Comparison of two techniques for organ reconstruction using Visible Human Dataset

Frédéric CORDIER and Nadia MAGNENAT THALMANN
MIRALab, C.U.I., University of Geneva, Switzerland
(Email: [thalmann|cordier]@cui.unige.ch)

Abstract: Though the Visible Human Dataset provides a complete volumic coverage of the human body, it does not give information about its structure. An important task is to identify the spatial extents of the organs of interest, and to provide their surface models. In this paper, we will present two methods for the reconstruction of organs from the Visible Human Dataset. The first method is the segmentation of organ on each slice with the snake technique. The second method uses the Shape Constrained Deformable Model. For these two methods, we present testing results of the extraction of different types of organ from the Visible Human Data.

1. Introduction

A tissue of the human body can be shown in three dimensions, using sequences of two-dimensional slices obtained by medical diagnostic equipment such as Computed Tomography (CT) and Magnetic Resonance Imagery (MRI). The Visible Human Dataset provides such data in very high quality. These images make it possible to simulate a complicated surgical procedure, educate medical students, and visualize the condition of a patient. As the data of CT or MRI forms volumetric data from a sequence of cross-sections, it is necessary for the effectiveness of the simulated medical procedure to create a 3D model for each object using the volumetric data, thus allowing it to be manipulated [16][17][18][19][20].

Many researchers have extracted 3D models for surgical simulation, using image-processing software [21]. Techniques such as Marching Cubes and Region Growing have been tested. J. Montagnat and H Delinguette [6] use deformable models to extract shape organs from medical images. Their method combines the deformable model approach with the registration approach. This hybrid model of deformations consists of an additional external force acting on a 3D model. This force is derived from the best geometric transformation between the actual model shape and the closest data points. Similarly, researchers have improved the robustness of deformable models by applying more global constraints. Terzopoulos and Metaxas have considered in [9][11] the superposition of a rigid component with a finite element mesh. A similar approach on deformable contours is proposed in [8]. Modal analysis [10][12] and Fourier representation [13][14] aim similarly at controlling the amount of deformation.

In this paper, we present two techniques for organ reconstruction. In the first part of this paper, we describe a method which is a semi-automatic reconstruction by defining organ contour on each slice. On these slices, the user defines the organ contour with the snake technique. Once all contours have been defined, these contours are joined to create the surface of the organ. In the second part of this paper, we present another method that uses Shape Constrained Deformable Model. The idea of this method is that, among patients, shapes of organs are most of the time, similar. The reconstruction process is defined in two steps. First, the user has to place an initial shape of the organ to the position of the organ to be reconstructed. Then, this initial shape is deformed using the image information and the shape memory. The shape memory of a model is its ability to come back to its original form once deformed. The image information is used to move vertices of the model to image contours. For these two methods, we present in the third part results and problems we have encountered for reconstruction of organs from the Visible Human Dataset.

2. Reconstruction with Snake technique

With this technique, the reconstruction process is defined in two steps described as follow:

- At the first step, the user defines the organ contour on each slice. For that, he/she has to put a set of points around the organ on the slice. The snake technique allows the polygon to fit exactly to the contour on the slice. This operation has to be done for every slice containing the organ.
- Once the elements have been interpreted and labeled on the different cross-sections as contours, the 3D reconstruction is performed by joining the contours of all the pertinent slices [5]. This gives a 3D surface of the anatomical object of interest.

In following paragraphs, we give a short description for the snake technique, the program which implements this technique and in finally, the results for the reconstruction of different types of organ.

2.1. Snake model

For exact fitting of the contour of the organ on a slice we use discrete snakes [22][23]. This allows the fitting of the contour to points of maximum contrast close to the already-defined rough contour. Though snakes are quite useful when anatomical information is visible, manual intervention is sometimes necessary due to the difficulties in interpretation.

Snakes are characterized by a few parameters (such as elasticity, rigidity, and speed,). Because manipulation of all parameters may burden users, they are provided with predefined settings dedicated to a particular contouring problem (skin, bones…). Thus, only a single command is necessary to achieve contour fitting. As our contours are modeled as closed polygons, we use a discrete snake model. In addition to elastic and rigid forces, an image force acts on each vertex and tends to move it to pixels of interest. To adjust the snake on points of strong contrast, we consider:

$$\mathbf{F}_{ext,i} = \mathbf{n}_i \cdot \nabla E(\mathbf{v}_i)$$

where $\mathbf{n}_i$ is the normal to the curve at the node i, whose position is $\mathbf{v}_i$. E is a measure of the contrast of the image at point $\mathbf{v}_i$ and is given by:

$$E(\mathbf{v}_i) = \left| \nabla I(\mathbf{v}_i) \right|^2$$

where $I(\mathbf{v})$ represents the image itself. To estimate the gradient of the image, we use a 3D generalization of the Sobel operator [24]. We consider all 26 neighbors to the current voxel. This combines edge detection and low-pass filtering, and therefore snake vertices are less subject to be stuck by a slight feature. Using neighboring slices re-enforces structures present in successive slices. Nonetheless this may confuse the user by fitting the contour to structures not visible in the current slice. To avoid this the weight of the gradient component transverse to the slice is tunable and may even be neglected.

Blending of the different color channels is changed to alter the snake's color channel sensitivity. For instance, we can make snakes sensitive to the excess of blue color, which is useful with our blue background. Skin can then be easily contoured.

Snakes are useful in many circumstances, particularly in the presence of high contrast such as bones on CT images. It helps users, but does not replace them, since the user validation of labeling is an important task.

2.2. Implementation

A program "Label" has been written to implement the technique described earlier. Label is composed of two windows. The first window shows a 3D view of the voxel map with organs under reconstruction. The voxmap is represented by its bounding box and a plan inside this bounding box. This plan corresponds to a slice inside the voxmap. A scrollbar allows the user to move this plan. The second window shows a 2D view of the current slice with the cross-sectional contour of the object corresponding to this slice. With this window, the user can add, delete or move points of the cross-sectional contour on the current slice.

The Label program has been written using the MFC library from Microsoft [26] and the OpenInventor library from TGS [25]. The following picture shows the user interface of Label.
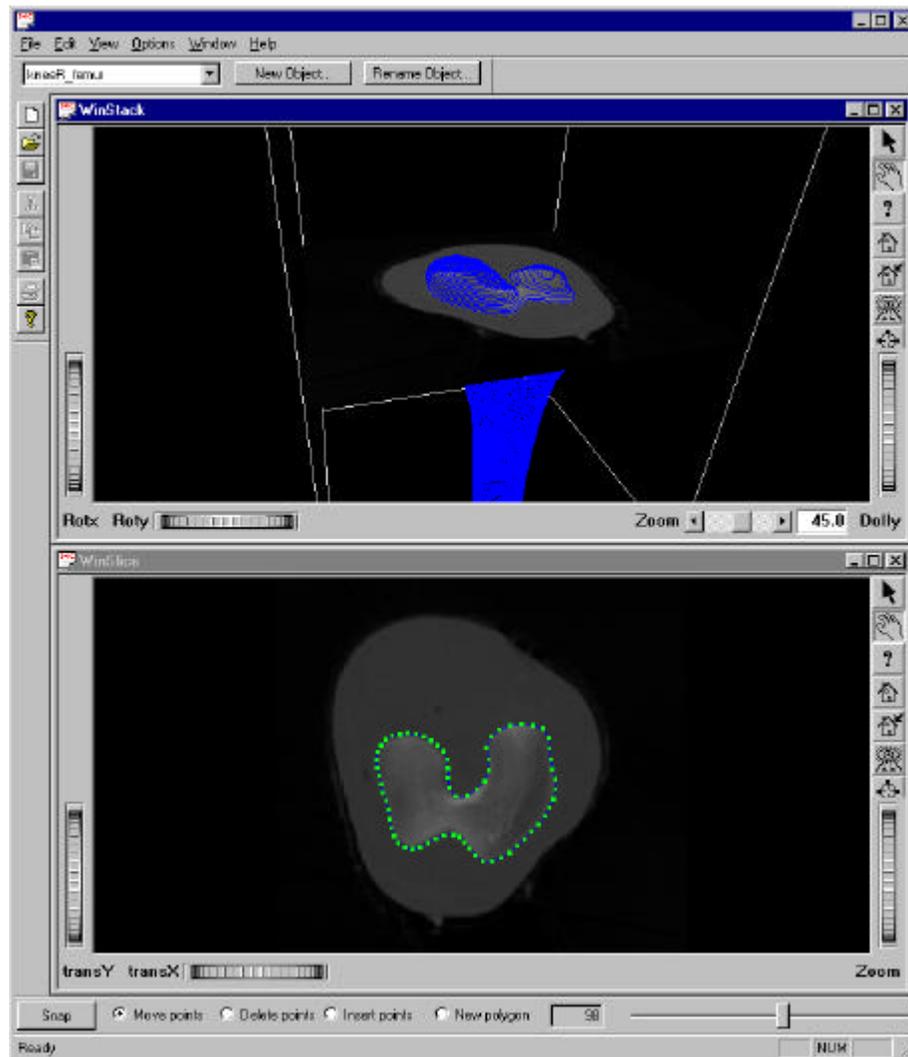

Figure 1: user interface of Label.

3.   Shape Constrained Deformable Model

As described early, the reconstruction process with snakes can be slow because the user has to define the cross-sectional contour on each slice. As organ boundaries do not appear clearly in some cases, reconstruction with snake also requires anatomical knowledge. Therefore another technique has been

developed. This technique uses Shape Constrained Deformable Model. In the next section, we will give a description of our method, its implementation and the results. Our method has been tested to reconstruct organs of the Visible Female with organs from the Visible Male.

3.1. Description of the Shape Constrained Deformable Model.

In this method, we use an initial model that is placed approximately to the position of the organ we want to reconstruct. This model is modified using a generic model and the information from the medical images of the patient. The segmentation process is formulated as the minimization of a cost functions associated to each point of the shape:

♦ A shape memory force that keeps the shape of the generic model.
♦ Image interest force that tends towards vertices to move into high edge density.

In the next sections, we first present the generic model, then describe different types of forces, and finally, the energy minimization process followed by the results.

3.1.1. *Initial shape and generic shape*

As described earlier, our method uses two shapes: the initial shape and the generic shape. The initial shape is the shape at the first iteration in the computing process. We can use an initial shape which is different from the generic shape. The only restriction is that initial shape and generic shape must have the same topology. Most of cases, initial shape is identical to generic shape. Sometimes for specific images as CT images, it's very helpful to define an initial shape. On these images, bones appear with two contours very close to each other. During the reconstruction process, some vertices move to the internal contour and some other to the external contour. In that case, we define an initial shape big enough to contain the entire organ to be reconstructed. In this way, vertices only move to the external contour. A generic shape is used to compute the shape memory force which will be described later.

3.1.2. *The deformable model*

Our model is represented by points that define the object contours. The surface of the object is defined by a triangular mesh where each vertex knows its neighboring vertices. Thus, reconstruction algorithms such as the labeling tool described earlier can directly be used in a first step to build the generic model. The vertices of the mesh triangles correspond with voxels in the volume being segmented.

3.1.3. *Cost functions*

These cost functions can be defined as a set of forces applied to each vertex. As described earlier, there are two types of force, shape memory force and image interest force. All these forces are explained in next two sections.

3.1.3.1. *Shape memory force*

The elastic properties of an object can be defined as its ability to come back in its original shape once deformed. It corresponds to a force, called shape memory force that is deduced from the generic shape and the current shape. It is different from the elastic model that uses spring to link each neighboring point and it offers better convergence. A triangular mesh defines the surface of the object and each vertex knows its neighborhoods. For each vertex, we use an approximate coordinate system calculated with neighboring vertices. The computing of the shape memory force is made into two steps described as follow:

- During initialization, we calculate for each vertex the local coordinate system with the generic model. In this local coordinate system, we compute the coordinates of the vertex. This is the position of the local null deformation of the vertex relative to its neighbors.
- For each iteration, we calculate for each vertex the local coordinate system relative to its neighbors. This local coordinate system changes at each iteration because the shape is modified during the fitting process. In this local coordinate system, we know the current position of the vertex and position of the local null deformation. This position has been computed in the initialization step. Between these two positions, we apply a force that will move the current position to the position of the local null deformation. The local shape tends towards to be similar of the local shape of the generic model.

In the following figure:

- G is the isobarycenter of the neighbors of point P.
- Z is the unit vector normal to the approximate neighbors plane. X and Y are two vectors normal to Z. *(X,Y,Z)* is an orthonormal coordinate system.
- *Pcurrent* is the current position of the vertex. This position is defined by three values *xc*, *yc* and *zc*.
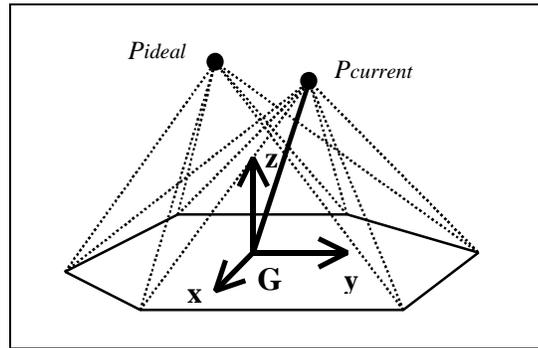- *Pideal* is the position of the local null deformation.

Figure 2: The local coordinate system.

With these two positions, the value of the shape memory force applied to each vertex P is:

$$\vec{F} = K \cdot \overrightarrow{P_{current}P_{ideal}}$$

K is the coefficient of the internal force. By increasing this coefficient, the elasticity of the deformable model will be lower.

The length of the vector Pideal is proportional to the size of the generic model. It means that if we apply a global scaling on the generic model, the direction of Pideal for all vertices will be invariant and the variation of the length will be equal to the variation of the model size. In that way, a coefficient can be added to Pideal in the previous equation. With this coefficient, we can add a force that inflates or deflates the deformable model during the fitting process. This force can be used on a closed contour to force the contour to expand in the absence of external influences.

3.1.3.2. Image interest force

External forces attract the deformable contour to interesting features, such as object boundaries, in an image. Any force expression that accomplishes this attraction can be considered for use. In our case, the computation of external forces is defined in two steps:

♦ At the initialization, we apply a 3D Sobel operator on the entire voxmap. The gradient map is then stored in a cache. Using a memory cache is a good strategy because we intensively use the value of the gradient for each vertex in the next process. The input voxmap can be represented as gray level data as well as color.

♦ During the process, we find the closest position of the gradient value which is greater or equal to the threshold parameter. This position is considered as ideal position. The threshold parameter is controlled by the user. In order to limit time, the research for the ideal position is restricted along the normal of the surface around the vertex. This normal of the surface has already been computed during the computation of the internal force. This normal is the Z vector in the local coordinate system described earlier. A parameter defines the size of the neighbor. This parameter can be edited by the user. We put a force to move the vertex to this ideal position.

On the following figure:

♦ *Pi1Pi2* is the segment on which the search of the ideal position is done.
♦ *Ni* is the normal to the surface
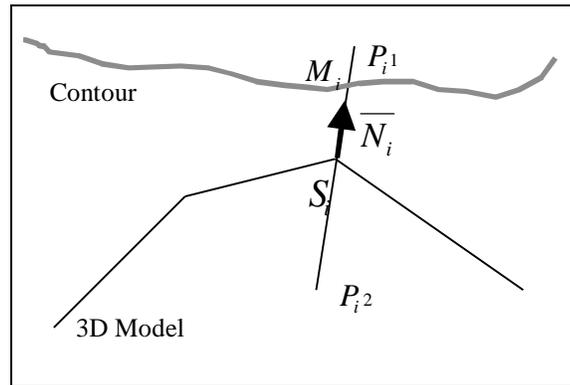♦ *Si* is the vertex and *Mi* is the position of the ideal position.

Figure 3: Computation of ideal position.

The expression of the external force is similar to internal force. The value of this force applied to a vertex *Si* is:

$$\vec{F}_i = K \cdot \overline{S_i M_i}$$

The parameter K is the coefficient of the external energy. This parameter can be modified by the user. Since we are considering noisy data, we may perform a smoothing of external forces by taking the average over a neighborhood. The expression of the smoothed force applied on *Pi* is:

$$\vec{f}_{ext}(P_i) = \frac{1}{\left| N_t(P_i) \right|} \cdot \sum_{P_j \in N_t(P_i)} \vec{f}_{ext}(P_j)$$

Nt(Pi) is the neighbor vertices of the vertex *Pi*. It can be defined as the set of points connected to *Pi* by a topological path of length lesser than or equal to *t*. The parameter *t* can be modified by the user. If this parameter is equal to zero, the smoothing is not used in the external force computation.

For each vertex, we can compute the total energy which is the sum of internal and external energy. We then use minimization of the total energy using variable metric methods in multidimensions [15]. The minimization process is made in several iterations. Between each iteration, the user can intervene to modify the deformable model.

3.2. Implementation

The software which implements this method has been developed on SGI machines. The user interface is composed of a main window and a dialog to set the parameters of the fitting process. The main window shows the voxmap with the deformable model. The user can modify the model in two ways. He can make some global transformation as translation, rotation or scaling. He can also modify a part of the model by moving a set of vertices at the surface. This modification can be done on the initial shape but also during the fitting process. The following figure shows a snap shot of the user interface.
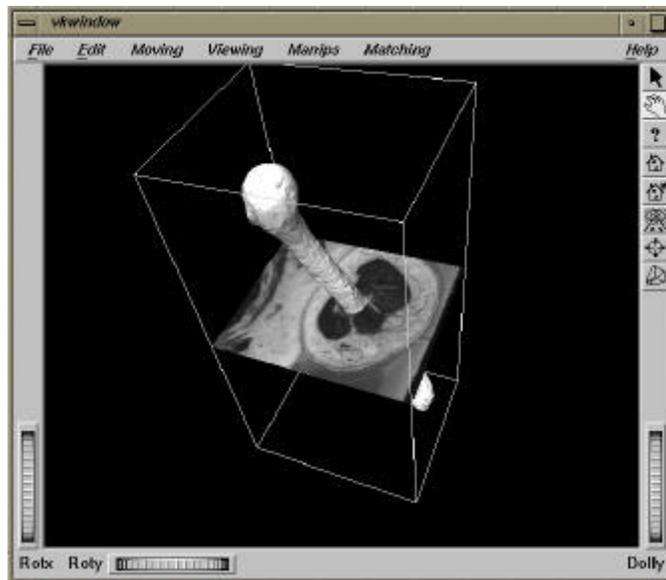

Figure 4: User interface.

4.   Results

In the following sub-sections, we provide a comparison of the reconstruction from the two techniques described early. There are primarily three types of anatomical elements (organs) that can be reconstructed: bones, muscles and skin with deep fascia. For these three elements, we describe the advantages and disadvantages for each reconstruction method.

4.1. Bones

For the reconstruction of bones, CT images and anatomical images have been used. Reconstruction of the bones on anatomical cross-sections is easy when they are surrounded by musculature, the white color of the compact bone contrasts well with the red one of the skeletal muscular tissues. In this case, reconstruction with Label, the first method presented in this paper, is almost automatic. Also the method that uses Shape Constrained Deformable Model gives good results and is faster than Label.

In the case where tendons and fibrous articular capsules cross or attach to bones, delineation of the osseous surface may become difficult. With Label, the user has to define the arbitrary boundary on each slice. Segmentation with Label becomes very slow. The second tool can interpolate missing boundaries as described early. In some case, the interpolation is not precise enough; the user can modify the surface by moving a set of vertices to the contour defined arbitrary. Generally speaking, the second tool is faster than Label for the reconstruction of bones on anatomical images. The first image below shows the segmentation of the humerus with Label. For the next two images we can see the result of reconstruction from anatomical female images with shape constrained deformable model. The generic model used for this reconstruction has been created with Label from anatomical male data.
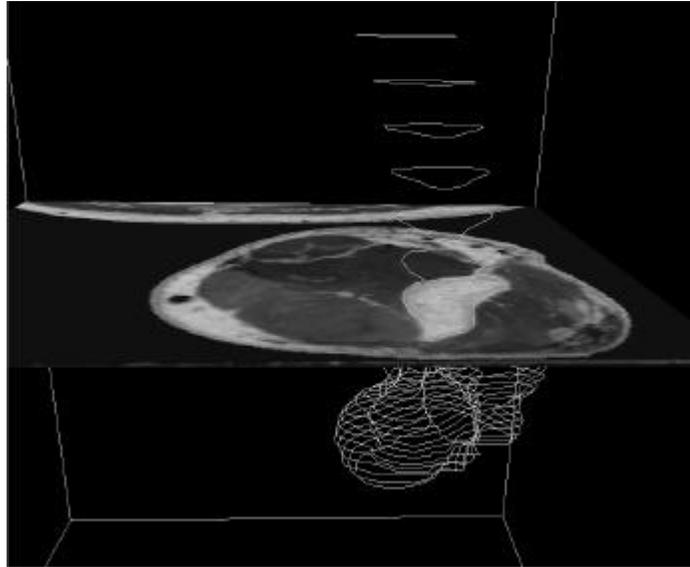

Figure 5: Reconstruction of the humerus with Label from anatomical male images.
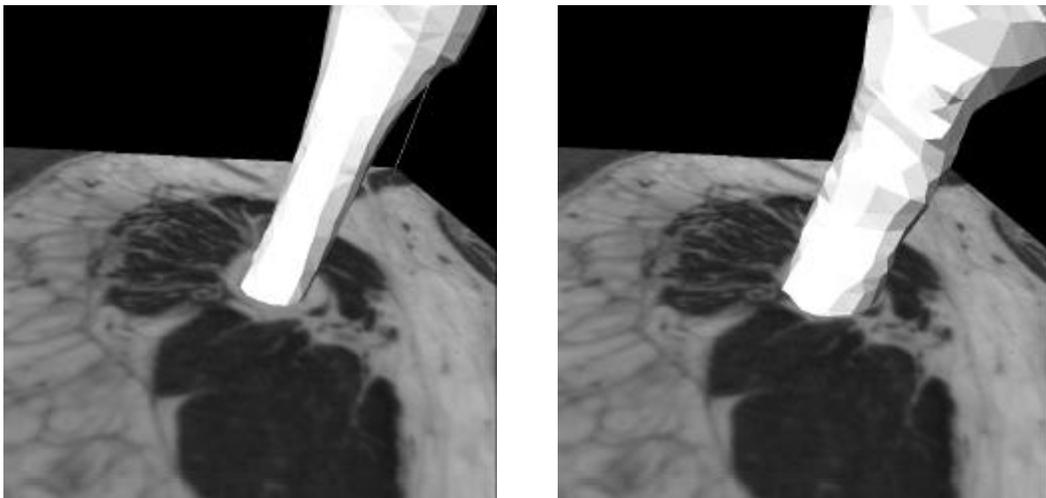

Figure 6: Reconstruction of the Humerus from anatomical images.

On CT images, distinction between calcified tissue and non-calcified fibrous structures is easier. Comments on reconstruction from these data are the same than comments on reconstruction from anatomical data. The only difficulty we have encountered is that on these images, bones appear with two

contours. These contours are very close to each other. During the reconstruction process with the second tool, some vertices of the deformable model move to the internal contour and others to the external contour. To solve this problem, we need to define an initial model enough bigger to contain the entire organ in the voxmap. In that way, vertices only detect the external contour. The figure below shows an example of a 3D reconstruction of a muscle. The first image represents the 3D model before the fitting and the second image shows the result of the fitting process.
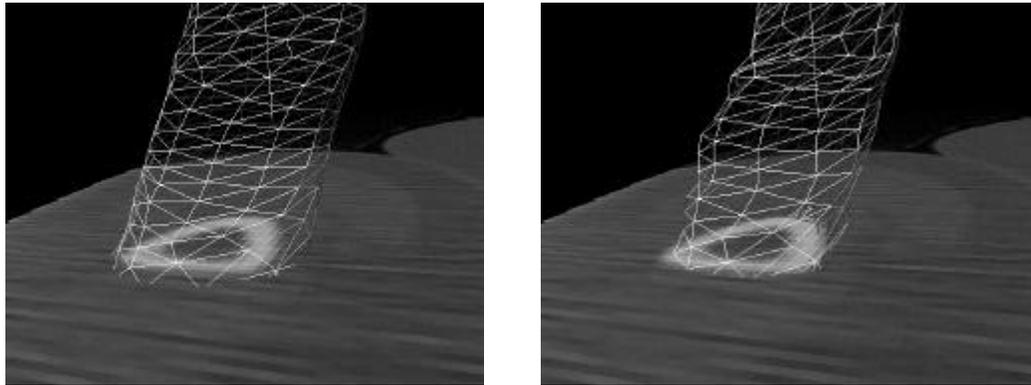

Figure 7: Reconstruction of the Humerus with the second tool from CT images.

4.2. Muscles

Segmentation of muscles is more difficult than bones. In most cases, muscles are surrounded by other muscles. The distinction between individual muscles is often impossible. Thus, it would be illusory to believe that the contour of every muscle can be clearly and easily identified.

Reconstruction of muscles with Label is very slow. The user has to use the different fiber orientation on anatomical images. In the case where delineation is impossible, the user has to arbitrarily trace a border between muscles where it is expected based on anatomical experience. The user then reconstructs the supposed muscles and judges the correctness of the segmentation with the aid of 3D rendering. When not satisfied, he/she returns to the slices and improves the segmentation until 3D rendering corresponds to the usual muscle size and shape. This segmentation process is very slow.

Reconstruction with shape constrained deformable model is easier. As described early, this method uses a generic model to interpolate missing boundaries. The reconstruction gives good results if the initial shape is closed to the shape of the muscle be reconstructed. In that case, extrapolation works fine and the segmentation can be done in a few minutes. The two following images illustrate a reconstruction of the Brachialis with the second tool.
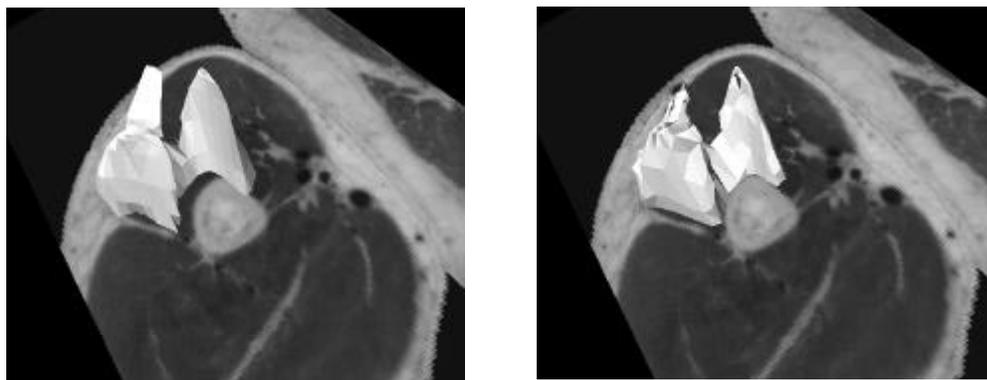

Figure 8: Reconstruction of the Brachialis from anatomical data.

4.2.1. Skin

The segmentation of the skin surface does not impose problems due to the fact that the embedding medium (blue) is in clear contrast to the white color of the skin. With the automatic contouring feature of Label, the reconstruction of skin on anatomical images is very fast. The second tool is not adapted to the segmentation of the skin. The shape of the skin surface depends on the posture of the body. It is impossible to define a generic model of the skin surface. The surface skin can be easily segmented with a technique such as Marching Cubes or Region Growing.

5. Conclusion

The two tools presented in this paper are complementary. The Label tool allows the user to create all generic models needed. Then, these generic models are used by the second tool to reconstruct organs for specific patient. The second advantage of these reconstruction tools is that they can employ different type of images as input. They can reconstruct organs from CT images or MRI images. The reconstruction can be adapted to all other types of data; the user just needs to adapt parameters used in the fitting process. The reconstruction can also be done on every other part of the human body.

6. Acknowledgements

7. Reference

[1] Akermann Michael J (1995), The Visible Human Project, http://www.nlm.nih.gov.
[2] P.Beylot, P. Gingins, P Kalra, N. Magnenat Thalmann, W. Maurel, D. Thalmann and J. Fassel (1996), 3D Interactive Topological Modeling using Visible Human Dataset in Eurographics '96, Blackwell Publishers, pp. C33-C44.
[3] P. Kalra, P. Beylot, P. Gingins, N. Magnenat-Thalmann, P. Volino, P.Hoffmeyer, J. Fasel, F.Terrier (1995), Topological Modeling Of Human Anatomy Using Medical Data, Proc. Computer Animation '95, IEEE Computer Society Press, pp. 172-180.
[4] P. Gingins, P. Kalra, P. Beylot, N. Magnenat-Thalmann, J. Fasel (1996), Using VHD to build a Comprehensive Human Model, The Visible Human Project Conference Proceedings, (CD-ROM)
[5] Geiger B (1993), Three Dimensional Modeling of Human Organs and Its Application to Diagnosis and Surgical Planning, Ph.D. Theses, INRIA, France.
[6] J. Montagnat and H. Delinguette (1997), Volumetric Medical Images Segmentation Using Shape Constrained Deformable Models in CVRMed-MRCAS'97, Lecture Notes in Computer Science, Springer Verlag, 1205, pp. 13-22.
[7] E. Promayon, P. Baconnier, C. Puech (1996), Physically-Based Deformations Constrained in Displacement and Volume in Eurographics '96, Blackwell Publishers, pp. C155-C162.
[8] K. Lai and R. Chin. Deformable Contours: Modeling and Extraction. In Computer Vision and Pattern Recognition, IEEE Computer Society Press, 1994.
[9] D. Metaxas and D. Terzopoulos. Constrained Deformable Superquadratique and nonrigid Motion Tracking. In Computer Vision and Pattern Recognition, IEEE Computer Society Press, June 1991.
[10] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Shape Modeling and Recognition. In Patern Analysis and Machine Intelligence, IEEE Computer Society Press, volume 13, pages 715-729, July 1991.
[11] D. Terzopoulos and K. Fleisher. Deformable models. Visual Computer, Springer Verlag, 4: 306-331, 1988.
[12] B. Vemuri and A. Radisavljevic. From Global to Local, a Continum Of Shape Models with Fractal. In European Conference on Computer Vision, Springer Verlag, 1993.

[13] L. Staib and J. Duncan. Deformable Fourier models for surface finding in 3D images. In Visualization in Biomedical Computing, Springer Verlag, pages 90-104, 1992.

[14] G. Székely, A. Kelemen, C. Brechbüler and G. Gerig. Segementation of 2D and 3D objects from MRI volume data using contrained elastic deformations of flexible Fourier surface models. Medical Image Analysis, Oxford University Press, 1(1): 19-34, July 1996.

[15] Numerical Recipes in C: The Art of Scientific Computing (1992), Cambridge University Press, pp. 420-425.

[16] D. T. Chen and D. Zelter, « Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element method », SIGGRAPH'92, Addison-Wesley Publishing Co., vol. 26, no. 2, pp. 89-98, July, 1992.

[17] J. Clyne, D. Middleton and K. Reining, « Computer Applications of the Visible Human Dataset », Visual Proc. SIGGRAPH'96, Addison-Wesley Publishing Co., pp. 110, 1996.

[18] E. Keeve, S. Girod, P. Pfeifle, and B. Girod, « Anatomy-Based Facail Tissue Modeling Using the Finite Element Method », Proc. Visualization'96, IEEE Computer Society Press, pp. 21-28, 1996.

[19] R. M. Koch, M. H. Gross, F. R. Carls, S. F. von Buren, G. Fankhauser, and Y.I.H. Parish, « Simulating Facial Surgery Using Finite Element Models », Proc. SIGGRAPH'96, Addison-Wesley Publishing Co., pp. 421-428, 1996.

[20] M. Bro-Nielsen and S. Cotin, Real-time volumetric deformable models for surgery simulation using finite elements and condensation », Computer Graphics Forum, Eurographics'96, Blackwell Publishers, vol. 15 no. 3, pp. 57-66, 1996.

[21] D. Quackenbush, P. Ratiu, and J. Kerr, « Segmentation of the Visible Human Dataset », The Visible Human Project Conference, pp. 69-70, 1996.

[22] Kass M, Witkin A, Terzopoulos D (1987), Snake: Active Contour Models, International Journal of Computer Vision, Kluwer Academic Publishers, 1(4): 321-331.

[23] Terzopoulos D, Waters K (1991), Techniques for Realistic Facial Modeling and Animation, Proc. Computer Animation '91, Geneva, Switzerland, Springer-Verlag, pp. 59-74.

[24] Gonzalez R, Woods R (1993), Digital Image Processing, Addison-Wesley: Reading, MA.

[25] Web site of TGS: http://www.tgs.com/

[26] Web site of Microsoft: http://www.microsoft.com/