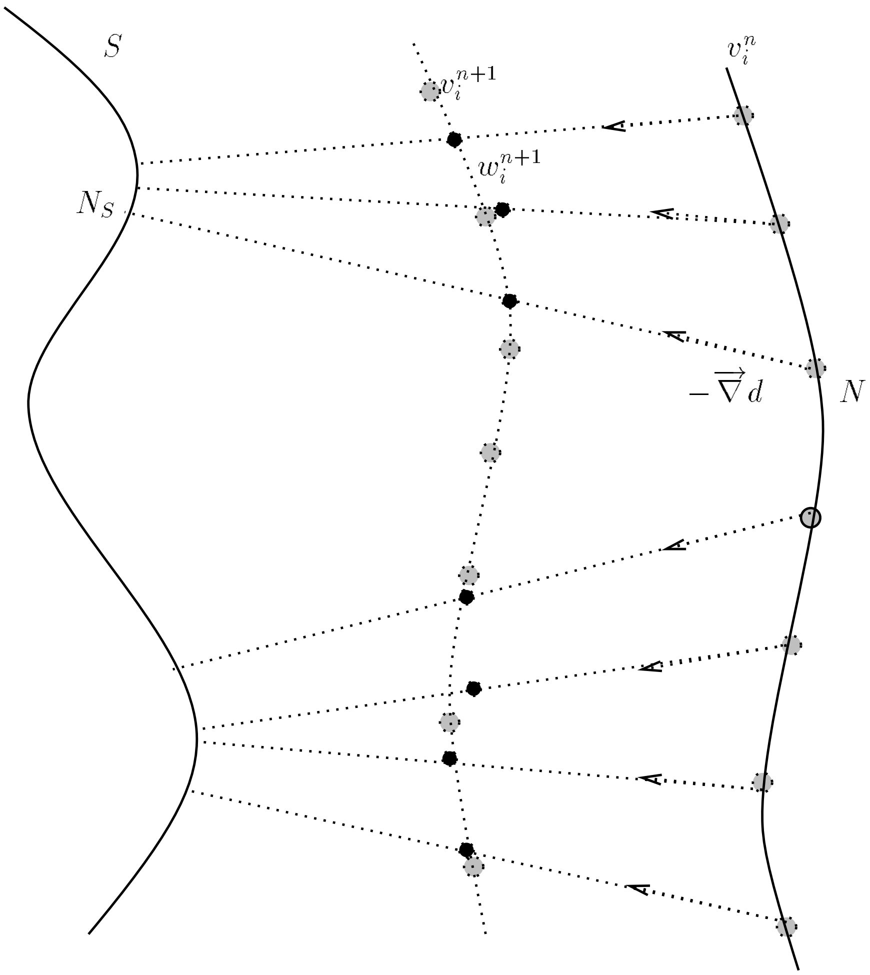

Figure 1: Illustration of one iteration of the w and v minimization. On the left is the data, on the right the current estimation of v_i's and in the middle the minimizing w_i's (black spots). The grey spots and the middle curve represent the new value of v_i's after regularization of the w_i's.

3.4 Solving the w minimization

Let us give a closer look at the minimization of E_{aux} in (11) with respect to w, with a chosen v. The first term of the energy $\int R(v)$ is then constant and we have to minimize only the second part of the energy:

$$E_2(v,w) = \frac{1}{2}\int \|v(s) - w(s)\|^2 ds + \int P_1(w(s))ds = \int \phi(v(s), w(s))ds \qquad (13)$$

where

$$\phi(N, M) = \frac{1}{2}\|N - M\|^2 + P_1(M). \tag{14}$$

M and N each represent a point of the 2D or 3D space and at the same time, for sake of brevity, the vector joining the origin to this point. In all that follows, N usually represents a point on the first curve v and M a point on the auxiliary curve w.

In the discrete case, this energy becomes:

$$E_2(v, w) = \frac{1}{2}\sum_{i=1}^{n} \|v(s_i) - w_i\|^2 + \sum_{i=1}^{n} P_1(w_i) = \sum_{i=1}^{n} \phi(v(s_i), w_i). \tag{15}$$

We see that we have to minimize $\phi(N, M)$ with respect to M for every N. If for given N, $\phi(N, M)$ is convex in M, there is only one minimum and the minimization of E_2 can be done independently for each point of the shape. As will be shown later, this hypothesis will be assumed on P_1. This simplifies the resolution of the implicit constraints since each $w(s)$ or w_i can be obtained as a function of $v(s)$ or v_i:

$$w_{min}(s) = \psi(v(s)) = ArgMin_M(\phi(v(s), M)) \tag{16}$$

and the minimum of ϕ is:

$$Q(v(s)) = \phi(v(s), \psi(v(s))). \tag{17}$$

The minimum of the energy becomes:

$$Min_w E_2(v, w) = E_2(v, \psi(v)) = \int Q(v(s))ds. \tag{18}$$

3.5 Successive minimization of E_{aux} and Two-step Algorithms

From Eqn. (18), it turns out that when we minimize the global energy $E_{aux}(v, w)$ with respect to w first and then v, the result in v is solution of the energy minimization E_Q analogous to the one of Eqn. (7) with a potential Q:

$$\begin{aligned}
\inf_{(v,w)} E_{aux}(v, w) &= \inf_v(\inf_w E_{aux}(v, w)) \\
&= \inf_v\{\int R(v) + \inf_w E_2(v, w))\} \\
&= \inf_v\{\int R(v) + \int Q(v(s))\} \\
&= \inf_v\{E_Q(v)\}
\end{aligned} \tag{19}$$

So by solving the auxiliary problem (11), we have found a way to solve the initial problem (7) but with a different potential Q instead of P. In the next sections, we find a condition on P which ensures the existence of a potential P_1 in E_{aux} such that the corresponding Q is equal to the initial potential P defining the interesting features, then by solving the minimization of E_{aux} we find exactly a solution of the initial problem of (7) minimizing E_P.

Although this alternate minimization does not ensure convergence to the global minimum, it likely converges to a local minimum. This is due to the fact that at each step, minimization in either v or w ensures a descent in energy. The use of stochastic minimization as used in [7] permits to obtain convergence to the global minimum.

Let us assume in this section that $Q = P$. We are then able to interpret two-step iterative schemes as the successive minimization of E_{aux} with respect to its two variables. Recall that:

$$E(v) = \int R(v(s))ds + \int P(v(s))ds \tag{20}$$

$$E_{aux}(v,w) = \int R(v(s))ds + \frac{1}{2}\int \|v(s) - w(s)\|^2 ds + \int P_1(w(s))ds \tag{21}$$

$$E_1(v,w) = \int R(v(s))ds + \frac{1}{2}\int \|v(s) - w(s)\|^2 ds \tag{22}$$

$$E_2(v,w) = \frac{1}{2}\int \|v(s) - w(s)\|^2 ds + \int P_1(w(s))ds \tag{23}$$

Assume that after n iterations you have the pair (v^n, w^n), then the next iteration is:

$$\left\{ \begin{array}{lll} \text{Step 1. Potential. Minimizing } E_2 & E_{aux}(v^n, w^{n+1}) & = \inf_w E_{aux}(v^n, w) = E(v^n) \\ \textit{Local deformation} & w^{n+1} & = \psi(v^n) \\ & & \\ \text{Step 2. Smoothing. Minimizing } E_1 & E_{aux}(v^{n+1}, w^{n+1}) & = \inf_v E_{aux}(v, w^{n+1}) \\ \textit{Global regularization} & v^{n+1} & = smooth(w^{n+1}) \end{array} \right. \tag{24}$$

It is obvious that E_{aux} is always decreasing from one half iteration to the next. From this we see that $E(v^n)$ is also decreasing since:

$$E(v^{n+1}) = E_{aux}(v^{n+1}, w^{n+2}) \le E_{aux}(v^{n+1}, w^{n+1}) \le E_{aux}(v^n, w^{n+1}) = E(v^n) \tag{25}$$

This works since the two steps can be interpreted as separate minimizations of the same energy E_{aux}. In the case P and P_1 do not satisfy the correct hypothesis and this formulation is not possible, there is no warranty for a descent in energy E after a single two-step iteration. Usually, in the minimization algorithms used, each half iteration makes one term of the energy E (the regularization term or the potential term) go down while the second part goes up. In general, there is no control on the balance of these two actions and if there is a negative balance, it will not converge. This corresponds to the same oscillations that were mentioned in [13] as instabilities due to image forces. When the potential satisfies the good hypotheses, there is always a descent in the global energy at each half iteration and this ensures convergence of the energy to a minimum. Using the same approach as in [20], this could imply that $(v^{n+1} - v^n)$ tends to 0 and that the limit set is connex. The two-step algorithm is illustrated in Section 8 in Figures 5 and 6.

Remark 1 *In the case the potential P is separated in n terms $P = \sum_{i=1}^{n} P^i$, then n auxiliary variables are introduced and the algorithm is made with $n+1$ steps. This may be the case for data fusion or for including various kinds of constraints on the reconstruction. Each term $P^i(v)$ is replaced in the energy by $\frac{1}{2}\|v - w_i\|^2 + P_1^i(w_i)$. Minimizing in each w_i gives n steps of local deformation according to the different features P^i. Minimizing in v operates a global smoothing on all the different estimates w_i. All the results in the following remain true providing all P^i satisfy the demanded hypotheses. Of course, these terms can have different weights as mentioned in Eqn. (12) and remark 6.*

Remark 2 *Although in the examples we only use quadratic regularization terms, we can generalize our approach in the case when both regularization and potential are not convex by introducing two auxiliary variables. This corresponds to reconstruction with discontinuities in the case of non convex attraction potential mixing energies of (7) and (8):*

$$E_{disc}(v) = \int \sum_i f(\|\frac{\partial v}{\partial x_i}(x)\|)dx + \int P(v(x))dx \tag{26}$$

This energy is then replaced by a combination of (9) and (11).

$$E_{aux}(v, w, \tilde{w}) = \int \sum_i f_1(\|w_i(x)\|)dx + \frac{1}{2}\int \|\nabla v(x) - w(x)\|^2 dx + \frac{1}{2}\int \|v(s) - \tilde{w}(s)\|^2 ds + \int P_1(\tilde{w}(s))ds \tag{27}$$

Minimization in $\tilde{w}$ gives a local deformation of the shape according to the potential. Minimization in w is a local deformation of the gradient estimate to detect discontinuities. Minimization in v is a global quadratic regularization taking into account a least square error with given shape $\tilde{w}$ and given gradient w. If both f_1 and P_1 satisfy the required hypotheses, alternate minimization of (27) will give a simple algorithm for minimization of (26).

4 Resolution using conjugate functions

Under some assumptions on a given P, we find a potential P_1 to obtain $Q = P$. For this we recall the definition of the Conjugate function (see [11, 21, 22]), also called Legendre transform.

4.1 Conjugate functions

Definition 1 *If φ is a function from a Euclidean space $\mathcal{E}$ to $I\!R$, the conjugate function of φ is*

$$\varphi^*(u) = \sup_{v \in \mathcal{E}}((u, v) - \varphi(v)) \tag{28}$$

Remark 3 *Here φ and φ^* may take infinite values but if we assume φ is not infinite everywhere, then so is φ^*. In the case φ is finite and $\lim_{\|v\| \to \infty} \dfrac{\varphi(v)}{\|v\|} = \infty$, φ^* is finite everywhere. Function φ^* is always convex.*

We have the following properties:

Theorem 1 *if φ is convex*

$$\varphi^{**} = \varphi \tag{29}$$

Moreover, if φ is strictly convex and C^2, then so is φ^* and we have equivalence of the following properties:

(a) The value $\varphi^*(u)$ is reached in (28) for u^*: $\varphi^*(u) = (u, u^*) - \varphi(u^*)$,

(b) $u = D\varphi(u^*)$,

(c) $u^* = D\varphi^*(u)$,

(d) The value $(\varphi^*)^*(u^*)$ is reached in (28) for u: $(\varphi^*)^*(u^*) = (u^*, u) - \varphi^*(u)$.

Remark 4 *By definition of φ^*, we have for all u and v: $(u, v) \le \varphi(v) + \varphi^*(u)$ and if there is equality then v reaches the maximum value $\varphi^*(u)$ in (28). Note that (b) and (c) mean that $D\varphi$ and $D\varphi^*$ are inverse.*

Proof. Different proofs for $\varphi^{**} = \varphi$ can be found in [11, 21, 22] when $\mathcal{E}$ is not assumed Euclidian.

It is simple to see that (a) implies (b) since (b) says that the differential of $(u, .) - \varphi(.)$ is zero at a maximum u^*. Since φ is convex, $(u, .) - \varphi(.)$ is concave, so if we have (b), u^* reaches the maximum, that is (a).

Assuming (a), $\varphi^*(u) = (u, u^*) - \varphi(u^*)$, we have $\varphi^{**}(u^*) = \varphi(u^*) = (u, u^*) - \varphi^*(u) = (u^*, u) - \varphi^*(u)$. This means that we have (d). The same way, (d) implies (a).

Since there is strict convexity, the maximum is unique and, assuming finite dimension for $\mathcal{E}$ ($\mathbb{R}^2$ or $\mathbb{R}^3$ in our case), $D^2\varphi$ is positive definite and thus invertible everywhere. By inversion theorem on (b), we deduce that $u^*(u) = D\varphi^*(u)$ is C^1 and thus φ^* is also C^2 where it is reached. In fact, we use these hypotheses for simplicity but a more general proof is also valid in the case φ is only C^1.

Now that φ^* is C^1, (c) and (d) are equivalent the same way as (a) and (b) for φ^*.

We showed that (a) to (d) are equivalent, if φ^* was not strictly convex, there would be a value u^* for which the conjugate value $(\varphi^*)^*(u^*)$ is reached for at least u_1 and u_2, which implies using (b) that $u_1 = u_2 = D\varphi(u^*)$. ■

4.2 Finding P_1

We now give a relation between P_1 and Q using the definitions of ϕ and ψ (Eqns. (14) and ff.):

$$\phi(N, M) = \frac{1}{2}\|N - M\|^2 + P_1(M) = \frac{1}{2}\|N\|^2 - (N, M) + \frac{1}{2}\|M\|^2 + P_1(M) \tag{30}$$

A point N being given, we have:

$$Q(N) = \inf_M \phi(N, M) = \frac{1}{2}\|N\|^2 - \sup_M\{(N, M) - (\frac{1}{2}\|M\|^2 + P_1(M))\} \tag{31}$$

$$Q(N) = \frac{1}{2}\|N\|^2 - P_2^*(N) \tag{32}$$

where

$$P_2(M) = \frac{1}{2}\|M\|^2 + P_1(M). \tag{33}$$

For Q and P to be equal, we must have

$$P_2^*(N) = \frac{1}{2}\|N\|^2 - P(N). \tag{34}$$

It follows from these properties that

Theorem 2 *If $\varphi(N) = \frac{1}{2}\|N\|^2 - P(N)$ is convex, we have $Q = P$ with P_1 defined by equality:*

$$\frac{1}{2}\|M\|^2 + P_1(M) = \varphi^*(M) = \{\frac{1}{2}\|N\|^2 - P(N)\}^*(M) \tag{35}$$

Remark 5 *Moreover, under these assumptions, since a conjugate function is convex, it follows from Eqns. (30) and (35) that ϕ is convex with respect to M and that the minimization has a unique solution. Since P is not convex in general, this shows a way to define an auxiliary problem which is convex. In fact both steps of the minimization are convex since E_1 and E_2 of Eqns. (22) and (23) are then convex with respect to both variables.*

Remark 6 *Although the hypothesis that $(\frac{1}{2}\|N\|^2 - P(N))$ is convex may be false, we may find more easily a constant α such that $\frac{\alpha}{2}\|N\|^2 - P(N) = \alpha(\frac{1}{2}\|N\|^2 - \frac{1}{\alpha}P(N))$ is convex. In this case we use the auxiliary problem (12) with $\tau = \frac{1}{\alpha}$ and in all hypothesis P has to be replaced by τP and P_1 by τP_1.*

Remark that the solution ψ of the minimization defined by (16):

$$Q(N) = \inf_M \phi(N, M) = \phi(N, \psi(N)). \tag{36}$$

corresponds to the argument M obtained for calculation of $P_2^*(N)$ in (28).

Usually, the determination of the corresponding P_1, for a given P satisfying the hypothesis of convexity, is not analytic and may be found through numerical computation. However, to implement the algorithm, we only need the argument of the minimization ψ. We show in the next sections how to calculate this function.

4.3 Approximation by look-up table

A point N being given, let us see how we can determine a minimum M of function $\phi(N, M)$ as a zero of its $M-$differential:

$$(N - M) - \nabla P_1(M) = 0 \tag{37}$$

The first term is a zero length spring attraction force towards N applied at M. The second is the force derived from the potential P_1, this is the steepest descent direction in P_1. When M is solution of the minimization, it reaches an equilibrium of these forces.

We have a simple estimate of the inverse of ψ by defining:

$$\mu(M) = M + \nabla P_1(M) \tag{38}$$

We are looking for a point M such that $\mu(M) = N$. We can simply make a table of these values once for all and obtain a good candidate for $\psi(N)$ by interpolation between the closest reached values or global interpolation of data obtained. When many M give the same $\mu(M)$, we have to choose the one that has the lowest value of Q or the greatest norm of ∇P, this is also the one that gives the greatest displacement.

In fact, we show in the next section that we can usually invert (38).

4.4 Explicit resolution

Actually, we now show that since to implement the descent algorithm in (24) only ψ is needed, computation of P_1 is in fact not necessary once you know it exists and is regular. Since in general P is known only on a discrete grid, we may assume that we work with a smooth interpolation of these discrete values unless there is a reason to have discontinuities in P.

Theorem 3 *Under the hypotheses of Theorem 2, if P is C^2 and $\frac{1}{2}\|M\|^2 - P(M)$ is strictly convex, then the local deformation is $\psi(N) = N - \nabla P(N)$.*

Remark 7 *In the case of energy (12) the hypothesis is with τP and $\psi(N) = N - \tau\nabla P(N)$. The reason we called it τ now appears since it is similar to a time step.*

Remark 8 *This shows that the new potential P_1 is chosen such that the minimization with respect to the auxiliary variable gives the same result as the gradient descent in P with time step 1. This is related to the fact that when P satisfies our hypothesis, the second differential is such that $\|D^2P\| \leq 1$ and the time step in the Jacobi descent algorithm for P is 1 (as explained in [21] or [17, Sec. 7.5]). However, in our case, the descent is ensured for the sum of terms and not only for the potential one. This permits to interpret the step of usual gradient descent as a minimization of the auxiliary energy E_{aux} with respect to the added variable.*

Proof. Denoting $P_2(M) = \frac{1}{2}\|M\|^2 + P_1(M)$, we first apply theorem 1 to $\varphi(N) = \frac{1}{2}\|N\|^2 - P(N)$ proving that P_2 is strictly convex and C^2 from our hypothesis and theorem 2 since $P_2 = \varphi^*$.

Recall that $\psi(N)$ minimizes $\phi(N, M) = \frac{1}{2}\|N - M\|^2 + P_1(M)$ with respect to M. From Eqn. (31), we see that $N^* = \psi(N)$ is the argument which reaches the value of the conjugate $P_2^*(N)$. It is unique since there is strict convexity. Applying the last part of theorem 1 to P_2, we deduce that the point N^* which reaches the value of the conjugate $P_2^*(N)$ is such that $DP_2(N^*) = N$ which is equivalent to $N^* = DP_2^*(N)$. Then since we have $P_2^*(N) = \frac{1}{2}\|N\|^2 - P(N)$, we conclude that $N^* = N - DP(N)$ which gives the result (∇P is the vector notation for DP). ■

Since the condition on P is not easy to check for a given potential, we now give a special case of potential for which we have found an expression of P_1. Moreover, in this case the condition will be weaker and easier to check than that on P.

5 Explicit resolution when $P = f(d)$

In the case of a potential P defined as a function $f(d)$ of the distance to the closest data point, we show that we can find $P_1 = f_1(d)$ of the same kind to retrieve the same energy by the two-stage algorithm. The results do not depend on the dimension of the space where we consider our objects (2-D or 3-D). This kind of potential is used more and more often in shape reconstruction problems. One reason is the easy way to compute the Chamfer distance that approximates the Euclidian distance to the closest point in a set [23, 24]. We show in figure 2 an example of edge data and distance map.

5.1 Defining the distance potential

We now assume that $P = f(d)$ is an attraction potential as defined in [13]. The force deriving from the potential attracts the curve towards already detected contour pixels (edgels). The potential is an increasing continuous function of the distance to the closest contour in the set S of contour points. For a point M and a set S in the image space, we define the distance to S by:

$$d_S(M) = d(M, S) = \inf_{N \in S} d(M, N) \tag{39}$$

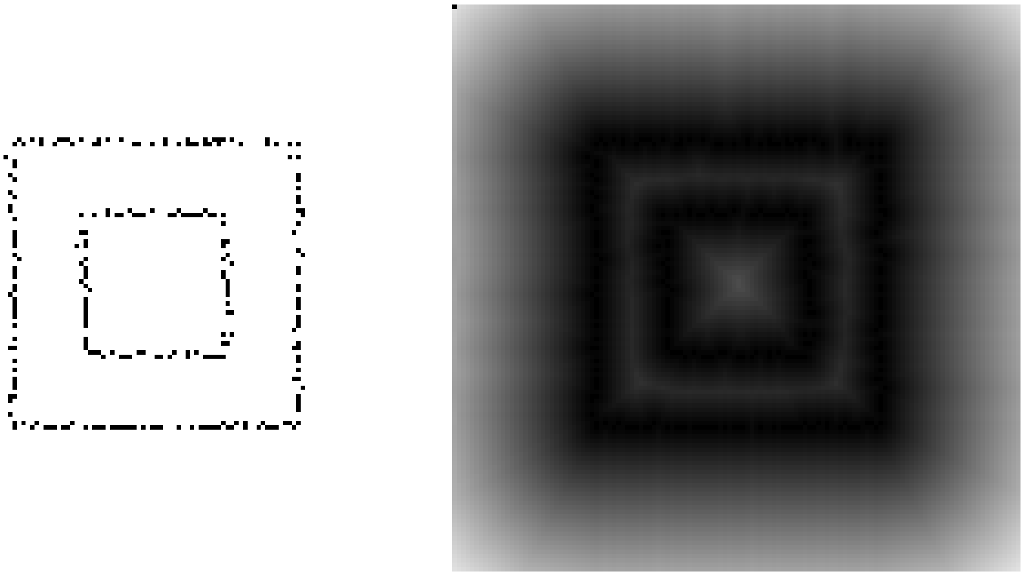

Figure 2: A noisy edge image of two squares and its distance map (black for $d_S = 0$).

and the potential is

$$P(M) = f(d_S(M)) \tag{40}$$

Usually we have $f(d) = \alpha \frac{d^2}{2}$ or as explained in [4], it can be useful to define f with a returned bell shape like $f(d) = -e^{-d^2}$ or a threshold function $f(d) = \inf(\alpha \frac{d^2}{2}, \beta)$. This permits to avoid influence of outlyers by giving the same contribution in the energy to points which are far enough to the data. When the distance to closest data is large enough, it means that there is no match between this point and data and it is meaningful to consider its contribution to the energy only as a penalty β which does not depend on the distance. This means that only the more likely reliable data is used. This is a different formulation for robust statistics (see for example [25]) as can also be seen from the end of [26] or in [27].

In the auxiliary energy we use a different potential P_1 of the same kind with an increasing continuous function f_1:

$$P_1(M) = f_1(d_S(M)) \tag{41}$$

Since the f and f_1 are useful only at positive values, we can assume they are both even continuous functions that are increasing on $\mathbb{R}^+$.

Since for practical implementation, S is a finite set of segments, we can assume that S is a closed set. This ensures that for M in the image space, there is at least one point M_S in S which reaches the minimal distance to S. We will call such a point a **projection point**. A point of S is its own projection point. This projection point M_S is uniquely defined almost everywhere. The set of points for which M_S is not unique is called the skeleton of S.

5.2 Properties. Preparatory Lemmas

Lemma 1 *For N in the image space, if M minimizes*

$$\phi(N,M) = \frac{1}{2}\|N-M\|^2 + f_1(d_S(M)) \tag{42}$$

and if M_S is a projection point of M, then N, M and M_S are aligned.

Proof. This is illustrated by Fig. 3. To simplify the proof, we assume $f_1(0) = 0$, but this is not necessary, only the fact that f is increasing is used. Suppose M is a minimum of $\phi(N,.)$ and M_S, N_S are projection points of M, N respectively, then by definition of ϕ:

$$\frac{1}{2}\|N-M\|^2 \leq \phi(N,M) \leq \phi(N,N_S) = \frac{1}{2}\|N-N_S\|^2 = \frac{1}{2}d_S(N)^2 \tag{43}$$

and

$$f_1(d_S(M)) \leq \phi(N,M) \leq \phi(N,N) = f_1(d_S(N)) \tag{44}$$

and since f_1 is increasing,

$$d_S(M) \leq d_S(N) \tag{45}$$

We have also from (43):

$$\|N-M\|^2 \leq d_S(N)^2 \leq d(N,M_S)^2 \tag{46}$$

Now, if M_1 is the orthogonal projection of M on the line (N, M_S), then M_1 is between N and M_S and

$$d_S(M_1) \leq d(M_1, M_S) \leq d(M, M_S) = d_S(M) \tag{47}$$

and we deduce

$$\phi(N,M_1) = \frac{1}{2}\|N-M_1\|^2 + f_1(d_S(M_1)) \leq \frac{1}{2}\|N-M\|^2 + f_1(d_S(M)) = \phi(N,M) \tag{48}$$

Since M is a minimum, this last inequality is an equality. This is possible only if $M = M_1$. So M is on the line (N, M_S). ■

Remark 9 *We see by the way that if M is different from N, M_S is unique.*

Remark 10 *We could also use Eqn. (37) to prove the Lemma, but preferred a geometric proof to avoid the fact that ∇P_1 is not defined everywhere. Indeed, we have $\nabla P_1 = f_1'(d_S)\nabla d_S$ and ∇d_S is defined everywhere except on the skeleton of S. At a point M outside the skeleton, there is only one closest point M_S in S and ∇d_S is the unit vector in the direction of $(M_S M)$ so that the force $F = -\nabla P$ is an attraction force to the closest data point (see Fig. 1).*

Lemma 2 *With the same notations, M_S is also a projection point for N, $d_S(N) = d(N, M_S)$ and on the other side, there is a minimizer M_2 of $\phi(N,.)$ on the line (NN_S).*

Remark 11 *If N is not on the skeleton, its projection point is unique and we have $N_S = M_S$.*

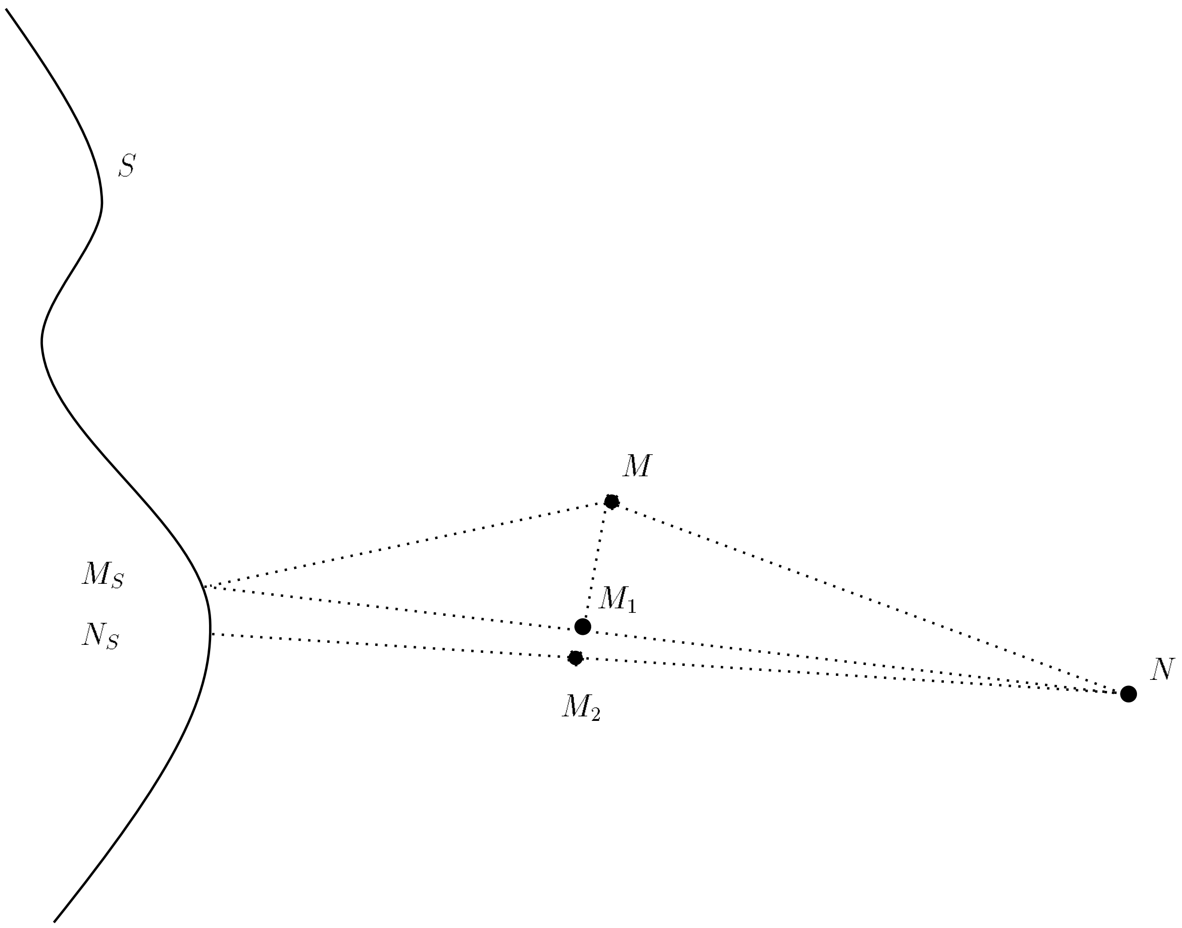

Figure 3: Illustration for Lemmas 1 and 2.

Proof. Using Eqn. (43), we can define the point M_2 (see Fig. 3) on the segment $[N, N_S]$ such that

$$d(M_2, N) = d(M, N) \tag{49}$$

We then have using Lemma 1:

$$d(N_S, M_2) + d(M_2, N) = d(N_S, N) \leq d(M_S, N) = d(M_S, M) + d(M, N) \tag{50}$$

and

$$d_S(M_2) \leq d(N_S, M_2) \leq d(M_S, M) = d_S(M) \tag{51}$$

We now see that since M is a minimum of $\phi(N, .)$, M_2 is also a solution :

$$\phi(N, M_2) = \frac{1}{2}\|N - M_2\|^2 + f_1(d_S(M_2)) \leq \frac{1}{2}\|N - M\|^2 + f_1(d_S(M)) = \phi(N, M) \tag{52}$$

Thus, this is an equality and we have:

$$d_S(M_2) = d(N_S, M_2) = d(M_S, M) \tag{53}$$

and

$$d_S(N) = d(N_S, N) = d(M_S, N) \tag{54}$$

and M_S is also a projection point for N. ■

Lemma 3 *Defining $f_2(\lambda) = \frac{\lambda^2}{2} + f_1(\lambda)$, for N in the image space we have*

$$\inf_M \{\frac{1}{2}\|N - M\|^2 + f_1(d_S(M))\} = \frac{1}{2} d_S(N)^2 - f_2^*(d_S(N)) \tag{55}$$

Proof. The point N being given, since the quantity $\phi(N, M) = \frac{1}{2}\|N - M\|^2 + f_1(d_S(M))$ is minored by $f_1(0)$ and tends to infinity at infinity, there is a value M which reaches the minimum. Lemmas 1 and 2 show that there is a projection point N_S (such that $d_S(N) = d(N_S, N)$) for which M is in the segment $[N_S, N]$. Thus the minimum of ϕ is the same as the minimum on this segment:

$$\inf_M \phi(N, M) = \inf_{\lambda \in [0, d_S(N)]} \{\frac{\lambda^2}{2} + f_1(d_S(N) - \lambda)\} = \inf_{\lambda \in [0, d_S(N)]} \{\frac{1}{2}(d_S(N) - \lambda)^2 + f_1(\lambda)\} \tag{56}$$

Let us define

$$g(d, \lambda) = \frac{1}{2}(d - \lambda)^2 + f_1(\lambda). \tag{57}$$

This is in fact the real variable version of ϕ. The problem has thus been transferred to one dimension. We remark that

$$g(d, 0) = \frac{1}{2}d^2 + f_1(0) = \inf_{\lambda \in (-\infty, 0]} g(d, \lambda) \tag{58}$$

and

$$g(d, d) = f_1(d) = \inf_{\lambda \in [d, \infty)} g(d, \lambda) \tag{59}$$

We then have similar to Eqns. (31) and (32) in Section 4:

$$\begin{aligned} \inf_{\lambda \in [0, d]} g(d, \lambda) &= \inf_{\lambda \in \mathbb{R}} g(d, \lambda) = \frac{1}{2}d^2 - \sup_{\lambda \in \mathbb{R}} \{d\lambda - (\frac{\lambda^2}{2} + f_1(\lambda))\} \\ &= \tfrac{1}{2}d^2 - f_2^*(d) \end{aligned} \tag{60}$$

which proves the lemma using $d = d_S(N)$. ■

Remark 12 *We deduce from these last two lemmas that we can always find a minimizer of $\phi(N,.)$ on the segment linking N and a projection point N_S, we can thus define a solution $\psi(N)$ (16) by:*

$$\psi(N) = N - l(d_S(N))\nabla d_S(N) \tag{61}$$

where $0 \leq l(d) \leq d$, since, as noted in remark 10, $\nabla d(N)$ is the unit vector in the direction $(N_S N)$. Remark that function l depends only on f and not on the data. We have $l(d) = d - \lambda^*(d)$ where $\lambda^*(d)$ is the value reaching the minimum in λ of $g(d,\lambda)$ in Eq. (57). Hence the determination of function l can be done once for all and gives the way to calculate the second step of the algorithm.

5.3 Main Theorem. Examples

Using the previous lemmas and the notations of section 3.4, we have now shown that in the case $P_1(M) = f_1(d_S(M))$, theorem 2 becomes:

Theorem 4 *If function $\varphi(x) = (\frac{x^2}{2} - f(x))$ is convex, defining f_1 by equality:*

$$(\frac{\lambda^2}{2} + f_1(\lambda)) = \varphi^*(\lambda) = \{\frac{1}{2}x^2 - f(x)\}^*(\lambda) \tag{62}$$

we have $Q(M) = P(M) = f(d_S(M))$.

Remark 13 *In case we have the hypothesis $(\frac{x^2}{2} - \tau f(x))$ convex, f_1 in problem (12) is then defined by $(\frac{\lambda^2}{2} + \tau f_1(\lambda)) = \{\frac{1}{2}x^2 - \tau f(x)\}^*$*

Proof. The only thing to check is that this definition of f_1 gives an even function increasing on $I\!R^+$. From (62), we have $f_1(\lambda) = \sup_x(f(x) - \frac{(\lambda+x)^2}{2}) = \sup_x(f(x) - \frac{(\lambda-x)^2}{2}) = f_1(-\lambda)$ since f was assumed even from the beginning. We also have $f_1(0) = 0$ using the convexity hypothesis. If $\lambda > 0$, it can be seen since f is increasing and even that $f_1(\lambda) = \sup_{x>0}(f(x+\lambda) - \frac{(x)^2}{2})$ which is an increasing expression of λ for all $x > 0$. This can also be seen from the sign of f_1' in Eqn. (66). ■

This has advantage on Theorem 2 that the computation of the conjugate in Eqn. (35) is now for a function of one real variable instead of two or three. We reduced the problem from any dimension to one dimension. Also for a given function f, f_1 does not depend on the data defined by $d_S(M)$. It has to be computed only once for all, and a precomputed pair (f, f_1) can be used for all problems. As for Theorem 2, it follows from Eqn. (62) that function $(\frac{\lambda^2}{2} + f_1(\lambda))$ is convex and $g(d,\lambda)$ is convex with respect to λ. It follows from the lemma that the minimizer $\psi(N)$ of $\phi(N,M)$ is obtained as the argument in the conjugate calculation. Note also that f may satisfy this condition while $P = f(d_S)$ does not satisfy the condition of Theorem 2 since d_S is not convex. This means that the condition here is a lot more general than the previous one.

5.3.1 Explicit resolution for $\psi(N)$

In a way similar to section 4.4, we give an explicit expression of $\psi(N)$ relatively to f which does not need computation of f_1.

Theorem 5 *Under the hypotheses of Theorem 4, if function $\frac{1}{2}x^2 - f(x)$ is strictly convex and C^2, then $\psi(N) = N - f'(d_S(N))\nabla d_S(N)$.*

Note that remark 8 still applies here.

Remark 14 *With the hypothesis of remark 13 we replace f by τf and have $\psi(N) = N - \tau f'(d_S(N))\nabla d_S(N)$.*

Remark 15 *At points where $\nabla d_S(N)$ is not defined, it is replaced by equality $d_S(N)\nabla d_S(N) = \overrightarrow{N_S N}$ with a choice of a closest point N_S. This case of potential is much simpler since our results apply directly to the function f which is independent of the data defined by d_S.*

Proof. This proof is simpler in the case $P = f(d_S)$, since we deal with real variable functions and derivatives instead of differentials. Also, the proof depends only on f and not on data defined by d_S. This is important since it is clear that on points of the skeleton, P cannot be assumed C^2 (see remark 10) to apply the proof of section 4.4, while f may be assumed even C^∞ without loss of generality. If $\lambda^*(d)$ minimizes $g(d, \lambda) = \frac{1}{2}(d-\lambda)^2 + f_1(\lambda)$ with respect to λ, we see from Eqn. (60) that $\lambda^*(d)$ is the argument which reaches the conjugate value of $f_2^*(d)$. Denoting now $\varphi(x) = \frac{1}{2}x^2 - f(x)$, we deduce from our hypotheses and theorem 1 that f_2 is strictly convex and C^2 since $f_2 = \varphi^*$. Using now theorem 1 for f_2, we see that since $\lambda^*(d)$ reaches the value $f_2^*(d)$, we have $f_2'(\lambda^*(d)) = d$ and $\lambda^*(d) = (f_2^*)'(d) = \varphi'(d) = d - f'(d)$. Now from remark 12, we have a minimizer $\psi(N) = N - l(d_S(N))\nabla d_S(N)$, with $l(d_S(N)) = d_S(N) - \lambda^*(d_S(N)) = f'(d_S(N))$ which is the result of the theorem.

We can find the same result directly without using the end of theorem 1, reminding that:

$$f(d) = \inf_\lambda g(d, \lambda) = g(d, \lambda^*) = \frac{1}{2}(d - \lambda^*)^2 + f_1(\lambda^*) \tag{63}$$

From the fact that the minimum is reached at λ^*, we have

$$(\lambda^* - d) + f_1'(\lambda^*) = 0 \tag{64}$$

and deriving the equality with respect to d (λ^* is C^1 in d from our hypotheses and using inversion theorem on the previous equation) we find:

$$f'(d) = (d - \lambda^*)(1 - \lambda^{*\prime}(d)) + f_1'(\lambda^*)\lambda^{*\prime}(d) \tag{65}$$

now replacing $f_1'(\lambda^*) = -(\lambda^* - d)$ from Eqn. (64), we obtain the good value of λ^* for every d:

$$f'(d) = (d - \lambda^*) = f_1'(\lambda^*) \qquad \blacksquare \tag{66}$$

5.3.2 Example 1

In the simple case where $f(d) = \alpha\frac{d^2}{2}, 0 < \alpha < 1$, $(\frac{d^2}{2} - f(d))$ is convex and f_1 is also quadratic using the formula:

$$(\alpha\lambda^2)^* = \frac{d^2}{4\alpha} \tag{67}$$

and we have $f_1(\lambda) = \frac{\alpha}{1-\alpha}\frac{\lambda^2}{2}$. We can see that the function $l(d) = \alpha d$, and we verify that we have $l(d) = f'(d)$. Eqn. (61) becomes

$$\psi(N) = (1-\alpha)N + \alpha N_S \tag{68}$$

since $d_S(N)\nabla d_S(N) = \overrightarrow{N_S N}$. Hence instead of replacing N by the closest data point N_S, a point on the segment linking N and N_S is taken. Although this seems less precise than taking N_S, it is more stable and ensures better convergence and should avoid oscillations around the data.

However, we can also take N_S in the limit case $\alpha = 1$. The function $(\frac{d^2}{2} - f(d)) = 0$ is still convex and $f_1(\lambda) = \infty$ everywhere except $f_1(0) = 0$. Theorem 4 still applies. In this case the new value is exactly the closest point. Although it is more precise in the data fitting it may be more sensitive to spurious data points and discrete computation of d_S. The greater α, the faster the curve progression in the very first iterations. However when α is close to 1, the numerical discretization may cause oscillations while smaller values will converge monotonously as expected.

5.3.3 Example 2

If $f(d) = Inf(\beta, \alpha\frac{d^2}{2}), 0 < \alpha < 1$, then $(\frac{d^2}{2} - f(d)) = Sup(\frac{d^2}{2} - \beta, (1-\alpha)\frac{d^2}{2})$ is convex and we can show that we have a closed form formula for f_1:

$$f_1(\lambda) = \begin{cases} \frac{\alpha}{1-\alpha}\frac{\lambda^2}{2} & if \quad 0 \le \lambda \le (1-\alpha)\sqrt{\frac{2\beta}{\alpha}} \\ \beta - \frac{1}{2}(\lambda - \sqrt{\frac{2\beta}{\alpha}})^2 & if \quad (1-\alpha)\sqrt{\frac{2\beta}{\alpha}} \le \lambda \le \sqrt{\frac{2\beta}{\alpha}} \\ \beta & if \quad \lambda > \sqrt{\frac{2\beta}{\alpha}} \end{cases} \tag{69}$$

Functions f and f_1 are represented in Fig. 4.

Remark 16 *We see how the threshold function usually used in robust estimation is transformed by this approach in a function of the kind of Blake and Zisserman's functions used for GNC $g^{(1/2)}_{\beta,\sqrt{\frac{\alpha}{2(1-\alpha)}}}$ in [17, Fig. 7.1]. Since $f_1 = g^{(1/2)}_{\beta,\sqrt{\frac{\alpha}{2(1-\alpha)}}}$, this means that if we add an auxiliary term to their energy:*

$$E_{gnc}(u) = \sum_{i,j=1}^{N}(d_{ij} - u_{ij})^2 + \sum_{i,j=1}^{N} f_1(u_{ij} - u_{i-1,j}) + \sum_{i,j=1}^{N} f_1(u_{ij} - u_{i,j-1}) \tag{70}$$

it becomes

$$E^{aux}_{gnc}(u) = \sum_{i,j=1}^{N}(d_{ij} - u_{ij})^2 + \sum_{i,j=1}^{N}(f_1(b^x_{ij}) + \frac{1}{2}(u_{ij} - u_{i-1,j})^2) + \sum_{i,j=1}^{N}(f_1(b^y_{ij}) + \frac{1}{2}(u_{ij} - u_{i,j-1})^2) \tag{71}$$

and minimization in b^x, b^y gives back the original energy before use of GNC:

$$E_f(u) = \sum_{i,j=1}^{N} (d_{ij} - u_{ij})^2 + \sum_{i,j=1}^{N} f(u_{ij} - u_{i-1,j}) + \sum_{i,j=1}^{N} f(u_{ij} - u_{i,j-1}) \tag{72}$$

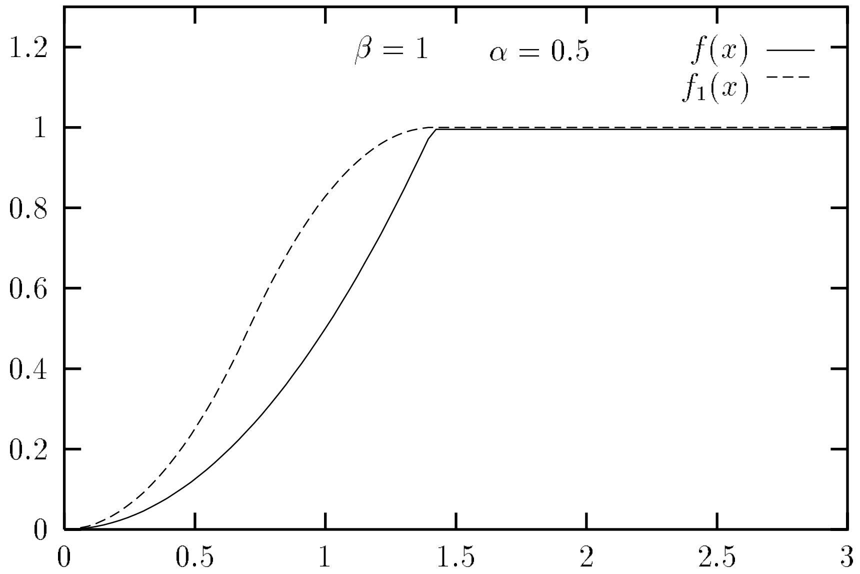

Figure 4: Functions f and f_1 in example 2.

Remark that the example given in [7] corresponds to the limit case of convexity when $\alpha = 1$ and the first expression in Eqn. (69) is void.

5.3.4 Other examples

We can find other examples of pairs (f, f_1) similar to those in [7]. An example of such a function which has the same "returned bell" shape as the one used in [4] is

$$f(d) = \frac{d^2}{(1+d^2)} = 1 - \frac{1}{(1+d^2)} \tag{73}$$

In such a case, f_1 has no closed-form expression but a numerical representation of f_1 can be found.

The approach in [7] is that since in fact f_1 is the function that is used in their algorithm, f_1 is chosen such that $(f_1(\lambda) + \frac{\lambda^2}{2})$ is convex and then f and l are found (numerically or by a closed form expression) using Eqn. (62). For example, the function equivalent to that used in [7] is

$$f_1(\lambda) = \frac{\lambda}{(1+\lambda)}. \tag{74}$$

The correspondent f also have a bell shape, but is known only numerically. From what we showed in section 5.3.1, it turns out that such an approach is not necessary since once f satisfies the hypothesis of Theorem 4, you need only f to implement the iterative algorithm.

6 Deformable Templates

We give here a result with a general formulation of parametric deformable templates.

6.1 Definition

Deformable Templates can be used with various simple regular shapes defined by a small number of parameters like circles, ellipses and parabola arcs [8], superquadrics [28] and hyperquadrics [29, 14]. Since the shape is imposed and regular, the smoothing term is not necessary. The data term is often an attraction potential summed either on the nodes of a parameterization or using the implicit equation of the shape. The unknown $\mathcal{A}$ is now a small set of parameters defining the shape $\mathcal{S}_\mathcal{A}$ and the energy is of the following form:

$$E(\mathcal{A}) = \int_{\mathcal{S}_\mathcal{A}} P \tag{75}$$

We are interested in the discrete form:

$$E(\mathcal{A}) = \sum_i P(v_i(\mathcal{A})) \tag{76}$$

where $(v_i(\mathcal{A}))_i$ is a set of nodes which discretize the shape $\mathcal{S}_\mathcal{A}$.

6.2 General two-step algorithm

A two-step approach is defined by a global model fitting followed by a local deformation of the set of nodes. Beginning with initialization of either a model $\mathcal{A}^0$ or a set of points $\mathcal{M}^0$, we iterate the following two steps:

Step 1 : **Model Fitting**. A set $\mathcal{M}^k = (M_i^k)_{1\leq i\leq n}$ being given in this order, a criteria is minimized to find the set of parameters $\mathcal{A}^k$ to define the best fit. This criteria can be either an evaluation of an implicit equation at each of the M_i^k or a distance between each point M_i^k and the model. In this last case, it can be written

$$E_1^k(\mathcal{A}) = \sum_i \|v_i(\mathcal{A}) - M_i^k\|^2 \tag{77}$$

The minimizing vector of parameters defines $\mathcal{A}^k$.

Step 2 : **Deformation of the set of points.** The vector $\mathcal{A}^k$ being given, a new set $\mathcal{M}^{k+1} = (M_i^{k+1})_{1\leq i\leq n}$ is defined by moving each point $v_i(\mathcal{A}^k)$ in the direction of descent of P to minimize

$$E_2(\mathcal{M}) = \sum_i P(M_i) \tag{78}$$

To avoid instability of the global shape (see [13] about time step), this displacement cannot be too large. The set of nodes $\mathcal{M}^k$ is then replaced for the next step by the new set $\mathcal{M}^{k+1}$ which is usually defined by:

$$M_i^{k+1} = v_i(\mathcal{A}^k) - \tau_i^k \nabla P(v_i(\mathcal{A}^k)) \tag{79}$$

The separation in two steps makes possible the minimization of the energy and avoids calculation of complex derivatives of $E(\mathcal{A})$. This was not possible directly since we do not always have a simple description of the shape when it is implicit.

6.3 Auxiliary variable formulation

If $(\frac{1}{2}\|N\|^2 - \tau P(N))$ is convex, or $(\frac{1}{2}x^2 - \tau f(x))$ is convex in the case $P = f(d_S)$, these two steps can be interpreted as the two variables minimization of an auxiliary energy:

$$E_{aux}(\mathcal{A}, \mathcal{M}) = \frac{1}{2\tau}\sum_i \|v_i(\mathcal{A}) - M_i\|^2 + \sum_i P_1(M_i) \tag{80}$$

where (τP_1) is associated to (τP) by equality (35) or (62) of theorems 2 and 4. Successive minimization with respect to the two variables $\mathcal{A}$ and $\mathcal{M}$ is as follows:

- $\mathcal{A}$ minimization corresponds to our first step of model fitting minimizing Eq. (77).
- $\mathcal{M}$ minimization corresponds to our second step of deformation. The minimizing $\mathcal{M}$ is a trade-off between the minimization of $P_1(M_i)$ and the closeness to $v_i(\mathcal{A})$. This gives the good direction of displacement and avoids a too large deformation of the set of nodes.

This has the advantage that the interaction between the shape $\mathcal{S}_\mathcal{A}$ and the data P is indirect through the M_i's. Also the two partial problems are easier to solve since we have a direct formulation for its resolution.

Using the results of Sections 3.4 to 5, we have

$$\begin{aligned} \inf_{\mathcal{M}} E_{aux}(\mathcal{A}, \mathcal{M}) &= \tfrac{1}{\tau}\sum_i \inf_{M_i}\{\frac{1}{2}\|v_i(\mathcal{A}) - M_i\|^2 + \tau P_1(M_i)\} \\ &= \sum_i P(v_i(\mathcal{A})) \end{aligned} \tag{81}$$

$$\inf_{\mathcal{A}}\{\inf_{\mathcal{M}} E_{aux}(\mathcal{A}, \mathcal{M})\} = \inf_{\mathcal{A}}\sum_i P(v_i(\mathcal{A})) = \inf_{\mathcal{A}} E(\mathcal{A}) \tag{82}$$

So the minimization of the two variables problem leads to the solution of the energy minimization of the initial problem (76). This shows that the usual second step (79) where potential P is minimized is the same as the minimization of (81) when $\tau_i^k = \tau$. This follows from theorem 3. However, the hypothesis on P may be difficult to check. As shown in section 5, if $P = f(d)$, the hypothesis on P depends only on f and not on the data defined by d.

Remark that the auxiliary variable can be seen as the local deformation of the global model best fit. This is very close to the idea used in [30] to locally deform superquadrics. So at convergence, $\mathcal{A}$ represents the best fit of the parametric model and $\mathcal{M}$ the local displacement between the model and the data. The main difference between the usual two-step approach and the two-variable minimization is that in the latter, we always have a descent in energy at each step. In the former case, the energy descent at one step can be balanced by an increase in the second step and generate oscillations. This remark is also valid for the other examples.

7 Pattern Matching. Registration.

The problem of pattern matching can be seen as a special case of deformable templates, where the parameters of the model correspond to the nature of the deformation.

7.1 Rigid or Affine Transform

Let us see first the case when the match between the model and data is done through an affine transform. We have a curve or surface model defined by a set of points $\mathcal{S}^1 = (X_i^1)_i$ and try to match this model to data. The data is a set of points $\mathcal{S}^2 = (X_j^2)_j$ and we try to minimize the average distance to data after a rotation or an affine transform R followed by a translation t. The energy we are trying to minimize has the form:

$$E(R,t) = \sum_i d(RX_i^1 + t, \mathcal{S}^2)^2 \tag{83}$$

where $d(RX_i^1 + t, \mathcal{S}^2) = \inf_j d(RX_i^1 + t, X_j^2)$. This problem would be very simple if to each point of $\mathcal{S}^1$, we knew in advance the point in $\mathcal{S}^2$ which corresponds, but this is one of the unknowns of the problem. Therefore, an iterative algorithm composed of two steps is often used to find the minimizing transform (see ICP, Iterative Closest Point algorithm in [31, 32, 33]).

Step 1 : **Matching**. For a previous estimate of the transform (R_n, t_n), find the matching between points of the transformed shape $(R_n\mathcal{S}^1 + t_n)$ and $\mathcal{S}^2$. This is done finding for each X_i^1 in $\mathcal{S}^1$ the closest point $X_{jn(i)}^2$ in $\mathcal{S}^2$ to $(R_nX_i^1 + t_n)$. The output is a list of couples $(X_i^1, X_{jn(i)}^2)_i$.

Step 2 : **Transform**. For a given set of matches, $(X_i^1, X_{jn(i)}^2)_i$, find the best transform (R_{n+1}, t_{n+1}) minimizing $\sum_i \|R_{n+1}X_i^1 + t_{n+1} - X_{jn(i)}^2\|^2$.

The necessity of these two steps comes from the fact that when matching the two sets of points, we do not know in advance which point of the first set corresponds to which point of the second set. The first step makes a tentative match and the second step solve the problem now made easier.

Defining $P(X) = \frac{\alpha}{2} d(X, \mathcal{S}^2)^2 = \frac{\alpha}{2} d_{\mathcal{S}^2}(X)^2$, we can identify this model to a deformable template since energy of (83) can be written:

$$E(R,t) = \sum_i P(RX_i^1 + t) \tag{84}$$

This is the same as Equation (76) with $\mathcal{A} = (R,t)$ and $v_i(\mathcal{A}) = RX_i^1 + t$. However the two steps are in reverse order, the matching corresponds to the deformation of the set of points in the previous section, while the minimizing transform corresponds to the model fitting in the deformable model.

7.2 Auxiliary variable formulation

In the case the distance d in Eqn. (83) is the same as the one induced by the norm in Step 2, this algorithm ensures convergence since at each step it is the same term that is minimized with respect to each of the variables (R,t) or j_n as shown in [31]. However, in the case d is a distance in a higher dimension space, including other features like in [33], or in the case the potential is a function of the distance taking into account only the best matches like in robust estimation, there is no reason for convergence and our formulation gives a way to ensure decreasing in the energy at each step.

As for deformable templates, we can understand these two steps as the minimization of an auxiliary energy with respect to two variables.

$$E_{aux}((R,t),\mathcal{M}) = \frac{1}{2}\sum_i \|RX_i^1 + t - M_i\|^2 + \sum_i P_1(M_i) \tag{85}$$

where P_1 is associated to P by equality (62) of theorem 4. In our case since $P(X) = \frac{\alpha}{2}d_{\mathcal{S}^2}(X)^2$, we have $P_1(M) = \frac{\alpha}{1-\alpha}\frac{1}{2}d_{\mathcal{S}^2}(X)^2$ which has the same expression.

Exactly as in the deformable template case, the (R,t) minimization corresponds to finding the best transform while the $\mathcal{M}$ minimization defines an auxiliary set of matching couples (X_i, M_i). The difference is that since the auxiliary points M_i are minimizing the energy (85), they are not necessarily points of $\mathcal{S}^2$. From equations (61) and section 5.3.2, we see that

$$M_i^n = (1-\alpha)(R_n X_i^1 + t_n) + \alpha X_{jn(i)}^2 \tag{86}$$

So instead of taking the match with $X_{jn(i)}^2$ directly, an intermediary point is chosen to satisfy better the convexity hypotheses and ensure descent in energy after the two steps.

7.3 General Non-Rigid Transform

If the deformation is nonrigid, it can be defined by a function v minimizing:

$$E_{nr}(v) = \sum_i d(v(X_i^1), S^2)^2 = \sum_i P(v(X_i^1)) \tag{87}$$

with the previous potential. Since this is not sufficient to define a unique deformation, we have either to constrain the kind of possible deformation and we come back to the problem of deformable templates of the previous section or regularize the problem and have a similar formulation as in the case of snakes dealt in next section.

8 Snakes

8.1 Active Contour Models

The snake model introduced in [3] is the minimization of the following energy:

$$E_{sn}(v) = \int_\Omega \{w_1\|v'(s)\|^2 + w_2\|v''(s)\|^2 + P(v(s))\}ds \tag{88}$$

All we say here is also true in the general case of regularization $R(v)$ like in energy of (7) and also in the case of 3D deformable surfaces ([34, 4]).
We show in [13] how the solution of the associated evolution equation

$$\begin{cases} \frac{\partial v}{\partial t} - (w_1 v')' + (w_2 v'')'' = F(v) = -\nabla P(v), \\ v(0,s) = v^0(s), \\ + \text{ boundary conditions} \end{cases} \tag{89}$$

is discretized into a two-step iteration algorithm. The system to solve is

$$(\mathcal{I} + \tau A)v^{t+1} = (v^t + \tau F(v^t)). \tag{90}$$

where A is the stiffness matrix obtained after discretization of the derivative terms either by finite differences or finite elements. It is decomposed in two steps, first a deformation along attraction force, then regularization (see figure 5):

$$\begin{cases} \textbf{Step 1:} & \text{Local deformation.} \quad v^{t+\frac{1}{2}} = (v^t + \tau F(v^t)) \\ \textbf{Step 2:} & \text{Regularization.} \quad v^{t+1} = (\mathcal{I} + \tau A)^{-1} v^{t+\frac{1}{2}} \end{cases} \tag{91}$$

Equation (89) is called a reaction diffusion equation and our decomposition in two steps could be also seen as a separation between these two aspects: the first step represents only reaction while the second represents only diffusion. Remark that in [35, Sec. 17.6], such a separation of iterations in substeps is also used to solve a linear equation by splitting an operator as a sum of operators. Each operator is then inversed in a separate step. It is applied there in a sum of differential operators while in the snake algorithm it applies to the differential and data operators. We now interpret each step of (91) as the minimization of an energy. In the first step, $v^{t+\frac{1}{2}}$ is the minimum with respect to the second variable v of

$$E_{sn2}(v^t, v) = \int_\Omega \{\frac{1}{\tau}\|v - v^t\|^2 + (P(v^t(s)) + \nabla P(v^t).(v - v^t))\} ds \tag{92}$$

where the potential term is in fact a first order linear approximation of $P(v)$. In the second step, v^{t+1} is the discrete minimization with respect to the first variable v of

$$E_{sn1}(v, v^{t+\frac{1}{2}}) = \int_\Omega \{w_1\|v'(s)\|^2 + w_2\|v''(s)\|^2 + \frac{1}{\tau}\|v - v^{t+\frac{1}{2}}\|^2\} ds \tag{93}$$

Remark that in fact the term $\int_\Omega \frac{1}{\tau}\|v - v^t\|^2$ is an approximation of $\int_\Omega \|\frac{\partial v}{\partial t}\|^2$ in (89).

8.2 Auxiliary variable formulation

Assuming $(\frac{1}{2}\|N\|^2 - \tau P(N))$ is convex, or $(\frac{1}{2}d^2 - \tau f(d))$ is convex in the case $P = f(d_S)$, these two steps can be interpreted as the alternate minimization of an auxiliary energy with respect to its two variables :

$$E_{sn}^{aux}(v_1, v_{aux}) = \int_\Omega \{w_1\|v_1'(s)\|^2 + w_2\|v_1''(s)\|^2 + \frac{1}{2\tau}\|v_1 - v_{aux}\|^2 + P_1(v_{aux}(s))\} ds \tag{94}$$

where v_1 is the feature curve, v_{aux} is an auxiliary curve and (τP_1) is associated to (τP) by equality (35) or (62) of theorems 2 or 4. The successive minimization of this energy with respect to the two variables v_1 and v_{aux} is as follows:

- The v_1 minimization corresponds to our second step of regularization, minimizing E_{sn1} in Eq. (93) above with respect to its first variable (2τ instead of τ).
- The v_{aux} minimization corresponds to our deformation step minimizing the partial energy:

$$E_{sn2}^{aux}(v_1, v_{aux}) = \int_\Omega \{\frac{1}{2\tau}\|v_1 - v_{aux}\|^2 + P_1(v_{aux}(s))\} ds \tag{95}$$

  This second energy is similar to E_{sn2} of step 1 in Eq. (92). The minimizing v_{aux} is a trade-off between the minimization of $P_1(v_{aux})$ and the closeness to v_1. This gives the good direction of displacement and avoids a too large deformation of the auxiliary curve. It follows from theorem 3 that the first step of Eqn. (91) is the same as the v_{aux} minimization.

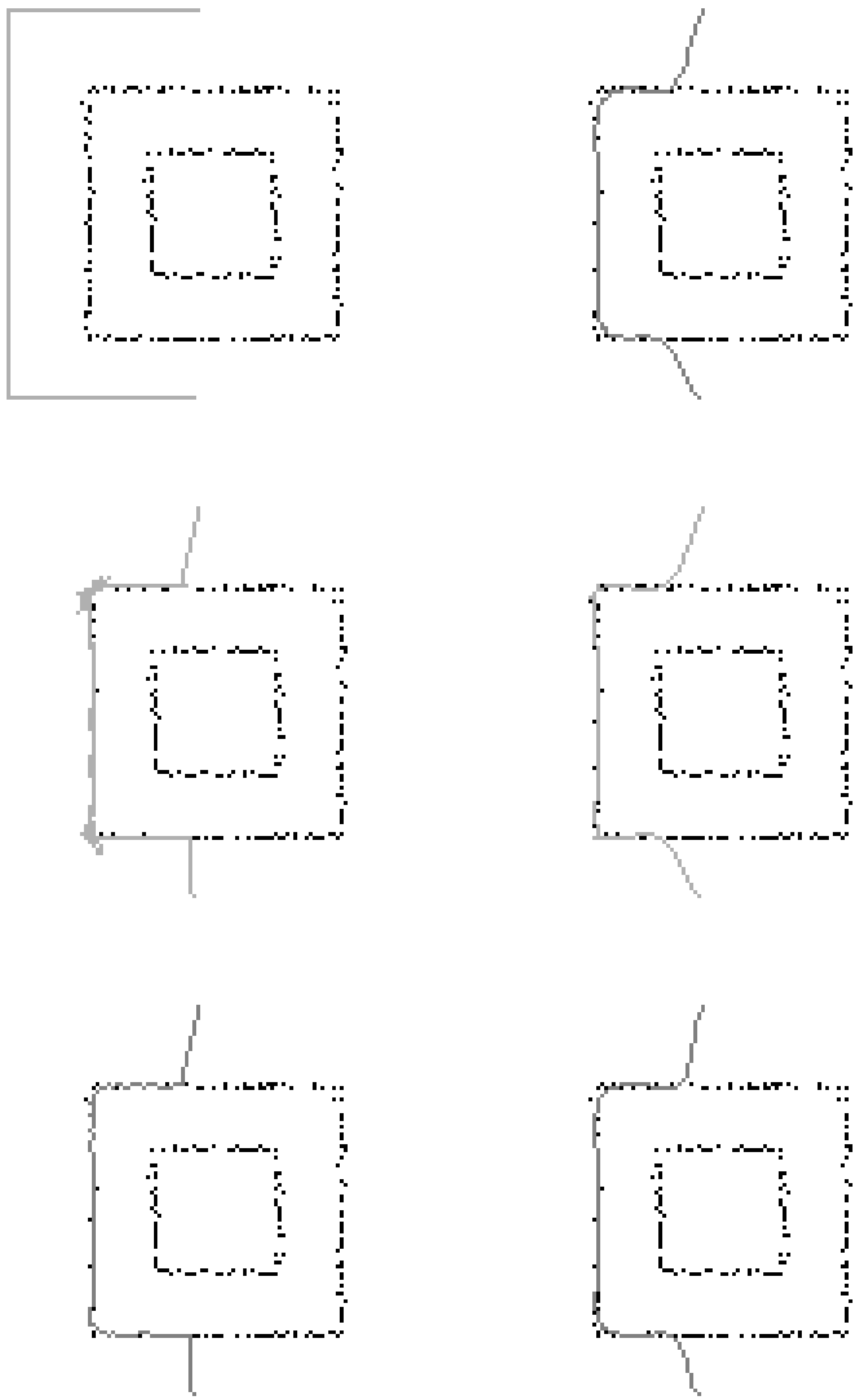

Figure 5: On the upper left the initial snake followed by one two-step iteration with $P = f(d)$ where f is in the limit case of the convexity in Theorem 4. The curve stabilizes very quickly to the solution. In the middle and bottom, the two half iterations are shown, local deformation (left) followed by regularization (right). The middle pair is the very first iteration and the bottom one is at convergence.

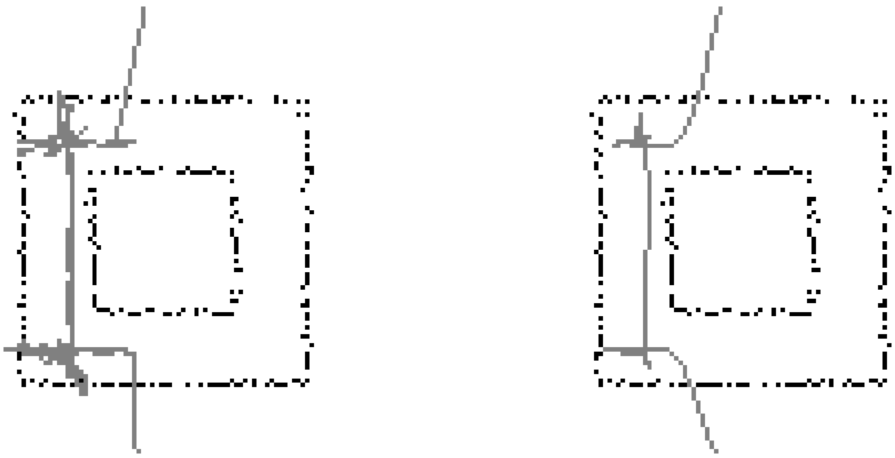

Figure 6: One two-step iteration (deformation on the left, smoothing on the right) with f being 1.5 times the limit case in Theorem 4. This is unstable and oscillates around edges.

This has the advantage that the interaction between the curve v_1 and the data P is indirect through v_{aux}. The two partial problems are easier to solve since we have a direct formulation for its resolution from sections 3 to 5 which give

$$\begin{aligned} \inf_{v_{aux}} E_{sn2}^{aux}(v_1, v_{aux}) &= \frac{1}{\tau}\int_\Omega \inf_{v_{aux}(s)} \{\frac{1}{2}\|v_1(s) - v_{aux}(s)\|^2 + \tau P_1(v_{aux}(s))\} ds \\ &= \int_\Omega P(v_1(s)) ds \end{aligned} \tag{96}$$

$$\inf_{v_1}\{\inf_{v_{aux}} E_{sn}^{aux}(v_1, v_{aux})\} = \inf_{v_1} E_{sn}(v_1) \tag{97}$$

So the minimization of the two variables problem leads to the solution of the energy minimization of the initial problem (88) and ensures convergence and avoid oscillations that can occur if the convexity hypothesis is not satisfied (see figure 6). This last result is also true for a local minima in v_1 of E_{sn}. Remark also that the auxiliary curve appears to be a more precise localization of features but less smooth than the actual curve. It maybe more interesting than the curve itself in case the features are not too much noisy and regularization is needed only for a global behavior.

Notice that in our snake algorithm [13], there is a resampling of the curve from time to time to avoid concentration of nodes in some areas and account for length variation. This may be interpreted as the use of a geometric model of the curve or close to a two step formulation of the shape reconstruction on a varying mesh of [9].

8.3 Other related problems with two step Iterative algorithms

We saw in section 8.1 that the two steps came from the semi-explicit scheme of equation (90). Other algorithms have this same property of being explicit in the non linear part and implicit in the linear part.

In some restoration problems where there is in the energy $E(u)$ a non linear function of the gradient of the image u, each iteration is replaced by two steps. The first minimizes the energy with the previous value of the gradient in this non linear term. The second is a new evaluation of the gradient from the new value of u. This may be seen as using an auxiliary linear problem at each time step. We give here two examples:

Anisotropic Diffusion. To solve the evolution equation of anisotropic diffusion ([36]) a kind of two-step algorithm was used in [37]. The equation is

$$\frac{\partial u}{\partial t} - \nabla(g(\|\nabla u\|)\nabla u) = 0 \tag{98}$$

with usually $g(d) = \frac{1}{(1+d^2)^m}$. This is discretized using finite differences:

$$\begin{cases} \text{Step 1.} & G^n = g(\|\nabla u^n\|) \\ \text{Step 2.} & u^{n+1} - u^n - \tau\nabla(G^n \nabla u^{n+1}) = 0 \end{cases} \tag{99}$$

Non linear Total Variation. The same kind of equation with $g(\nabla u) = \frac{1}{\|\nabla u\|}$ also appears in a restoration algorithm based on non linear total variation [38] minimizing $\int \sqrt{u_x^2 + u_y^2}$. This gives the evolution equation similar to (98):

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}}\right) + \frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}}\right) \tag{100}$$

This is solved using a totally explicit scheme. A similar approach is used for optical flow computation in [39], but the scheme is semi-explicit. This means that the non linear part $\frac{1}{\sqrt{u_x^2+u_y^2}}$ is explicit while the partial differential equation solved is similar to (99). This could be interpreted as solving an auxiliary problem where this nonlinear term is not time dependent.

Note that in these examples, the original problem is modified by freezing a nonlinear part at each step. This can be also interpreted as smoothing algorithms with variable weights. A first step estimates the weights and a second solves the minimization with these constant weights. Such an approach is also used for deformable templates in [40]. Those variable weights algorithms have been transformed using a different duality relation in two variables problems in [18, 20, 19]. This is also related to the EM algorithm. This makes use of a function f_1 such that

$$\inf_v v\|\nabla u\|^2 + f_1(v) = f(\|\nabla u\|) \tag{101}$$

and defines the auxiliary energy such as:

$$E_{aux}(u, v) = \int v\|\nabla u\|^2 + f_1(v) + \int \|u - d\|^2 \tag{102}$$

This duality permits to give a closer relation between the anisotropic diffusion and the Blake and Zisserman algorithm. Indeed, the non linear term in the latter energy is:

$$\int f(\|\nabla u\|) = \int \frac{f(\|\nabla u\|)}{\|\nabla u\|^2}\|\nabla u\|^2 = \int g(\|\nabla u\|)\|\nabla u\|^2 \tag{103}$$

and for $f(d) = Inf(\beta, \alpha\frac{d^2}{2})$, $g(d) = Inf(\frac{2\beta}{d^2}, \alpha)$. This function is very close to the one used in anisotropic diffusion.

There is also successive minimization over two variables in the more general Uzawa algorithm [22, Chapter VII]

EM algorithm. Adding auxiliary variables for alternate minimization was also used in a different way in the EM algorithm ([41]). Examples of application of this algorithm for image identification and restoration are presented in [42, Chapter 7]. It is also close to a minimization in a Mumford & Shah [6] like image segmentation presented briefly in [26], where the energy depends on the reconstruction of the image and of the line process field l. This last paper also quotes the link to the problem of smoothing algorithms with variable weights. These weights can be considered also as unknown extra variables. In these examples as well as in our approach, an energy is minimized alternatively with respect to each variable. A further similarity is that for each iteration there is one quadratic step and one analytic step.

The general idea of the EM algorithm deals with the problem of finding the parameters of a model with some incomplete data (observed variables), and there are two other multi-dimensional variables or arrays - the model parameters and the unobserved variables. The missing information does have a physical meaning but the goal is to estimate the model parameters, which would be easy or easier if one could observe the unobserved variables. To solve the problem, extra (unobserved) variables are then estimated to restore the missing data. Alternate minimization of a probability distribution (which may be obtained by an energy using Gibbs distribution) gives an estimation of the unobserved variables for a given set of parameters and then the new complete data leads to a new set of parameters. In the image restoration example in [42], the data is the noisy blurred image, the unknown is the pair of arrays given by the true (restored) image (unobserved variable) and the point spread function (unknown parameters). In [43], the EM algorithm is used for 3D curve reconstruction from projections using an approach close to B-snakes described in section 9.1.

An important difference with our work is that in the EM approach, extra variables have a real meaning but are not available as data. In our approach, we introduce new artificial variables, that have no physical meaning, to make the minimization work better by adding convexity to the energy. This is related in spirit but different in details upon closer examination for most of our examples. However in the Pattern Matching example (see section 7), we may identify the observed data as the deformed pattern $\mathcal{S}^2$, the parameters as the transformation (R, t) and the unobserved variables as the correspondence j between a data point and the points of the transformed original model $\mathcal{S}^1$.

9 B-Snakes

We present the B-snake model and show how it can be interpreted as a two-step minimization of a two variables energy.

Our purpose in this section is not only to interpret an algorithm as an auxiliary energy minimization as in the other examples. It is mainly to set a more precise link between classic snakes, as presented in the previous section, and the B-snake model. We show that B-snakes or spline-snakes may be interpreted as the solution of a discretization of a snake energy like (112). This means that the solution may be seen either as the minimum of a data energy among **all cubic spline functions** or a minimum of a classic snake energy among **all functions**.

Leitner *etal* [5] introduced a simplified active contour model they called "Spline-Snakes." This model was also used by many authors with the name of B-Snakes. The solution of their snake model is found by deformation of a set of node points submitted to an attraction force and then by curve reconstruction using B-Splines.

Note that in [44], the snake problem is solved using a finite element method. B-Splines can be seen as a special case of finite element, but the main difference is that in the snake-spline model, the energy is only external. Formulations equivalent to the finite element one presented in [44] have also been used in [45, 46], keeping the internal energy term. In [47] some intermediary model was introduced.

9.1 Description of the method

The main idea is that since the classical snake energy contains a regularization part and cubic splines already minimize a regularization energy, it may be easier to limit the curve set to cubic spline curves and minimize only the potential energy (see [48]).
The active contour is now parametrically represented by

$$\forall s \in [0,1], v_{\mathcal{A}}(s) = \sum_{i=1}^{n} B_i(s)\alpha_i, \tag{104}$$

where $\mathcal{A} = (\alpha_i)_{1 \le i \le n}$, $\alpha_i \in I\!R^2$ are the control points, B_i are the basis cubic B-spline functions.

These basis B-Spline functions $B_i(s), 1 \le i \le n$ already minimize the regularizing energy:

$$\int_0^1 b''(s)^2 ds, \tag{105}$$

for all C^2 real valued function $b(s)$ satisfying $b(s_i) = 1,\ b(s_j) = 0,\ 1 \le j \le n,\ j \ne i,\ \ s_j = \frac{(j-1)}{(n-1)}$.

It follows that $v_{\mathcal{A}}$ minimizes the following energy

$$\int_0^1 [(x''(s))^2 + (y''(s))^2] ds, \tag{106}$$

for all $v(s) = (x(s), y(s))$ satisfying $v(s_j) = \alpha_j, 1 \le j \le n, s_j = \frac{(j-1)}{(n-1)}$.

A cubic B-Spline function is defined by its control points, and it represents among all functions passing by these control points, the one minimizing the above regularization energy.

A snake is minimizing an energy making a trade-off between data fitting and regularization. Accordingly, the snake-spline is the B-spline curve that minimizes only the second part of the snake energy, the potential:

$$\text{Minimize} \int_0^1 P(x(s), y(s)) ds \tag{107}$$

This external energy can take the discrete form:

$$\text{Minimize} \sum_{i=1}^{m} P(M_i) \tag{108}$$

where the M_i form a set of nodes representing the curve. We formulate the method of resolution proposed in [5] as a two-step iterative algorithm beginning with an initial set of data points $\mathcal{M}^0$.

Step 1 : **Spline Fitting**. A set of points $\mathcal{M}^k = (M_j^k)_{1\leq j\leq m}$ being given, the set of control points $\mathcal{A}^k = (\alpha_i^k)_{1\leq i\leq n}$ is determined to minimize the least square error $||\mathcal{M}^k - \mathcal{B}\mathcal{A}||^2$:

$$\sum_{j=1}^{m} \|M_j^k - \sum_{i=1}^{n} B_{ij}\alpha_i\|^2, \tag{109}$$

where $B_{ij} = (B_i(t_j))_{1\leq i\leq n, 1\leq j\leq m}, t_j = \frac{(j-1)}{(m-1)}$, the solution is $\mathcal{A}^k = (\mathcal{B}^t\mathcal{B})^{-1}\mathcal{B}^t\mathcal{M}^k = \mathcal{C}\mathcal{M}^k$.
If $m = n$, $\mathcal{B}$ is the identity matrix, and the least square corresponds to the B-spline interpolation of the $M_j^k = \alpha_j^k$.
If $m > n$, the B-spline is over-determined and some smoothing of the nodes M_j^k takes place to determine the α_i^k. This smoothing of nodes M_j^k is a further justification for omitting the regularization term in the snake energy.

Step 2 : **External Force**. A set of control points $\mathcal{A}^k$ being given, the external force is applied separately to each point of $\mathcal{B}\mathcal{A} = (v_{\mathcal{A}^k}(t_j))_{1\leq j\leq m}$ to define a new set of data points.

$$\forall j \in \{1, m\}, M_j^{k+1} = v_{\mathcal{A}^k}(t_j) + \tau F_j^k, \tag{110}$$

where the force $F_j^k = -\nabla P(v_{\mathcal{A}^k}(t_j))$. After applying a local deformation to the set of nodes, we go back to the previous step with this $\mathcal{M}^{k+1}$.

Remark that an analogous algorithm was also used for 3-D splines solving the inverse problem of Free-Form Deformations in [15]. It alternates determination of control points and displacement field computation. In fact, in [5], the first of the two steps is included in the second. This gives an evolution equation for the set $\mathcal{A}$:

$$\frac{d\mathcal{A}}{dt} = \mathcal{C}\frac{d\mathcal{M}}{dt} = \mathcal{C}F(\mathcal{M}) = \mathcal{C}F(\mathcal{B}\mathcal{A}). \tag{111}$$

However, this does not matter for our purpose since this evolution equation is in fact a two step algorithm and our purpose in this example is to show the link between this algorithm and the minimization of the original snake energy.

9.2 Interpretation as an Auxiliary Variables Problem

In the context of the previous section, the problem we want to solve is defined through the minimization of the following energy similar to a half discrete version of the snake model (88):

$$E_{bs}(v) = \int_0^1 \|v''(s)\|^2 + \sum_{i=1}^{m} P(v(s_i)) \tag{112}$$

This is an implicit version of the problem presented in [12]:

$$E_{u_i}(v) = \lambda^2 \int_0^1 \|v''(s)\|^2 + \sum_{i=1}^{N} \|u_i - v(s_i)\|^2 \tag{113}$$

If a function v is a minimizer of one of these last two problems, we can show that its fourth derivative vanishes everywhere except at the nodes s_i where v is singular. It follows that the solution is a cubic spline and for problem (113), the coefficients can be defined from the values u_i as a convolution kernel if the s_i are regularly spaced. The basis of solutions is sometimes called interpolation splines.

For problem (112), the equation found at the junctions s_i involves P and the resolution is not straightforward like for energy (113). Therefore we have to use a minimization algorithm and the one described in the previous section 9.1 corresponds to the successive minimization of each term of the energy. We now give a better understanding of how this minimization is done in that algorithm using the definition of auxiliary variables in the energy:

$$E_{bs}^{aux}(v,\mathcal{M}) = \int_0^1 \|v''(s)\|^2 + \sum_{i=1}^{n} \frac{1}{2\tau}\|v(s_i) - M_i\|^2 + \sum_{i=1}^{n} P_1(M_i) \tag{114}$$

where τP_1 is the potential associated to τP by equality (35) or (62) of theorems 2 and 4 assuming P satisfies the demanded hypotheses.

- When $\mathcal{M}$ is given, solving the minimization with respect to v is the same as solving the problem (113). So, the solution in v is a cubic spline with nodes at s_i that approximates the M_i's. It can be defined by a set of coordinates $\mathcal{A}$ in the spline basis. This corresponds to the first step of "spline fitting".

- When v (or $\mathcal{A}$) is given, minimizing the energy with respect to $\mathcal{M}$ is the determination of the M_i's that are at the same time close to the $v(s_i)$ and have low value of P_1. So v being given the M_i's solve a trade off between localization of features and small deformation from v by minimizing:

  $$E_{bs2}^{aux}(\mathcal{M}) = \sum_{i=1}^{n} \frac{1}{2\tau}\|v(s_i) - M_i\|^2 + \sum_{i=1}^{n} P_1(M_i) \tag{115}$$

  This corresponds to the second step in the algorithm of the previous section ("external force").

It turns out that the two steps of the Spline-snakes method correspond to minimizing the energy of (114) alternatively with respect to v or $\mathcal{M}$. However, the basis of spline functions is different and corresponds to the interpolation splines. This makes sense since in the snake model a deformation at one point of the curve has influence on the whole curve. This is the case in the interpolation splines minimizing (113) while a deformation in the case of B-splines has only local consequences. Although this last property may be desired sometimes, this makes a difference with the snake model.

In a way similar to Eqns. (81) and (82), the minimum of E_{bs}^{aux} in (114) is the minimum of (112):

$$\begin{aligned} \inf_{\mathcal{M}} E_{bs2}^{aux}(\mathcal{M}) &= \sum_i \inf_{M_i}\{\frac{1}{2\tau}\|v(s_i) - M_i\|^2 + P_1(M_i)\} \\ &= \sum_i P(v(s_i)) \end{aligned} \tag{116}$$

$$\inf_v\{\inf_{\mathcal{M}} E_{bs}^{aux}(v,\mathcal{M})\} = \inf_v E_{bs}(v) \tag{117}$$

If the potential used is of the form $P = \alpha\frac{d^2}{2}$, (as in [5]) as seen in section 5.3.2, the potential P_1 that has to be used for intermediate steps is of the same kind $P_1 = \frac{\alpha}{1-\tau\alpha}\frac{d^2}{2}$. So in this case, equation (68) becomes:

$$\forall i \in \{1, m\}, M_i^{k+1} = (1-\tau\alpha)v_{\mathcal{A}^k}(s_i) + \tau\alpha v_S^{k,i}, \tag{118}$$

where $v_S^{k,i}$ is the projection on the data S of $v_{\mathcal{A}^k}(s_i)$. Remarks of the previous sections still apply.

Notice that in our formulation, we deal with the case $n = m$ in (109), but in the general case, to have m auxiliary points and keep $n < m$ basis functions, we could consider the auxiliary term $\sum_{i=1}^{m} \frac{1}{2\tau} \| \sum_j B_{ji} v(s_j) - M_i \|^2$, with the same B_{ji} as before. Our auxiliary variable interpretation is still valid and the minimized energy becomes

$$E_{bs}(v) = \int_0^1 \|v''(s)\|^2 + \sum_{i=1}^{m} P(\sum_j B_{ji} v(s_j)) \tag{119}$$

which integrates in the discretization of the potential term the spline weights between chosen values $v(s_j)$.

In the energy of (112), if the regularization term contains a first derivative or a combination of the two first derivatives (like in snakes), the result of the v minimization gives a different kind of splines presented in [49, 50]. In case the snake internal term is half discretized $a \int v''(s) + b \sum v'(s_i)$, the minimizing curves are called $\nu-$Splines [50]. In all these cases, since the solution of the semi-discretized problem, similar to (112) with various regularization terms, can be described in a space of spline functions, the solution can be formulated by its decomposition $\mathcal{A}$ in this space. It is then natural to look for minimization of the potential term in this space of spline functions. Our formulation gives the link between the original snake like problems and, through an auxiliary variable energy, the two-step algorithm minimizing P along all curves in the spline space. This is a further justification for finite element methods[4] or spline-snakes.

10 Conclusion

We have presented a new mathematical formulation to shape extraction and reconstruction problems. By introducing auxiliary variables we defined a two-variables energy and showed some mathematical results on its successive minimization with respect to each variable. These variables represent an intermediary reconstructed shape. This permits to transform a problem with implicit data constraints defined through the minimization of a non convex potential into an explicit reconstruction convex problem defined by regularization of known data. We showed a way to make this transformation easily in the case where the potential is a function of the distance to the closest data. This kind of potential is more and more used in shape extraction work.

This permits to give a better understanding of many already existing two-step algorithms used for deformable templates and deformable models. This permits also to give a more precise link between snakes and B-snakes. They can be interpreted as a minimization of a two-variables energy. Our work can give a good mathematical formulation to many of these data extraction and reconstruction algorithms to modify current algorithms and ensure that the initial energy is minimized and that the algorithm converges.

Acknowledgments

I would like to thank **Donald Geman** and **Demetri Terzopoulos** for stimulating discussions and comments, and also **Eric Séré, Robert Azencott** and the reviewers for their useful suggestions to improve this paper.

References

[1] W.E.L. Grimson. *From Images to Surfaces: A computational study of the Human Early vision system.* The MIT Press, 1981.

[2] Demetri Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438, July 1988.

[3] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[4] Laurent D. Cohen and Isaac Cohen. Finite element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-15(11), November 1993.

[5] F. Leitner, I. Marque, S. Lavallée, and P. Cinquin. Dynamic segmentation: finding the edge with snake-splines. In *Proceedings of International Conference on Curves and Surfaces*, pages 1–4, Chamonix, France, June 1990. Academic Press.

[6] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proceedings of Computer Vision and Pattern Recognition*, San Francisco, June 1985.

[7] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. Technical report, Umass, July 1993. to appear in *IEEE Transactions on Image Processing.*

[8] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(3), September 1993.

[9] Isaac Weiss. Shape reconstruction on a varying mesh. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(4), April 1990.

[10] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems.* Winston and sons, 1977.

[11] H. Brezis. *Analyse Fonctionnelle, Théorie et applications.* Masson, Paris, 1983.

[12] T. Poggio, H. Voohrees, and A. Yuille. A regularized solution to edge detection. *Journal of Complexity*, 4:106–123, 1988.

[13] Laurent D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 53(2):211–218, March 1991.

[14] Isaac Cohen and Laurent D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. In *Proceedings of the 12th IEEE International Conference on Pattern Recognition (ICPR'94)*, pages B–403–405, Jerusalem, 1994. Part of Inria TR 2188, to appear in *Computer Vision, Graphics, and Image Processing : Image Understanding.*

[15] Eric Bardinet, Laurent Cohen, and Nicholas Ayache. Fitting 3D data using superquadrics and free-form deformations. In *Proceedings of the 12th IEEE International Conference on Pattern Recognition (ICPR'94)*, pages A–79–83, Jerusalem, October 1994.

[16] D. Suter. Constraint networks in vision. *IEEE Transactions on Computers*, 40(12):1359–1367, December 1991.

[17] Andrew Blake and Andrew Zisserman. *Visual Reconstruction.* The MIT Press, 1987.

[18] D. Geman and Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:367–383, March 1992.

[19] G. Aubert, M. Barlaud, L. Blanc-Féraud, and P. Charbonnier. A deteministic algorithm for edge-preserving computed imaging using Legendre transform. In *Proc. 12th International Conference on Pattern Recognition*, pages C–188–191, Jerusalem, Israel, October 1994.

[20] A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problems. Technical Report 9509, CEREMADE, Université Paris-Dauphine, Paris, France, February 1995.

[21] P. G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation.* Masson, Paris, 1985.

[22] I. Ekeland and R. Temam. *Convex analysis and variational problems.* NorthHolland, Amsterdam, 1976.

[23] Gunilla Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[24] P. E. Danielsson. Euclidean distance mapping. *Computer Vision, Graphics, and Image Processing*, 14:227–248, 1980.

[25] Huber P.J. *Robust Statistics.* Wiley, 1981.

[26] D. Geiger and A.L. Yuille. A common framework for image segmentation. *International Journal of Computer Vision*, 6(3):227–234, August 1991.

[27] N. Kiryati and A.M Bruckstein. What's in a set of points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14(4):496–500, April 1992.

[28] A.P. Pentland. Recognition by parts. In *Proc. First International Conference on Computer Vision*, pages 612–620, 1987.

[29] Song Han, Dmitry B. Goldgof, and Kevin W. Bowyer. Using hyperquadrics for shape recovery from range data. In *Proc. Fourth International Conference on Computer Vision*, pages 492–496, Berlin, June 1993. IEEE.

[30] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[31] Paul Besl and Neil McKay. A method for registration of 3−D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[32] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[33] J. Feldmar and N. Ayache. Locally affine registration of free-form surfaces. In *IEEE Proceedings of Computer Vision and Pattern Recognition 1994 (CVPR'94)*, Seattle, USA, June 1994. To appear in *International Journal of Computer Vision.*

[34] Isaac Cohen, Laurent D. Cohen, and Nicholas Ayache. Using deformable surfaces to segment 3-D images and infer differential structures. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 56(2):242–263, September 1992.

[35] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C. The art of Scientific Programming.* Cambridge University Press, 1990.

[36] Pietro Perona and Jitendra Malik. Scale space and edge detection using anisotropic diffusion. In *Proc. IEEE Workshop on Computer Vision*, pages 16–22, Miami, FL, 1987.

[37] Laurent Cohen. Filtres adaptatifs et contours. Technical report, Schlumberger Montrouge Recherche, Octobre 1989.

[38] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. In *Ecoles CEA - EDF - INRIA; Problèmes Non Linéaires Appliqués: Modélisations Mathématiques pour le traitement d'images*, pages 149–179, Rocquencourt, France, March 1992.

[39] Isaac Cohen. Nonlinear variational method for optical flow computation. In *Proceedings of the 8th Scandinavian Conference on Image Analysis*, pages 523–530, Tromso, Norway, June 1993. IAPR.

[40] S. Kumar and D. Goldgof. A robust technique for the estimation of the deformable hyperquadrics from images. In *Proceedings of the 12th IEEE International Conference on Pattern Recognition (ICPR'94)*, pages A–74–78, Jerusalem, October 1994.

[41] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

[42] Reginald L. Lagendijk and Jan Biemond. *Iterative Identification and Restoration of Images*. Kluwer Academic Press, 1991.

[43] B. Chalmond, F. Coldefy, and B. Lavayssiere. 3D curve reconstruction from degraded projections. In L. Schumaker P.J. Laurent, A. Le Mehauté, editor, *Wavelets, Images, and Surface Fitting. Proceedings of the conference on Curves and Surfaces*, pages 113–119, 1994.

[44] Laurent D. Cohen and Isaac Cohen. A finite element method applied to new active contour models and 3D reconstruction from cross sections. In *Proc. Third International Conference on Computer Vision*, pages 587–591, Osaka, Japan, December 1990.

[45] Sylvie Menet, Philippe Saint-Marc, and Gerard Medioni. B-snakes: Implementation and application to stereo. In *Proceedings of the Seventh Israeli Conference on Artificial Intelligence and Computer Vision*, pages 227–240, Tel aviv, Israel, December 1990.

[46] S. Sinha and B. Schunk. Surface approximation using weighted splines. In *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, Hawai, June 1991.

[47] André Guéziec. Large deformable splines, crest lines and matching. In *Proc. Fourth International Conference on Computer Vision*, Berlin, May 1993.

[48] N. Ayache, P. Cinquin, I. Cohen, L. Cohen, F. Leitner, and O. Monga. Segmentation of complex 3D medical objects : a challenge and a requirement for computer assisted surgery planning and performing. In R. Taylor and S. Lavallee, editors, *Computer Integrated Surgery*. MIT Press, 1994.

[49] T. A. Foley. Interpolation with interval and point tension controls using cubic weighted ν-Splines. *ACM Transactions on Mathematical Software*, 13(1):68–96, March 1987.

[50] B.A. Barsky. Exponential and polynomial methods for applying tension to an interpolating spline curve. *Computer Vision, Graphics, and Image Processing*, 27:1–18, 1984.