Subvoxel model-based 3D segmentation using implicit snakes

Eric Bainville

July 9th, 1997

CMU-RI-TR-97-26

The Robotics Institute
Carnegie Mellon University
5000 Forbes avenue
Pittsburgh Pennsylvania 15213-3890

Abstract

Building accurate representations of the surfaces of bones in 3D medical images is an important task in many computer-assisted surgery procedures. We first present a survey of the existing solutions to this problem. Then, we give the results of experiments on new 2D and 3D deformable models. We propose to maximize the correlation of raw image values along contours parallel to the surface of bones. We also propose new representations of deformable contours : an orientation-based representation for 2D contours, and an implicit representation using 4D splines for 3D contours. We finally present some results of subvoxel segmentation of bone surfaces.

This research was supported by a post-doctoral grant of the *Institut National de Recherche en Informatique et Automatique* (INRIA, France)

1 Introduction

The problem of accurately building a representation of the surfaces of bones from 3D medical images is an important step in many computer-assisted surgery procedures.

In this report, we first give a rapid survey of the existing segmentation techniques in section 2, with a special interest on model-based segmentation. We also give in section 3 some references of the works related to sub-pixel feature extraction in 2D images used in computer vision.

We present in section 4 some experiments using these two techniques in order to segment the bone surfaces in 3D images at a subvoxel precision.

2 Segmentation and model-based segmentation

Segmentation is the process taking in input an image (two- or tri-dimensional, binary, grayscale, color) of an object, and giving as output a representation of the object. This representation may be of different structuration levels.

An unstructured representation is a set of geometric primitives (points, segments, lines, surface patches,...) each representing locally the object.

A structured representation is a structure representing globally the object (set of internal voxels, triangulation, implicit surfaces, parametrized surfaces, splines, volumic mesh, skeleton).

The application determines the choice of a specific representation.

We describe here the different techniques used to perform segmentation. We make a distinction between low level algorithms, directly working on the image, and high level algorithms, working on the characteristics provided by the low level algorithms.

2.1 Low level algorithms

2.1.1 Manual segmentation

This tool is the most used. An operator with anatomical knowledge draws contours interactively on different slices of the dataset. The result is a set of 2D oriented contours. This method is slow, tedious and usually not reproducible. Its accuracy is determined by the operator's skill and the image resolution.

2.1.2 Mathematical morphology

Erosion, dilatation, and thresholding are simple tools that can be very efficient. The result is a set of voxels representing either the boundary of the interior of the object. The problem with these methods is to find the correct succession of

operators to apply and the correct set of parameters, which may vary among datasets.

2.1.3 Partial derivatives

Considering the image as a function $\mathbb{R}^3 \rightarrow \mathbb{R}$, we can compute the partial derivatives, and from these values, compute some characteristic values like the gradient, the laplacian.

From these values, one can construct sets of points that have a high probability to lie on the boundary of the objects.

Under the assumption that the boundary of the object corresponds to an iso-surface of the image function, one can also compute tangent planes, principal curvature directions and crest lines on the object surface.

The paper by Monga and Benayoun [28] gives references about partial derivatives and curvatures computation. The work of Thirion, Gourdon, Subsol and Ayache [31, 36, 35] show how to extract crest lines and use them for registration between different objects.

The gradient amplitude is a reliable and frequently used characteristic, crest lines remain stable under anatomical variations. The computation of crest lines in an image requires a lot of CPU time, and pre-filtering is necessary to attenuate the effects of the noise.

2.2 High level algorithms

2.2.1 Deformable surfaces

The idea is to start from an already structured representation, and to deform it to match the object surface. This is an optimization process, with a surface (i.e. a finite set of parameters associated to a given surface representation) as unknown.

The "energy" to optimize is the sum of an image term and a regularisation term. The image term is designed to be minimal when the surface matches the object boundary, and the regularisation term prevents the surface from becoming irregular.

Usually, the image term is the sum of the gradient magnitude in the image taken over the current surface. The regularisation term is the sum of first and second order derivatives on the surface itself.

Introduced by Kass, Witkin and Terzopoulos in 1988 [23], these models have in the last 5 years been ported to the 3D case (deformable models were first introduced by Terzopoulos et al. for computer graphics in 1987 [34]). We may cite the 3D models of Cohen, Cohen and Ayache [8], and Delingette, Hebert and Ikeuchi [10].

The result is frequently dependent on the initial contour. In 3D, some other problems are the parametrization of the surface, and the high number of degrees of freedom.

Dealing with the problem of parametrization, there are works on adaptive topology deformable surfaces, by Leitner and Cinquin [14], the author and Lachaud [25], McInerney and Terzopoulos [27].

Some authors propose to constrain the surface to belong to a restricted set of shapes, corresponding to the anatomical variations between different patients. Thus, the optimization space is far smaller. This approach requires the creation of a large enough set of training surfaces (segmented by hand), and a statistical study of these surfaces. See the work of Szekely et al. [32], and Hill, Cootes and Taylor [18]. This last work does not use the gradient, but identifies patterns in the gray levels in the direction normal to the contour.

2.2.2 Surface reconstruction from points or contours

The oldest algorithm is the *marching cubes* algorithm by Lorensen and Cline [26], which creates a triangulated surface from a set of interior voxels. Algorri and Schmitt [1] propose an two-step amelioration of this algorithm. See also the work of Bloomenthal [4] on polygonalization of implicit surfaces.

The number of created triangles is about twice the number of voxels of the surface, typically between 20,000 and 1,000,000 triangles. Such a number of faces may not be easily handled by the applications, and the triangulations may need to be resampled (see next paragraph).

An other class of algorithms reconstruct triangulated surfaces from a set of contours in parallel planes. The *NUAGES* package, provided by B. Geiger at INRIA, is an implementation of such algorithms. It is described in [15]. The number of triangles is also about twice the number of points.

Some algorithms take as input an arbitrary set of points to build a triangulation of the most probable surface passing through the points. Bittar et al. [3] uses implicit surfaces and medial axis transformation, and Attali [2] uses Delaunay simplicial decomposition.

Other algorithms choose to match either globally or locally surface patches with the data points, minimizing a distance criterion. Taubin proposes in [33] an algorithm to fit an algebraic surface and a set of points with an Euclidean distance criterion. Bricault and Monga [7] construct a set of quadrics approximating a set of points. The successive articles of Hoppe, DeRose, Duchamp, McDonald, Stuetzle, Halstead, Jin, Schweitzer, Eck and Lounsbery [20, 21, 19, 11, 12] present algorithms to build more and more complex and accurate structured representations (from triangulation to optimized B-spline mesh) from an unstructured set of 3D points.

2.2.3 Triangulation simplification

As noted above, some reconstruction algorithms produce a too large number of triangles. Solutions have been proposed to this problem by transforming a triangulation to another having far less triangles, but remaining under a given distance of the original surface.

Gueziec presents [16] an algorithm based on local vertex fusion. See also the references given in a recent paper of Delingette [9].

2.2.4 Meshing

Meshing the interior of a volume known through a triangulated surface is an important research problem. As a mesh is needed to solve number of engineering problems using the finite element methods, constructing meshes is an active research field. In 2D, the existing solutions can provide triangulations verifying all the desired constraints (no small angles, no thin triangles,...), for arbitrary shapes (with holes, for example). In 3D, there are only a few algorithms, and there is still work to do. See *Mesh Generation and Grid Generation on the Web* at the URL `http://www-users.informatik.rwth-aachen.de/ roberts/meshgeneration.html`, maintained by Robert Schneiders.

Building 3D regular meshes from the 3D skeleton is also an interesting area to explore.

3 Subpixel feature detection

The precise detection of the position of characteristic features (edges, discs, corners, crosses) in video images is used in computer vision, for example in vision system calibration.

Shortis, Clarke and Short [30], Tian and Huhns [37], Valkenburg, McIvor and Power [39] present studies of different subpixel detection methods.

Bose and Amir [5] compare different fiducials (characteristic marks drawn on objects) for an application in precise registration. Efrat and Gotsman [13], Havelock [17] study the projections of fiducials and the uncertainty of their detection.

Kammoun and Astruc [22], Kisworo, Venkatesh and West [24] deal with the problem of precise detection of edges.

Brand and Mohr [6] give an algorithm to detect corners at a sub-pixel precision, optimizing a model of corner locally describing the image. Peuchot [29] presents a similar algorithm working on crosses.

4 Experiments

The basic idea of these experiments is to use the local correlation between image values when moving in a direction tangent to the surface.

Starting from a first rough estimate of the surface position, we propose to optimize a scalar field defined locally such that at the end of the process, the field values represent the signed distance to the actual surface. The result is the iso-surface of the field corresponding to the value 0.

4.1 2D correlation-snakes

This first experiment focuses on the 2D bone segmentation problem. We define a parametrized curve patch (open), and associate to each position of the patch a correlation value. Then, we optimize the patch parameters to maximize the correlation.

To prevent the open patch from collapsing, and also to reduce the number of parameters (the smaller this number, the faster the optimization), the length of the curve is fixed, and only the normal vectors at a finite number of positions can vary.

The curve is open and is a set of $2p$ cubic Bezier patches, and then has $8p+1$ control points. The curve is parametrized by the position of the central control point M_0 and the $2p+1$ normal vectors at control points $M_{-p} \ldots M_p$ (see figure 1).

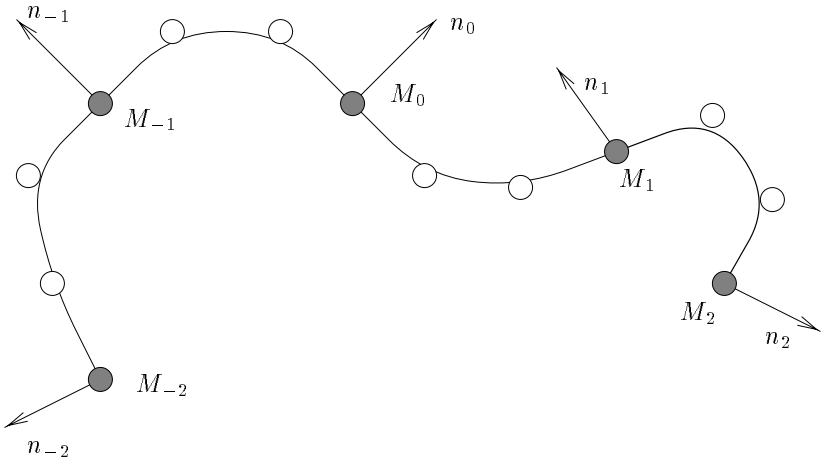

Figure 1: Curve composed of $2p$ cubic Bezier patches (here, $p = 2$). The curve is parametrized by M_0 and the unit normal vectors n_i. The positions of the other $8p$ control points are deduced from these values.

Given a length unit δ and $i = 0..p-1$, the points M_{i+1} and M_{-i-1} are

defined by:

$$\begin{aligned} M_{i+1} &= M_i + \frac{1}{2}\,\delta\,(n_i + n_{i+1})^{\perp} \qquad (1)\\ M_{-i-1} &= M_{-i} + \frac{1}{2}\,\delta\,(n_{-i} + n_{-i-1})^{\perp} \quad \text{where} \quad \begin{pmatrix} x \\ y \end{pmatrix}^{\perp} = \begin{pmatrix} y \\ -x \end{pmatrix}. \end{aligned}$$

Given a signed distance e to the curve, we can compute the variance of the image values at the $2p+1$ points $M_i + e\,n_i$:

$$\sigma^2(e) = \frac{\sum_{-p}^{p} f(M_i + e\,n_i)^2}{2p+1} - \left(\frac{\sum_{-p}^{p} f(M_i + e\,n_i)}{2p+1}\right)^2. \qquad (2)$$

We then minimize the sum of $\sigma(e)$ in a range $-e_{max} \ldots e_{max}$; optimizing on the n_i orientations, with M_0 constant. Some results obtained using this approach are shown on figure 2. The optimization is done using the package `subplex` written by Tom Rowan, available on the `GAMS` software repository. We had to add rigidity terms to the optimization.

The irregularity we can observe on the normal directions comes from the discretization of the image. In the next section, we propose an extension of this idea to the 3D case, using the techniques presented in section 3 to take into account the discrete nature of the image.

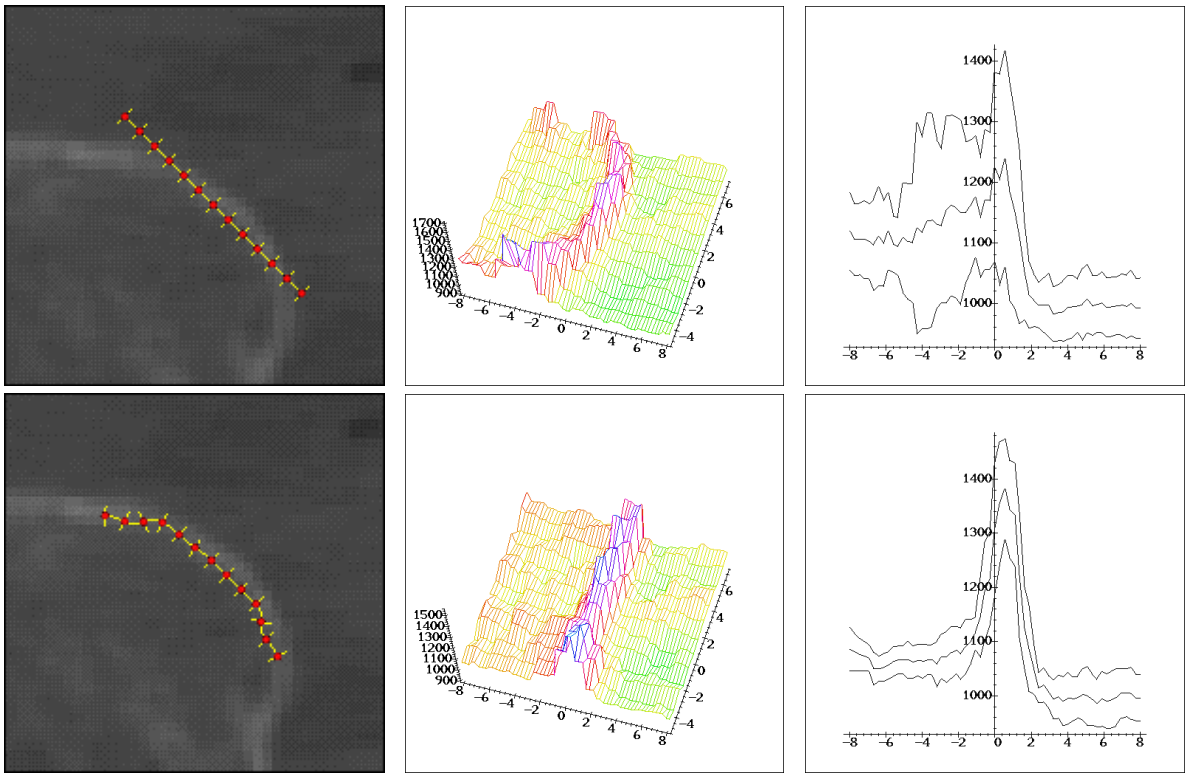

Figure 2: First and second rows: position of the snake before and after optimization. Left: curve ($p = 6$) and orientations of the 13 normals. Middle: image values among the normal directions for each point M_i of the contour (bottom horizontal axis is e in millimeters, right horizontal axis is i, and vertical axis is the image value). Right: mean and mean $\pm\sqrt{\sigma(e)}$ of the values taken at points $M_i + e\,n_i$ (horizontal axis is e in millimeters, vertical axis is the image value).

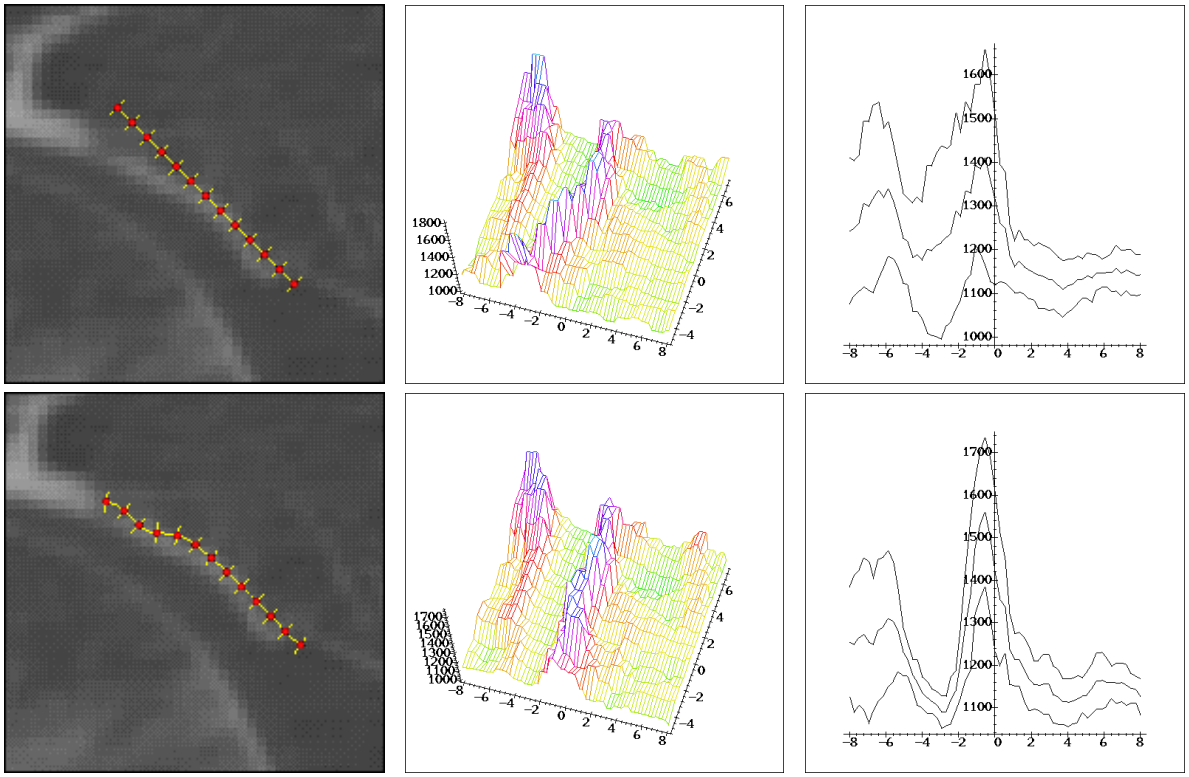

Figure 3: Same as previous figure in another test position.

4.2 3D correlation-patches

The chosen representation of the unknown surface is slightly different in this experiment. Instead of explicitly moving a surface, we will just define a scalar field and thus represent the surface implicitly. The scalar field will be defined on an area representing a first approximation of the researched contour.

Figure 4 illustrates the definition of the scalar field in 2D. Given a first C^1 guess of the contour to find (the bold central curve), we define a curve by a set of p control points C_i and, using the normals to this curve, a set of upper and lower control points U_i and L_i at a fixed distance d to this central curve. This defines a 2D map $m(u,v)$ from $[0,1]^2$ to $\mathbb{R}^2$, using Lagrange basis of orders p in u and 3 in v.

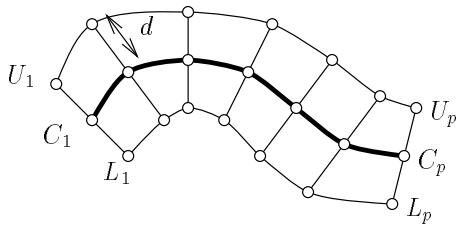

Figure 4: Definition of the implicit contour in 2D (see text).

We now add a third component to each control point, representing the signed distance of the control point to the implicit contour. Only the third component s_i of the C_i control points is a parameter of the contour ; the values of the third component of U_i and L_i are imposed to be respectively $s_i + d$ and $s_i - d$.

The map $m(u,v)$ is now defined from $[0,1]^2$ to $\mathbb{R}^3$, with components $X = (x,y)$ and s. The scalar field s of the "distances" to the implicit contour is then parametrized by the p values s_i and defined implicitly inside the patch by :

$$\forall U \in [0,1]^2, \qquad s(m(U)_X) = m(U)_s. \tag{3}$$

The 3D case is defined the same way. We first have a C^1 local estimate of the surface to find, and define a map from $[0,1]^3$ to $\mathbb{R}^4$ with coordinates (x,y,z,s), using $3pq$ control points (see figure 5). The scalar field is defined by the pq values of the s coordinate at the control points of the central layer of the 3D patch. Let S denote the vector of these pq parameters, and $s(S,X)$ the distance field defined for X inside the 3D patch.

Let V be the set of voxels located entirely inside the patch, and $f(v) \in \mathbb{R}$ denote the gray level value of voxel $v \in V$. v is a subset (indeed a box) of $\mathbb{R}^3$.

We propose two energy terms for the optimization. Both of these terms are integrals on the patch. We discretize the distance interval $[-d,d]$ into a set

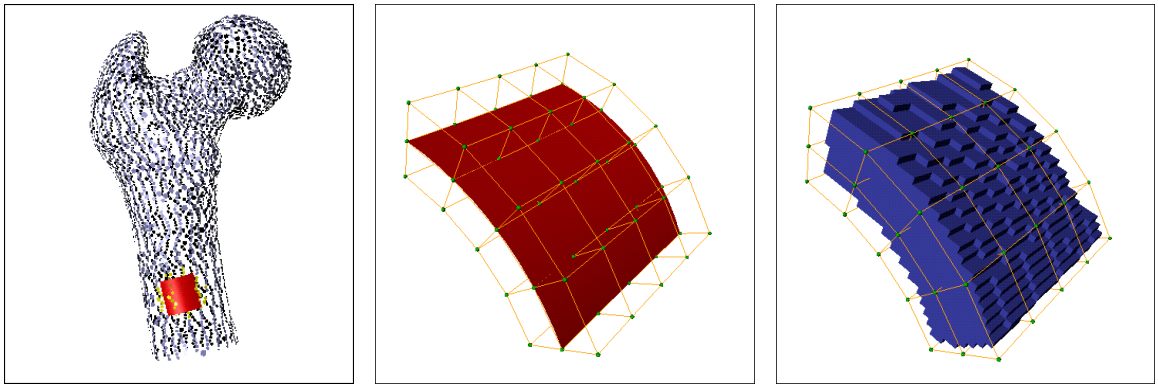

Figure 5: (left) Localization of the patch on the femur. (middle) Control mesh of orders $p = q = 5$ build over a surface patch. (right) Set of voxels V located inside the patch.

of intervals T. For example, we may choose regular intervals, for an integer r :

$$T = \{[-d, -(r-1)d/r], [-(r-1)d/r, -(r-2)d/r], \ldots [(r-1)d/r, d]\}. \tag{4}$$

Let $\mu(S, v, t)$ be the relative volume (in $[0, 1]$) of the intersection of voxel $v \in V$ and the "slice" of 3D space defined by $s(S, X) \in t$ for a distance interval $t \in T$. If the vertex v lies entirely inside the slice t, $\mu(S, v, t)$ will be 1, and if the vertex does not intersect the slice, it will be 0.

4.2.1 Sum of variances

The first energy is defined as in the previous section as the sum on all slices of the variance of the image values in the slice.

Let $\sigma_k(S, t) = \sum_{v \in V} \mu(S, v, t) f(v)^k$ for $k = 0, 1, 2$, and $t \in T$. The energy term is then :

$$E_1(S) = \sum_{t \in T} \frac{\sigma_2(S, t)}{\sigma_0(S, t)} - \left(\frac{\sigma_1(S, t)}{\sigma_0(S, t)} \right)^2 . \tag{5}$$

4.2.2 Distance to the closest profile

Let h be a function defined from T to $\mathbb{R}$ and giving an estimate of the value of the image in the distance slice t. We want to minimize the square distance between the real voxel values and the values obtained using this hypothesis function.

The value of pixel v deduced from hypothesis h is $\sum_{t \in T} \mu(S, v, t) h(t)$. The energy term is then :

$$E_2(S) = \sum_{v \in V} \left(f(v) - \sum_{t \in T} \mu(S, v, t) h(t) \right)^2 . \quad (6)$$

4.2.3 Implementation issues

We first build the set of voxel corners (8 points/voxel) interior to the patch, and find for each of these points the parameter values in the patch (a vector of $[0, 1]^3$). This requires for each point the resolution of a polynomial system of equations. This step takes between 40s and 80s of CPU time on a SGI Indy R5000 at 150MHz, for all the femur examples shown. In the case of figure 5, there are 3598 voxel corners inside the patch.

Then, we build the set of voxels interior to the patch (having their 8 corners inside the patch), and store for each voxel the references of the 8 corners in the previous set. In the case of figure 5, there are 2563 voxels inside the patch.

To compute either of the energies defined, we must first, for a given value of S, compute the s component value at each of the interior voxel corners. These values depend linearly of the vector S, and this step is just a matrix-vector product. Then, we now the value of s at the 8 corners of each voxel. We use a combination of subdivision and bilinear interpolation (in directions x and y), and linear interpolation (in direction z, the largest dimension of the voxel) to compute the values of μ for the interval of slices intersected by the voxel. Only a few number of slices intersect a given voxel.

4.2.4 Results

Figure 6 shows the result of the optimization of the first energy form on two different patches of the femur. Each optimization took about 2 minutes of CPU time on an SGI Indy R5000 at 150MHz. The patches have an approximate dimension of 8mm and a thickness of 3mm ; they are currently build from a set of hand-segmented slices of the same scanner image. No regularization term were added to the energy in these cases.

5 Conclusion

We have presented the problem of segmentation of bones in medical images, and given some references about existing solutions to the problem of medical image segmentation.

Using the optimization techniques used in computer vision to localize features at a subpixel accuracy, we have presented experiences on 2D and 3D

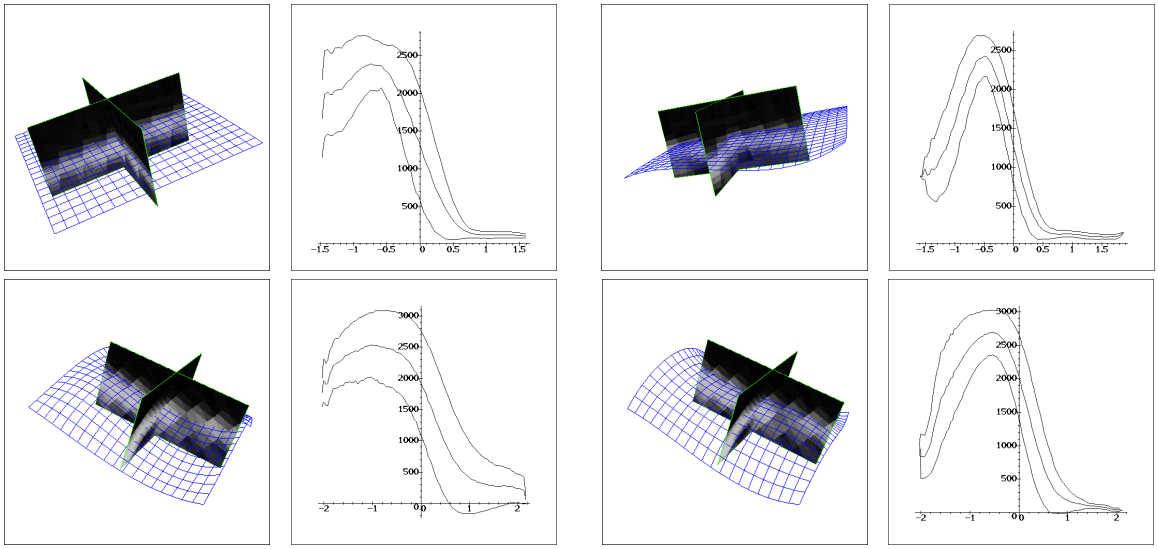

Figure 6: Each row shows an optimization process on a different patch. From left to right in each row: initial iso-surface, initial gray level profile (mean, mean $\pm$ standard deviation), final iso-surface, final gray level profile

segmentation of bone surfaces at a subvoxel resolution, starting from an existing estimate model of the surfaces.

Future work include the creation of a generic patch set for a given bone (femur, pelvis,...), and the deformable matching of this generic surface model and a 3D image to segment. This matching would give a set of patches representing a first estimate of the surface to segment, used to initialize the subvoxel optimization process.

References

[1] Maria-Helena Algorri and Francis Schmitt. Surface reconstruction from unstructured 3D data. *Computer graphics forum*, 15(1):47–60, 1996.

[2] D. Attali. r-regular shape reconstruction from unorganized set of points. In *13th ACM Symposium on Computational Geometry*, Nice, June 1997.

[3] E. Bittar, N. Tsingos, and M-P. Gascuel. Automatic reconstruction of unstructured 3D data: Combining a medial axis and implicit surfaces. *Computer Graphics forum (Eurographics'95)*, 14(3):457–468, August 1995.

[4] J. Bloomenthal. Polygonization of implicit surfaces. *Comput. Aided Geom. Design*, 5(4):341–355, 1988.

[5] C.B. Bose and I. Amir. Design of fiducials for accurate registration using machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1196–1200, December 1990.

[6] Pascal Brand and Roger Mohr. Accuracy in image measure. In *SPIE Videometrics III*, volume 2350, pages 218–228, Boston, USA, 1994.

[7] Ivan Bricault and Olivier Monga. From volume medical images to quadratic surface patches. *Computer Vision and Image Understanding*, 65(2), 1997.

[8] I. Cohen, L.D. Cohen, and N. Ayache. Using deformable surfaces to segment 3D images and infer differantial structures. *CVGIP: Image understanding*, 56(2):242–263, 1992.

[9] H. Delingette. Decimation of isosurfaces with deformable models. In Troccaz et al. [38], pages 83–92.

[10] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and vision computing*, 10(3):132–144, April 1992.

[11] Matthias Eck, Tony DeRose, Tom Duchamp, Hughes Hoppes, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. *SIGGRAPH'95*, pages 173–182, August 1995.

[12] Matthias Eck and Hughes Hoppes. Automatic reconstruction of B-spline surfaces of arbitrary topological type. *SIGGRAPH'96*, pages 325–334, 1996.

[13] A. Efrat and C. Gotsman. Subpixel image registration using circular fiducials. *International Journal of Computational Geometry and Applications*, 4(4):403–422, December 1994.

[14] Leitner F. and P. Cinquin. Complex topology 3D objects segmentation. In *SPIE Conference Vol. 1609*, pages 708 – 713, Boston, November 1991.

[15] B. Geiger. Three-dimensional modeling of human organs and its application to diagnosis and surgical planning. Technical Report 2105, Institut National de Recherche en Informatique et Automatique (INRIA), Sophia-Antipolis, December 1993.

[16] André Guéziec. Surface simplification with variable tolerance. In *MRCAS'95*, pages 132–139, Baltimore, November 1995.

[17] D.I. Havelock. The topology of locales and its effects on position uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):380–386, 1991.

[18] A Hill, T.F. Cootes, and C.J. Taylor. Active shape models and the shape approximation problem. In D.Pycock, editor, *British Machine Vision Conference BMVC'95*, Birmingham, 1995.

[19] Hughes Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean Schweitzer, and Werner Stuetzle. Piecewise smooth surface reconstruction. *SIGGRAPH'94*, pages 295–302, July 1994.

[20] Hughes Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH'92*, 26(2):71–77, July 1992.

[21] Hughes Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. *SIGGRAPH'93*, pages 19–26, August 1993.

[22] F. Kammoun and J. P. Astruc. Optimum edge detection for object-background picture. *Computer Vision, Graphics and Image Processing : Graphical Models and Image Processing*, 56(1):25–28, January 1994.

[23] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[24] M. Kisworo, S. Venkatesh, and G. West. Modeling edges at subpixel accuracy using the local energy approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):405–409, April 1994.

[25] J.O. Lachaud and E. Bainville. A discrete adaptative model following topological modifications of volumes. In *Discrete Geometry for Computer Imagery*, Grenoble, France, September 1994.

[26] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3D surface reconstruction algorithm. *ACM SIGGRAPH'87*, 21(4):163–169, 1987.

[27] Tim McInerney and Demetri Terzopoulos. Medical image segmentation using topologically adaptable surfaces. In Troccaz et al. [38], pages 23–32.

[28] Olivier Monga and Serge Benayoun. Using partial derivatives of 3D images to extract typical surface features. *Computer Vision and Image Understanding*, 61(2):171–189, March 1995.

[29] Bernard Peuchot. Camera virtual equivalent model 0.01 pixel detectors. *Computerized medical imaging and graphics*, 17(4/5):289–294, 1993.

[30] M. R. Shortis, T. A. Clarke, and T. Short. A comparison of some techniques for the subpixel location of discrete target images. In *SPIE Videometrics III*, volume 2350, pages 239–250, Boston, USA, 1994.

[31] G. Subsol, J.-P. Thirion, and N. Ayache. A general scheme for automatically building 3D morphometric anatomical atlases: application to skull atlas. In *MRCAS'95*, pages 226–233, Baltimore, November 1995.

[32] Gábor Székely, András Kelemen, Christian Brechbühler, and Guido Gerig. Segmentation of 2D and 3D objects from MRI volume data using constrained elastic deformations of flexible fourier contour and surface models. *Medical image analysis*, 1(1):19–34, 1996.

[33] Gabriel Taubin. An improved algorithm for algebraic curve and surface fitting. In *Fourth international conference on computer vision ICCV'93*, pages 658–665, Osaka, June 1993.

[34] D. Terzopoulos, J. Platt, A. Barr, and K. Fleisher. Elastically deformable models. *Computer Graphics (proceedings SIGGRAPH)*, 21(4):205–214, 1987.

[35] Jean-Philippe Thirion and Alexis Gourdon. The marching lines algorithm: new results and proofs. Technical Report 1881, Institut National de Recherche en Informatique et Automatique (INRIA), Sophia-Antipolis, April 1993.

[36] Jean-Philippe Thirion and Alexis Gourdon. Computing the differential characteristics of isointensity surfaces. *Computer Vision and Image Understanding*, 61(2):190–202, March 1995.

[37] Q. Tian and M.N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image Processing*, 35:220–233, 1986.

[38] J. Troccaz, E. Grimson, and Ralph Mösges, editors. *First joint conference in computer vision, virtual reality and robotics in medicine and medical robotics and computer-assisted surgery CVRMed–MRCAS*, number 1205 in Lecture notes in computer science, Grenoble, France, March 1997. Springer.

[39] R. J. Valkenburg, A. M. McIvor, and P. W. Power. An evaluation of subpixel feature localisation methods for precision measurement. In *SPIE Videometrics III*, volume 2350, pages 229–238, Boston, USA, 1994.