Segmentation and Modeling of Approximately Rotationally Symmetric Objects in 3D Ultrasound*

Johannes Ruisz
Advanced Computer Vision GmbH - ACV
Vienna, Austria
johannes.ruisz@acv.ac.at

Dieter Hönigmann
Advanced Computer Vision GmbH - ACV
Vienna, Austria
dieter.hoenigmann@acv.ac.at

Helmut Pottmann
Institute of Geometry
Vienna University of Technology, Austria
helmut.pottmann@geometrie.tuwien.ac.at

Abstract

We describe a method for segmentation and modeling of approximately rotationally symmetric objects in 3D ultrasound. The proposed algorithm is based on 2D segmentation of several slices of an ultrasound volume dataset. It requires semiautomatic 2D segmentation of only two slices by an interactive process which is based on a discrete dynamic contour. A tiling algorithm leads to a 2D starting contour in every slice. A fully automatic segmentation process is then performed to all slices simultaneously. The result is a triangular mesh of the segmented object.

1. Introduction

Segmentation is a task of fundamental importance for several diagnostic problems in ultrasound imaging: E.g., in assisted reproductive therapy, repeated measurement of the size of follicles is essential for an effective treatment of infertility. In urology, diagnosis of hyperplasia of the prostate gland requires a precise estimation of its volume. In radiation therapy, prostate boundaries form the basis for treatment planning. In either case, segmentation is necessary for a quantitative analysis. Manual outlining of the contours of an organ is a tedious as well as time consuming task for 3D data sets. Thus, there is a need for a semiautomatic segmentation system, motivating recent publications on this topic.

Hu et al. present an algorithm for semiautomatic segmentation of the prostate from 3D ultrasound images [1]. The algorithm uses an initialization based on six points provided by the user. Subsequent mesh refinement using a deformable model yields the final model of the prostate. Although their algorithm produces results of relatively high accuracy (average difference in volumes was found to be 7.2%), we think that more precise results could be obtained by a different user interface based on semiautomatic segmentation of a limited number of slices of the data set, e.g., as suggested by Ghanei et al. [2]. In their approach the authors also use a deformable model of vertices forming triangle facets in the 3D space. The authors propose a method for initialization of the model from a number of initial contours, drawn on several 2D images from a stack of parallel slices. Subsequently, internal and external forces propagate the vertices iteratively until the model converges to an equilibrium state. For higher robustness and better convergence, the entire model is applied in a multiscale scheme.

The approach by Ghanei et al. is closely related to our approach. Their approach requires the user to outline an initial contour in several 2D cross-sections extracted from the data set. In our approach, depending on the shape of the organ, the number of slices used for initialization is much lower. Further, we aid segmentation in these slices by a segmentation algorithm similar to the one used for 3D segmentation. Finally, our algorithm does not require a (computationally demanding) fully 3D propagation of vertices, but propagates vertices more efficiently only within slices. Thus, we obtain results of similar accurracy with much lower computational effort. We further enhance efficiency of our algorithm by avoiding a multiscale approach. Instead, we use an *active image*, introduced in [3], an approach related to recent work by Sarti et al. [4]. The active image is a geometric model and a computational method for segmentation of images with missing boundaries. In many situations, the human visual system fills in gaps in edges and boundaries, building and completing information that is not present. Ultrasound data is especially prone of miss-

*This work had been carried out within the K plus Competence Center ADVANCED COMPUTER VISION. This work was funded from the K plus Program.

ing contours. The *active image* aims at substituting missing image information by using spatial information.

The content of this paper is structured as follows: In Section 2 we give an overview on interactive 2D segmentation of ultrasound images by our concept of an *active image*. Section 3 shows the concept of our active polygon model that builds together with the *active image* the basis of the segmentation algorithm. Section 4 introduces an efficient tiling technique to obtain starting contours in every slice used by the active polygon algorithm. In Section 5 we extend the concept of the *2D* active polygon of section 3 to the *3D* framework and finally in Section 6 we show some results to demonstrate the quality of the 3D models.

2. 2D interactive segmentation

The main feature making ultrasound images difficult to handle is their inherent noise. Although there are great differences in image quality, even the best ultrasound images contain a high level of noise and speckle artefacts. This makes it extremely difficult to perform a segmentation of ultrasound images.

Therefore it is a key idea of our approach not to base the computation directly on pure image information. Instead we build up an auxiliary function, composed of a spatial and a textural component, which we call *active image*. This *active image* is then used to perform a segmentation task by means of an active polygon.

2.1. Active Image

The *active image* is a function that determines a distance for each image pixel to a user defined position. The user input can be either a single point P or an ordered set of input points P_i (which describes a whole input region R) in the target area.!

The first component of the *active image* is the spatial component.

2.1.1. Spatial Distance Function. The spatial distance function assigns to each image pixel the square root of the shortest distance of that pixel to an input region. For our present application it is sufficient to compute an approximation of this distance function as follows:

The graph of the distance function to a fixed point is depicted in Fig. 1. The graph of the distance function of another point is obtained by translation of this function. We obtain the distance function for an input region R by computing the minimum distance to every point $P \in R$ (see Fig. 2). A sufficiently dense sequence of points is necessary to obtain a smooth function. We achieve this by an interpolatory subdivision scheme, see [5].

Reliabality and accurracy of every segmentation scheme heavily depends on the image features which drive the segmentation. Therefore, careful feature selection is necessary.

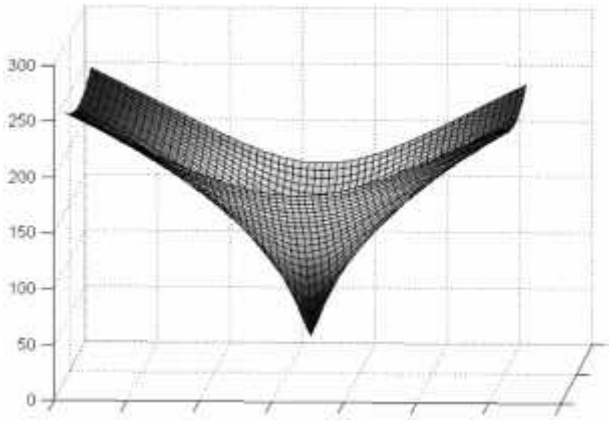


Figure 1. Graph of the square root of the distance to one single point

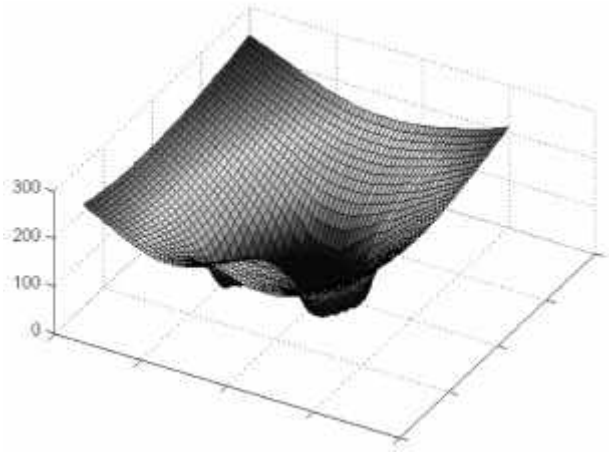


Figure 2. Graph of a the square root of the distance function to an input region R

2.1.2. Texture Component. A variety of different methods for texture extraction is avaliable, and all fit nicely into the suggested framework. A frequently-used way to extract textural features are co-occurrence matrices, originally proposed by Haralick et al. [6].

Co-occurrence matrices comprise a statistical method for describing texture in a form suitable for pattern recognition. Their evaluation provides a feature vector of properties which represents a point in a high-dimensional feature space.

Haralick et al. suggested a set of 28 textural features. Because of the close relation between some of the features we selected a subset consisting of *angular second moment*, *contrast*, and *correlation*.

After setting up the feature space corresponding to the sonographic image we now have to introduce a proper measure of similarity between the texture within the region R defined by points P_i entered by the user and the remaining image. We do this in order to infer which regions, from a textural point of view, are similar to patches within the target area and which are not. An appropriate calculation of this similarity is an essential component of the active image.

For convenience, we refer to the set of feature vectors corresponding to patches within the reference region R as $\mathbf{F}_{\mathrm{ref}}$, whereas $\mathbf{F}$ is the set of feature vectors of the entire image. Obviously, $\mathbf{F}_{\mathrm{ref}}$ is a subset of $\mathbf{F}$. Now, we aim at a distance function $d(\mathbf{f}_{\mathrm{i}}), \mathbf{f}_{\mathrm{i}} \in \mathbf{F}$ which assigns a low value to $\mathbf{f}_{\mathrm{i}}$ if $\mathbf{f}_{\mathrm{i}}$ is similar in texture to any $\mathbf{f}_{\mathrm{ref}} \in \mathbf{F}_{\mathrm{ref}}$. On the other hand, the distance function d shall obtain larger values if $\mathbf{f}_{\mathrm{i}}$ represents different texture.

In order to compute $d(\mathbf{f}_{\mathrm{i}})$ in an efficient way, we translate $\mathbf{F}$ such that the barycenter of $\mathbf{F}_{\mathrm{ref}}$ is in the origin. This allows to use a *Euclidean distance* to the origin as a measure of textural similarity.

The *canonical Euclidean metric* has the drawback of prefering features with larger ranges of values, whereas features with only subtle differences have only minor influence. Therefore, we pass over to a more general metric: The canonical Euclidean distance is a special form of a *general Euclidean metric*

$$d = \sqrt{\mathbf{f}_{\mathrm{i}} \cdot \mathbf{A} \cdot \mathbf{f}_{\mathrm{i}}^{T}}. \qquad (1)$$

Choosing $\mathbf{A} = \mathbf{E}$, $\mathbf{E}$ the unity matrix, we obtain the canonical Euclidean distance of vector $\mathbf{f}_{\mathrm{i}}$ from the origin. A metric taking different variances of the components into account is the *Mahalanobis metric*. In this case of Equ. 1 we choose $\mathbf{A} = \mathbf{\Sigma}^{-1}$, $\mathbf{\Sigma}$ the covariance matrix of $\mathbf{F}_{\mathrm{ref}}$,

$$d = \sqrt{\mathbf{f}_{\mathrm{i}} \cdot \mathbf{\Sigma}^{-1} \cdot \mathbf{f}_{\mathrm{i}}^{T}}, \qquad \Sigma = \mathrm{cov}(\mathbf{F}_{\mathrm{ref}}). \qquad (2)$$

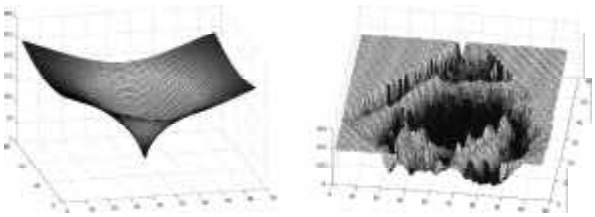
By computing the Mahalanobis distance of every $\mathbf{f}_{\mathrm{i}} \in \mathbf{F}$ we obtain an image I_{T} which captures the textural similarity with respect to the texture within the seed region R.

2.1.3. Combination of spatial and textural components. The *active image* I_{A} is the result of the combination of the spatial component I_{S} and the textural component I_{T}. We suggest a combination of the form

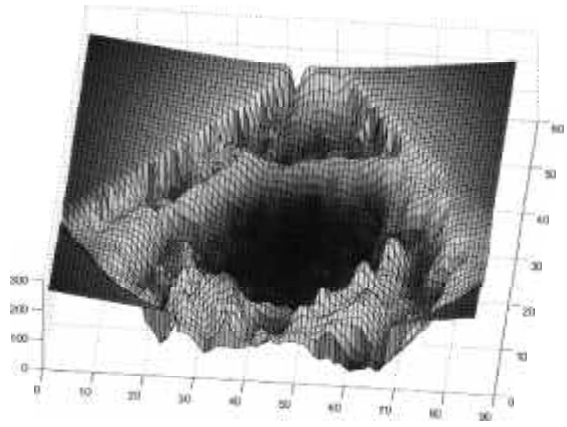
$$I_{\mathrm{A}} = (I_{\mathrm{T}} + \lambda) I_{\mathrm{S}}. \qquad (3)$$

The parameter λ is of great importance because it determines the resulting active image to a high degree. The higher λ, the more the spatial component I_{S} determines the active image I_{A}. Thus, it is reasonable to set λ according to image quality. In images with distinct textural properties of the target area we set λ to small values in order to increase the influence of I_{T}. If texture evaluation does not yield a distinct discrimination of the target area from the surrounding tissue, we set λ to larger values to obtain a higher influence of I_{S}. In these situations, I_{S} is absolutely necessary since otherwise a final segmentation technique such as active contours will hardly yield any useful result. A suggestion for an automatic determination of λ can be found in [3].

Now we are able to combine the spatial and the texural distance function to obtain the active image. See Fig.3 for an example of the components and their combination.



(a) Spatial distance (b) Textural distance



(c) Resulting active image

Figure 3. Components of the active image

Note, that we did *not* incorporate the original image data. However, an appropriately filtered or modified original image as third component might make sense.

3. Active Polygon Algorithm

The key idea of our 3D segmentation approach is that we rotate an image plane about a rotation axis. This leads to several slices of the volume data each of them providing a different 2D section of the object. Within these sections we apply a 2D active polygon segmentation.

In *two orthogonal slices* we perform a semiautomatic segmentation based on the input of starting contours. These contours are used twofold: for setting up the *active image* and as starting contours of the active polygon algorithm.

In the first step we prepare the contours by scaling them down to 85% of their original size. This ensures that all vertices are located inside the target area which is necessary for the computation of the *active image*. The vertices will be used for the spatial component and define the textural reference area R for evaluating the textural distance function.

3.1. Concept

The method of active polygons is based on vertices moving according to

- gradient information drawn from the *active image* at points defined by the vertices (*stopping term*),
- current shape of the polygon that influences the "moving direction" of each polygon vertex (*external force*),
- discrete version of affine curvature flow [7], which has a smoothing effect on the shape of the polygon (*internal force*).

The propagation of the polygon is given by

$$V_{move}(v) = \beta(v) * [F_{int}(v) + \alpha F_{ext}(v)] \qquad (4)$$

where V_{move} is the movement vector of each vertex, β is the stopping term, F_{int} is the internal force, F_{ext} is the external force, and α is a parameter that affects the propagation speed.

3.2. Stopping Term

The stopping term is similar to Perona-Malik-type diffusivity-terms ([8]),

$$\beta(v) = \frac{1}{1 + \frac{I_A(v) \cdot |\nabla I_A(v)|}{\lambda^2}} \qquad (5)$$

where $I_A(v)$ is the value of the *active image* at the position of vertex v, $\nabla I_A(V)$ is the gradient of I_A at the position of vertex v, and λ defines the sensitivity of the stopping term.

The stopping term is the only part that uses image information by means of the active image steering the polygon propagation.

3.3. External Force

The external force is responsible for propagation of the active polygon. It is not based on any image information. Instead the direction of the external force is determined by the *bisector* of the two adjacent edges (see Fig.4). The moving speed is a constant value α (see Equ.4) that influences the number of iterations required.

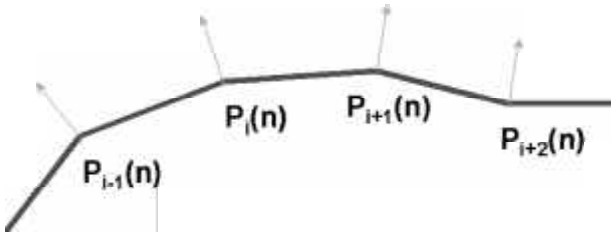


Figure 4. Direction of external force at each vertex

3.4. Internal Force

The aim of the internal force is to reduce the curvature of the active polygon in order to achieve a smooth contour. The direction of the internal force vector of each vertex is towards the centre of mass of the triangle which is set up by two adjacent edges (see Fig.5). The norm of the internal force vector is given by

$$||F_{int}(v)|| = 1 - e^{-\frac{s^2}{b^2} + \frac{1-\cos\varphi}{2\sin\varphi}} \qquad (6)$$

where b denotes the base of the triangle which is set up by the two adjacent edges of the polygon of vertex v, s denotes the length of the vector towards the center of mass, and φ is given by $\varphi = 2\pi/N$ with N the number of vertices of the current polygon.

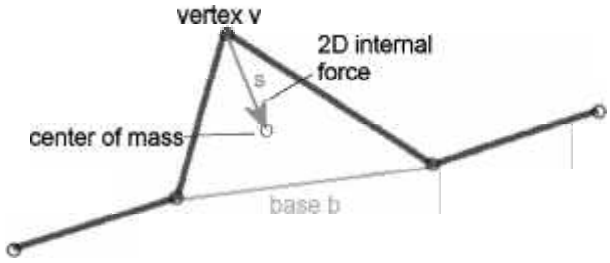


Figure 5. The internal force vector in a 2D context

The final result of the active polygon algorithm is a 2D segmentation of two orthogonal slices of the volume dataset.

4. Tiling

In the previous section we explained a method to obtain a semiautomatic segmentation of two slices of a volume data set (see Fig. 6). The results are two polygons S_{ref1} and S_{ref2}. We now present a method to obtain a rough estimation of contours in the remaining slices. This is done by rotating S_{ref1} and S_{ref2} about the rotation axis into a

common plane. The following tiling algorithm yields an interpolated polygon for each in-between slice and finally a very simplified 3D model of the object.

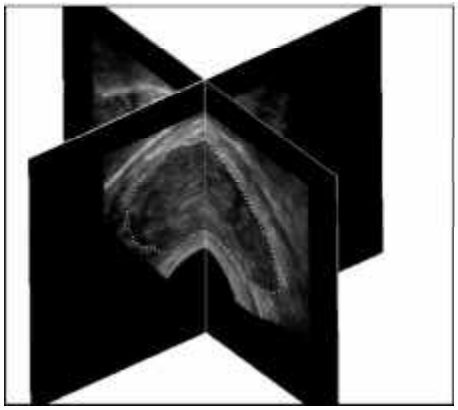


Figure 6. User input - 2D segmentation in two perpendicular slices

For the polygon interpolation some ideas of tiling ([9]) are adopted to our needs. First of all a correspondence between the vertices of the given polygons has to be defined. Subsequently we interpolate between the locations of corresponding vertices.

4.1. Correspondence Between Polygon Vertices

A correspondence between the vertices of two polygons P and Q (where the number of vertices of P is m and number of vertices of Q is n and $m \geq n$) is defined as follows:

Let P and Q both be parameterised over the parameter t, with $0 \leq t \leq 1$. Let p_i be the i-th vertex of P and q_j the j-th vertex of Q with $1 \leq i \leq m$ and $1 \leq j \leq n$ and let t_i be the parameter value of p_i. We define a correspondence $c(p_i)$ as:

$$\begin{aligned} c(p_i) = q_j \text{ if} \\ ||q(t_i) - q_j)|| < ||(q(t_i) - q_k)|| \\ \forall\, 1 \leq k \leq n,\ k \neq j. \end{aligned} \tag{7}$$

Many other methods for computing correspondences have been proposed in the context of tiling algorithms; the suggested one being among the computationally most efficient.

The interpolation between corresponding vertices can be done as follows: Each correspondence line is linearly parameterised over a parameter s, with $0 \leq s \leq 1$. With $s = 0$ we get the source polygon P, with $s = 1$ we obtain the target polygon Q. Therefore, for any parameter value $0 < s < 1$, we get an "in-between polygon" (see Fig. 7).

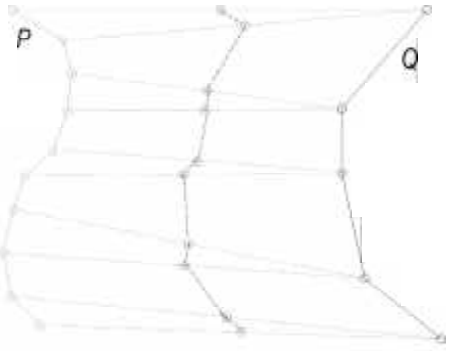


Figure 7. Two contours P and Q with correspondence lines and the vertices of an interpolated polygon with parameter $s = 0.5$

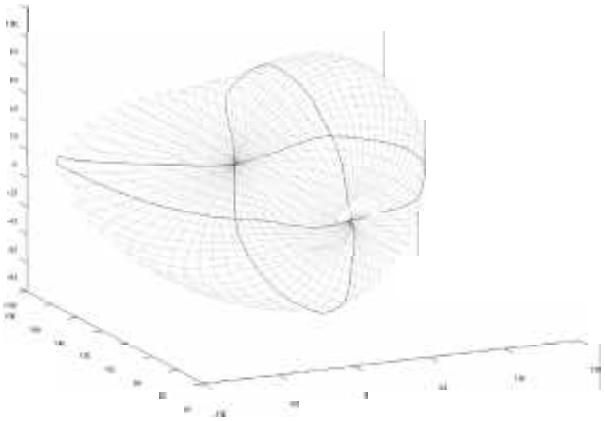


Figure 8. User input (solid lines) and tiling result (dotted lines) of all slices in 3D

5. 3D Segmentation by 2D Active Polygons

Having computed initializations for both, the *active image* and the active contour by a tiling scheme, we may obtain a 3D segmentation of the volume data set by using the 2D active polygon from section 3 in every slice.

Since the external force and the stopping term use both information based only on 2D information we have to incorporate 3D information into the *internal force model* to induce coherent propagation of neighbouring slices.

Instead of the 2D vector to the center of mass of a triangle we use the *umbrella vector* (see [10]), the vector from the considered point T to the barycenter of a set of neighboring points B_i:

$$\mathbf{U} = \frac{1}{k} \sum_{i=1}^{k} (\mathbf{B_i} - \mathbf{T}). \tag{8}$$

Remark:

The norm of the umbrella vector is a measure of the mean curvature under certain conditions: At each point T of a surface Φ, there exists an *approximating paraboloid* Γ, which agrees at T with Φ up to second order. This paraboloid can be written in a locally adapted coordinate system with origin at T as

$$z = \frac{\kappa_1}{2}x^2 + \frac{\kappa_2}{2}y^2. \tag{9}$$

Here, κ_1, κ_2 denote the principal curvatures of Φ and Γ at T. We now work directly with Γ and arrange points B_i on Γ around T in a rather special way. The projections B_i' of the points B_i onto the tangent plane $z = 0$ at T shall form a regular hexagon with center T. Hence, these points B_i' have as first two coordinates $(\pm r\cos\varphi, \pm r\sin\varphi, z)^T$, $(\pm r\cos(\varphi + \pi/3), \pm r\sin(\varphi + \pi/3), z)^T$, and $(\pm r\cos(\varphi - \pi/3), \pm r\sin(\varphi - \pi/3), z)^T$. We compute the z-coordinates of the points B_i with (9) and find for the umbrella vector according to (8),

$$\|\mathbf{U}\| = \frac{1}{2}r^2\left(\frac{\kappa_1 + \kappa_2}{2}\right). \tag{10}$$

Thus, the norm of the umbrella vector does not depend on the rotational angle ϕ, but just on the mean curvature $(\kappa_1 + \kappa_2)/2$ and on the square of the distance r.

End of remark

In our case, the B_i consist of three vertices with shortest distance to T of each neighbouring plane N_1 and N_2. The B_i do not necessarily lie in a plane. Therefore the norm of the umbrella vector gives only a rough estimation of the mean curvature at T. Generally, the umbrella vector of vertex T does not lie in the slice plane of T. In our case, we use the shortest rotation of the umbrella vector into the slice plane (see Fig.9) to finally obtain the internal force vector F_{int} for propagation of all active contours.

As discussed above, the umbrella vector is a measure of the mean curvature only in the case that the base of the pyramid is an approximately regular hexagon. Therefore an approximately regular triangulation is necessary.

5.1. Stopping Criterion

When the active contour algorithm is in progress we have to make sure to stop propagation of the polygons at the optimal position i.e. the result polygon meeting the boundary of the target area in all slices. In our approach we compare the area of the polygons S_{ref1}, S_{ref2} obtained by the initial segmentation according to section 3 with the area of the propagated polygons S_1 and S_2 in these sections during iteration. If the area of the polygon of the relevant slices is

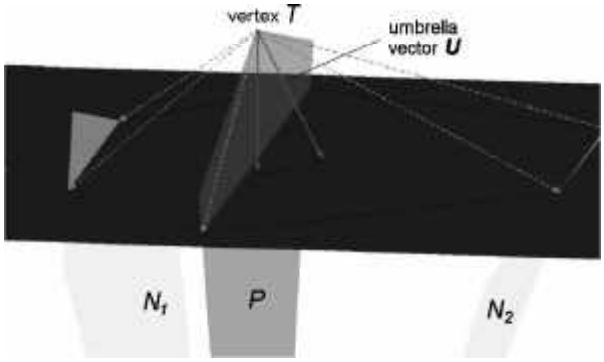


Figure 9. Umbrella vector at a vertex of triangulated points. Plane P is the (2D)-polygon plane, the neighbouring planes are N_1 and N_2, respectively

smaller than the area of S_{ref1} or S_{ref2} respectively, polygons of all slices are propagated. If one of the polygons S_1 or S_2 reaches the area size of S_{ref1} or S_{ref2} , propagation is stopped by setting the external force to 0. The second user input polygon keeps its original external force. We propagate the polygons of the remaining slices by an external force of interpolated magnitude. In Fig.10 the polygons S_1 and S_2 are denoted by the solid lines and the remaining polygons by dotted lines. The propagation finally stops as soon as both, S_1 and S_2, reach their original area size.

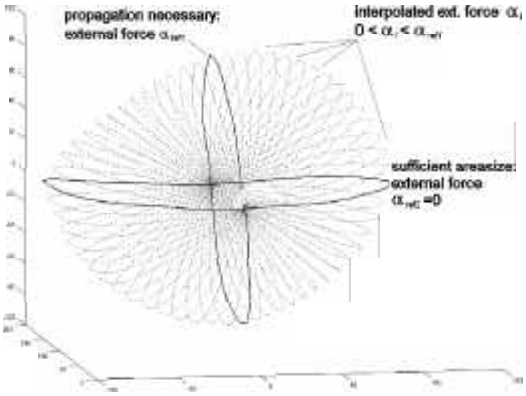


Figure 10. By dynamically changing the magnitude of the external force propagation stops

6. Results

We demonstrate the results one can obtain by the suggested framework by segmentation of a 3D data set of a prostate gland. The prostate is an organ of roughly rotationally symmetric shape, hence our approach is applica-

ble. Segmentation starts in two orthogonal slices, depicted in Fig. 11, leftmost column. The final 3D segmentation result by concurrent fully automatic segmentation in all slices is depicted in Fig. 12 and Fig. 15. The contour propagation in intermediate slices stops at the border of the target region, see Fig. 11, middle and right column, for some examples.

Another example of approximately rotationally symmetric objects are cancers. In Fig. 13 and Fig. 14 the segmentation and model of an ovary cancer can be seen.

It should be mentioned that all figures show the results of our approach without any postprocessing step. Due to the fact that in some cases the 3D models contain some "roughness" at the surface caused by the slice based evolution of the active contours, it will be very useful to apply some kind of smoothing to avoid these artefacts.

7. Conclusions

We presented an efficient method for segmentation of approximately rotationally symmetric objects from 3D data sets. Our approach uses highly efficient methods of computational geometry such as an *active image*, *polygon tiling*, *interpolating subdivision*, mean curvature estimation by an *umbrella vector* approach, and a 2D *active polygon* for the segmentation.

It is important to note that an extension of the basic idea of our scheme to objects with different geometric properties is possible in a straightforward way. Similar applications are surfaces where the intersections with planes of a pencil are simply connected. A sufficient but not mandatory condition is a star shaped surface. Another interesting application is the use of parallel slices, see 16. Thus, in computer tomography the application of the proposed method is possible with only minor modifications.

Another important point is the runtime performance of the algorithms. Computation on a P4-2260 takes about 15 seconds for a complete segmentation in all slices and the generation of the 3D-model. By further optimization it can be expected that the computation time will be beneath 10 seconds in the near future. This makes our algorithms of 3D model generation a highly preferable method in practical applications.

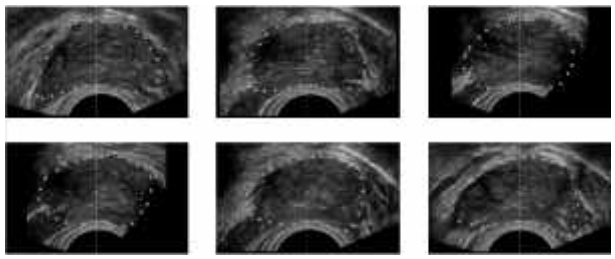


Figure 11. Six slices of a volume dataset that show the 2D results of the 3D segmentation algorithm

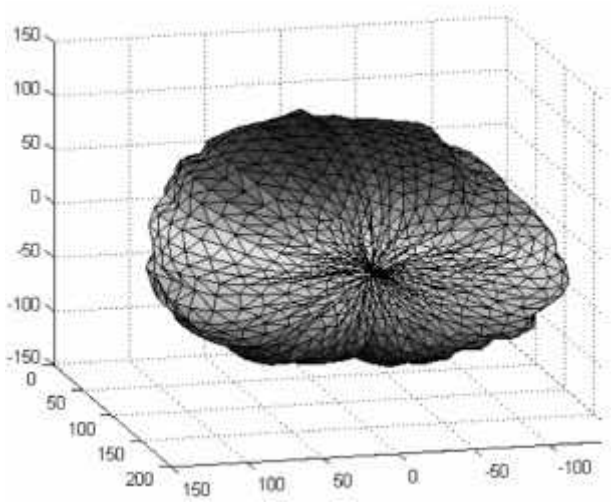


Figure 12. Triangulation of the result

References

[1] Ning Hu, "Prostate surface segmentation from 3d ultrasound images," in *Biomedical Imaging, 2002. Proceedings. 2002 IEEE International Symposium on*, 2002, pp. 613–616.

[2] A. Ghanei, H. Soltanian Zadeh, A. Ratkewicz, and Fang Y. Fang, "A three-dimensional deformable model for segmentation of human prostate from ultrasound images," *Medical Physics.*, vol. 28, no. 10, pp. 2147–2153, 2001.

[3] D. Hönigmann, J. Ruisz, and H. Pottmann, "Fast model-based segmentation of ultrasound data using an active image," in *IEEE Symposium on biomedical imaging*, 2002, pp. 225–228.

[4] A. Sarti, R. Malladi, and J.A. Sethian, "Subjective surfaces: A geometric model for boundary completion," *International Journal of Computer Vision*, 2002.

[5] N. Dyn, D. Levin, and J. Gregory, "4-point interpolatory subdivision scheme for curve design," *Computer Aided Geometric Design*, 1987.

[6] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics.*, vol. SMC-3, no. 6, 1973.

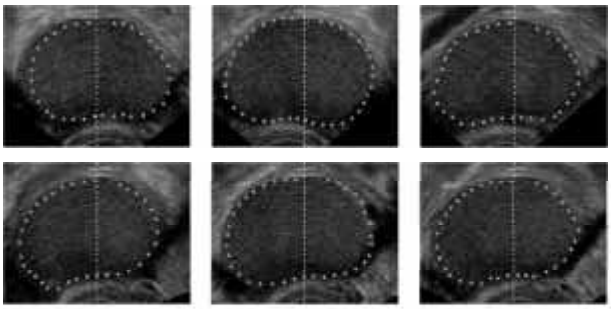


Figure 13. Another six result slices of a volume dataset that show an ovary cancer

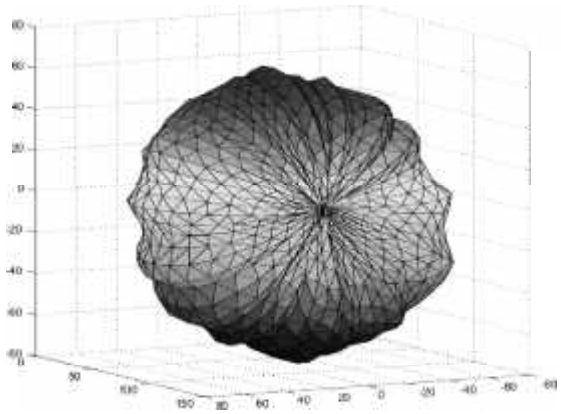


Figure 14. Triangulation of the model of the ovary cancer

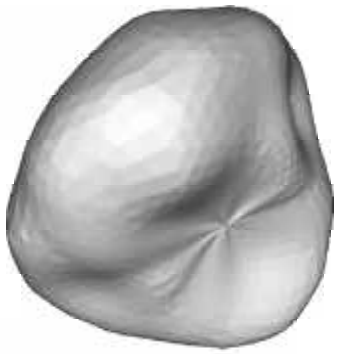


Figure 15. Rendered surface of the 3D model of Fig.14

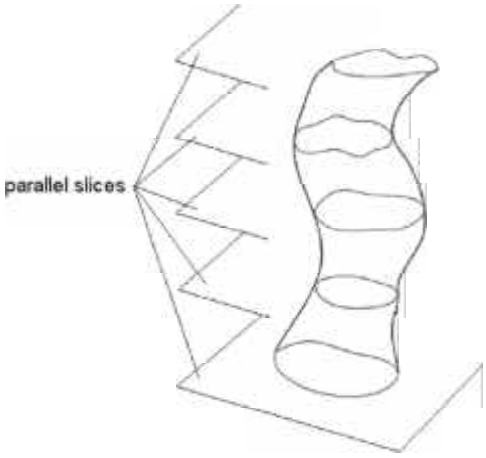


Figure 16. Model with parallel slices

[7] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, 2001.

[8] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–39, 1990.

[9] G. Barequet and M. Sharir, "Piecewise-linear interpolation between polygonal slices," Proc. 10th Annu. ACM Sympos. Comput. Geom., 1994, pp. 93–102.

[10] L. Kobbelt, *Iterative Erzeugung glatter Interpolanten*, Ph.D. thesis, University Karlsruhe, 1995.