Tamed Snake: A Particle System for Robust Semi-automatic Segmentation

Johannes Hug, Christian Brechbühler, and Gábor Székely

Swiss Federal Institute of Technology
ETH Zentrum, CH-8092 Zürich, Switzerland
{jhug,brech,szekely}@vision.ee.ethz.ch

Abstract. Semi-automatic segmentation approaches tend to overlook the problems caused by missing or incomplete image information. In such situations, powerful control mechanisms and intuitive modelling metaphors should be provided in order to make the methods practically applicable. Taking this problem into account, the usage of subdivision curves in combination with the simulation of edge attracted mass points is proposed as a novel way towards a more robust interactive segmentation methodology. Subdivision curves provide a hierarchical and smooth representation of a shape which can be modified on coarse and on fine scales as well. Furthermore, local adaptive subdivision gives the required flexibility when dealing with a discrete curve representation. In order to incorporate image information, the control vertices of a curve are considered mass points, attracted by edges in the local neighbourhood of the image. This so-called Tamed Snake framework is illustrated by means of the segmentation of two medical data sets and the results are compared with those achieved by traditional Snakes.

1 Introduction

Fully automatic and manual segmentation methods represent two extremes of a wide spectrum of segmentation algorithms. Between these, a whole family of interactive semi-automatic tools [2,5,10,11] has been developed, promising to combine the advantages of computational support, resulting in precise border detection, with the benefits from manual manipulation possibilities. The application of these algorithms to the segmentation of medical imagery supports the human operator considerably in the case of clear and sharp organ outlines. But whenever edge information at organ boundaries is disturbed or missing, as is often the case in rather demanding tasks, such as the model-making of an abdominal cavity [12], manual outlining based on interpolating B-splines proved superior in every respect to the semi-automatic approaches. In particular, the controllability of geometric shapes is very poor in interactive operation mode. The generic gap-closing mechanism provided by these algorithms between clear border segments seems to be unsuited and not sufficiently under control of the operator.

Most existing segmentation tools lack an intuitive manipulation semantic, especially on coarse scales, and sometimes on finer scales as well. Hence the

C. Taylor, A. Colchester (Eds.): MICCAI'99, LNCS 1679, pp. 106–116, 1999.

geometric representation must be chosen in order to alleviate the interaction difficulties arising when the human operator is required to intervene during the segmentation process. Fortunately, the computer graphics community supplies us with a large number of animation and modelling techniques well suited to overcome the aforementioned inadequacies. In particular, hierarchical multiresolution representations are promising candidates regarding the necessary interactive modelling capabilities and the suitability for numerical simulations.

The work presented in this paper is a first step towards combining traditional computer vision approaches with intuitive modelling semantics in order to provide a truly operational interactive segmentation system. At first, we will review selected representatives of the class of interactive segmentation algorithms which have been developed during the last twenty years. Having taken a quick glance at recent advances in geometric modelling, Sect. 2 presents a novel combination of edge attracted mass points with subdivision curves, adopting both automatic boundary detection and interactive modelling. Finally, Sect. 3 compares this new framework with traditional Snakes on the basis of two medical data sets.

1.1 Earlier Interactive Segmentation Approaches

The first class of algorithms we would like to mention considers the image I as a discrete neighbourhood graph, where each pixel corresponds to a node in the graph. A cost function $P(I)$ assigns a value to each node in the graph. For edge tracking purposes, the cost function is usually based on the magnitude of the gradient of the low-pass filtered image. A very popular cost function for minimisation algorithms is therefore

$$P(I) = 1 - \frac{||\nabla(\text{Gauss} \otimes I)||}{\max(||\nabla(\text{Gauss} \otimes I)||)} \quad . \tag{1}$$

With such a cost function, edge tracking can be considered as a minimal cost path search problem between two points in the image graph. Fischler, Tenenbaum and Wolf [2] used the well known $F^{\star}$ algorithm to solve this minimisation problem in a semi-automatic framework for road detection; Barrett and Mortensen [10] applied the very similar $A^{\star}$ algorithm for the interactive segmentation of medical imagery. Interaction possibilities are limited to the choice of start and end points of the path. Subsequent modifications of the shape can only be accomplished by moving each single polygonal vertex.

In order to link polygonal vertices together, whilst abandoning the discrete working domain, Kass, Witkin and Terzopoulos [5] introduced Snakes: A Snake is the simulation of a physical system consisting of an elastically deformable mass-spring object living in a potential field, $P(I)$, as defined above. Unfortunately, the shape regularisation provided by the elasticity and rigidity terms is only available during automatic evolution, wherefore manual editing still remains very cumbersome. Several extensions, such as Ziplock Snakes [11], have been developed, but the basic problem with the underlying geometry was not addressed until the invention of B-spline Snakes [6,7] using dynamic knot insertion. Despite manual modifications on coarse scales being possible with B-Snakes, fine scale editing results in a proliferation of control points, destroying the coarse scale manipulation semantic.

1.2 Hierarchical Modelling

Hierarchical modelling is composed of two very general key components: Iterative refinement of the geometry which defines a hierarchy of representations with increasing degrees of freedom; and local detail encoding which represents the details of a finer level with respect to the next coarser one. These two building blocks enable a natural editing semantic on coarse and on fine scales, as required for a practically useful segmentation tool. This concept has been introduced by Forsey and Bartels [3] for B-spline curves and surfaces, and has led to several experimental and industrial modelling systems [13,4].

Subdivision curves and surfaces are well suited for hierarchical modelling since their representation implicitly comprises a hierarchy of refined shapes. Furthermore, local detail encoding is only a simple extension in the context of subdivision schemes. The favourable properties of this geometric description have already been used among others for finite element based shape recovery [9], and for feature based indirect volume rendering [8].

2 Tamed Snakes

In order to take advantage of the benefits from both the Snake-like edge tracking and hierarchical modelling, we present a new combination of the two underlying components. In the following paragraphs, we will give a detailed description of the components establishing the core of the Tamed Snake framework.

2.1 Subdivision Curves

Univariate subdivision schemes are usually defined as the iterative application of an operator which maps a given polygon $\mathcal{P}_k = [\mathbf{p}_i^{(k)}]$ to a refined polygon $\mathcal{P}_{k+1} = [\mathbf{p}_i^{(k+1)}]$. Such an operator is given by two rules for computing the new so-called even vertices $\mathcal{P}_{k+1}^{\natural} = \{\mathbf{p}_{2i}^{(k+1)}\}$ and the new odd vertices $\mathcal{P}_{k+1}^{\sharp} = \{\mathbf{p}_{2i+1}^{(k+1)}\}$. For interpolatory refinement, the rule for even points is always $\mathbf{p}_{2i}^{(k+1)} = \mathbf{p}_i^{(k)}$.

In our Tamed Snake implementation, the first smooth interpolating subdivision scheme from Dyn, Levin and Gregory [1] is employed. The subdivision rules, which are also known as DLG-subdivision scheme, are given by

$$\mathbf{p}_{2i}^{(k+1)} = \mathbf{p}_i^{(k)}, \qquad \mathbf{p}_{2i+1}^{(k+1)} = (\frac{1}{2} + \omega)(\mathbf{p}_i^{(k)} + \mathbf{p}_{i+1}^{(k)}) - \omega(\mathbf{p}_{i-1}^{(k)} + \mathbf{p}_{i+2}^{(k)}) \ . \tag{2}$$

If the magnitude of the free tension parameter ω is chosen inside the interval $(0, \frac{1}{4})$, the limit curve will be continuous, and for $0 < \omega < \frac{1}{8}$, the limit curve has a continuous tangent vector given by the following equation:

$$\mathbf{p}'_i = \frac{1}{1-4\omega}\left[\frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) - \omega(\mathbf{p}_{i+2} - \mathbf{p}_{i-2})\right] \tag{3}$$

An example depicting three refinement steps carried out with this 4-point subdivision stencil is displayed in Fig. 1.

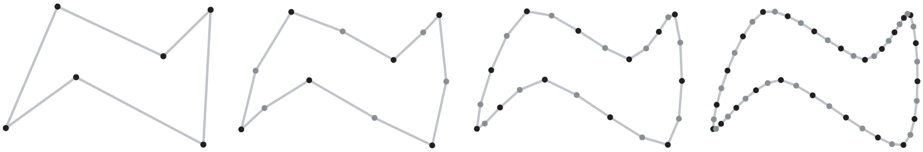

Fig. 1. Interpolatory refinement: The iterative application of the DLG-subdivision operator transforms an initial control polygon $\mathcal{P}_0$ (left) into a finer polygon $\mathcal{P}_3$ (right). Black vertices represent even points, and grey vertices denote odd points.

Local detail encoding is achieved by establishing a local coordinate system $f_i^{(k)}$ in each vertex $\mathbf{p}_i^{(k)}$ and by representing details with respect to this local frame (see Fig. 2). A very natural choice for $f_i^{(k)}$ is the normalised tangent $\frac{\mathbf{p}'_i}{\|\mathbf{p}'_i\|}$ together with the normal vector counterclockwise perpendicular to the tangent in each vertex.

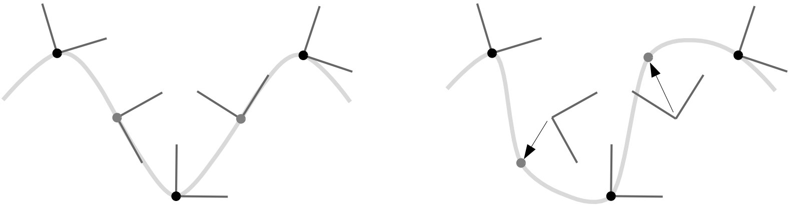

Fig. 2. Local coordinate frames for detail encoding. The two odd points of the left curve are modified by adding local detail coefficients (right).

2.2 Local Adaptive Subdivision

In order to have efficient data structures and fast algorithms at hand, we can exploit the hierarchical, tree-like structure of a subdivision curve as well as the local support of the refinement operator. As illustrated in Fig. 3, a flexible representation of the hierarchy of vertices is a combination of a doubly linked list with a collection of binary trees. In the following, we will refer to such a structure as a *listed tree*. Each point $\mathbf{p}_i^{(k)}$ is represented by a node on level k in a tree. The links between two nodes can be interpreted as the polygonal segments in between two points $\mathbf{p}_i^{(k)}$ and $\mathbf{p}_{i+1}^{(k)}$. Please notice that the nodes are also connected across tree boundaries, admitting of fast navigation in different resolution levels. In order to save memory, it is advantageous to separate the real vertex data from the tree nodes and to keep it in a dynamic array, since that way even points must only be stored once.

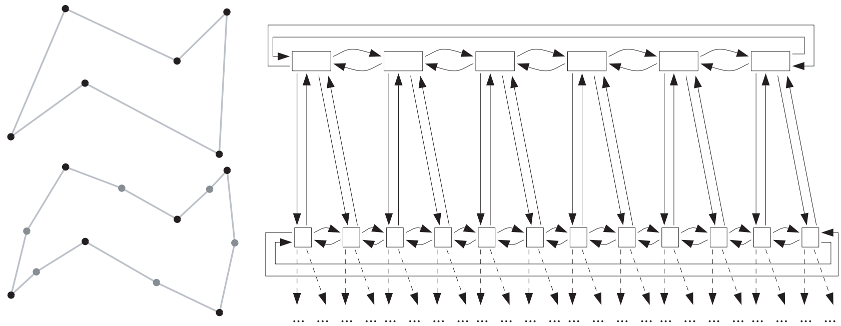

Fig. 3. A *listed tree* as a data structure for efficient and adaptive subdivision. The corresponding polygons $\mathcal{P}_0$ and $\mathcal{P}_1$ are displayed on the left of the nodes representing the points $\mathbf{p}_i^{(0)}$ and $\mathbf{p}_i^{(1)}$.

The main reason for choosing such a sophisticated data structure is the fact that flat regions of a curve need only a small number of subdivision steps, while segments with high curvature or with fine details require more degrees of freedom. Therefore the refinement operator is only applied to those parts of the curve that have additional details or where the local flatness is still too small. Figure 4 shows the effect of such an adaptive synthesis of a curve using a discrete curvature estimator as a flatness oracle.

Applying adaptive synthesis interactively produces very dynamic data structures. The resulting listed trees may become pretty unbalanced when flat regions and highly detailed segments occur in the same curve. Nevertheless, adaptive synthesis is well suited for interactive segmentation since it always assures an appropriate sampling of the shape.

As a consequence of the local support of the DLG-subdivision operator, the modification of a point $\mathbf{p}_i^{(k)}$ does not enforce the recalculation of the whole curve, but only a small part of it, depending on $\mathbf{p}_i^{(k)}$. Figure 5 shows these dependencies between two successive subdivision levels. The refinement of level k triggered by the modification of the point $\mathbf{p}_i^{(k)}$ can be sketched as follows: Firstly, the invalidated positions of the dependent vertices $\mathbf{p}_{2i-3}^{k+1}$, $\mathbf{p}_{2i-1}^{k+1}$, $\mathbf{p}_{2i+1}^{k+1}$, and $\mathbf{p}_{2i+3}^{k+1}$ must be recalculated using (2). Secondly, all affected local frames in level $k+1$

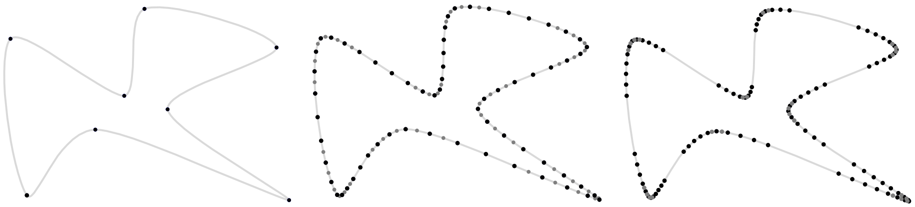

Fig. 4. Adaptively subdivided curve with vertices of level 0 (left), level 4 (middle), and level 5 (right). A discrete curvature estimator served as a flatness criterion. Regions with high curvature are subdivided, while flat regions remain unrefined.

are updated using (3), and lastly, detail coefficients are added. For the refinement of the next level $k+1$, only those of the altered vertices — those not passing the flatness test — are regarded as modified. This procedure is applied iteratively until no further segments need refinement.

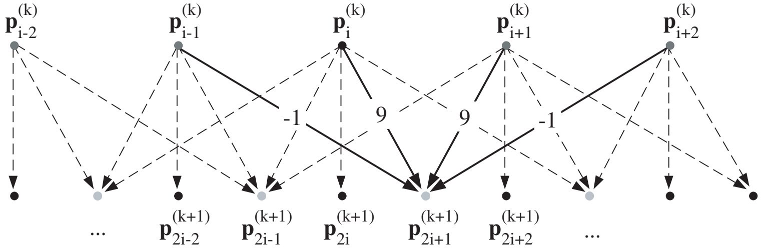
$\mathbf{p}_{i-2}^{(k)}$
$\mathbf{p}_{i-1}^{(k)}$
$\mathbf{p}_{i}^{(k)}$
$\mathbf{p}_{i+1}^{(k)}$
$\mathbf{p}_{i+2}^{(k)}$
-1
9
9
-1
...
$\mathbf{p}_{2i-2}^{(k+1)}$
$\mathbf{p}_{2i-1}^{(k+1)}$
$\mathbf{p}_{2i}^{(k+1)}$
$\mathbf{p}_{2i+1}^{(k+1)}$
$\mathbf{p}_{2i+2}^{(k+1)}$
...

Fig. 5. Local support of the points in level k; weights according to $\omega = \frac{1}{16}$ for the recalculation of the point $\mathbf{p}_{2i+1}^{(k+1)}$.

2.3 Particle Systems for Interactive Segmentation

The hierarchical structure of subdivision curves suggests starting the segmentation process with a reasonably coarse model and then to iteratively adjust and refine the control vertices of the resulting curve. Since we apply an interpolating subdivision scheme, all odd points of the current working level must be moved to a correct boundary position, before proceeding to the next finer resolution. If this holds for all vertices on coarser levels, the prediction of the refinement operator improves continuously with respect to the vertex positions on the next finer level and converges to the correct boundary position.

In order to combine such a coarse-to-fine strategy with Snake-like edge tracking, we must modify the traditional Snake energy to be optimised (see [5,11]). Since the regularisation term provided by the elastic rod model is useless on a coarse scale, we drop it in favour of a spring energy originating from springs attached to each odd control point (see Fig. 6). A spring allows for a snapping of the attached control vertex $\mathbf{p}_i$ into the correct boundary position, within the vicinity of its starting position $\mathbf{p}_i(0) = \mathbf{p}_i \mid_{t=0}$. The imposed restriction on the search space to the local neighbourhood is reasonable under the assumption of working with a good initialisation. Additionally, since the error of the refinement operator's prediction tends to decrease with each subdivision step, we can increase the spring constant $\delta^{(k)}$ in order to restrict the search area accordingly. Hence, the energy originating from all springs on refinement level k is given by:

$$E_{\mathrm{S}} = -\frac{1}{2} \sum_{\mathbf{p}_i \in \mathcal{P}_k^\sharp} \delta_i^{(k)} \, \|\mathbf{p}_i(t) - \mathbf{p}_i(0)\|^2 \tag{4}$$

Besides the spring energy, Tamed Snakes incorporate a kinetic energy E_{kin} depending on the mass μ, and an image energy E_{I} originating from the potential function defined in (1). The full energy term $E_{\mathrm{TS}} = E_{\mathrm{kin}} + E_{\mathrm{pot}}$, is therefore as follows:

$$\begin{gathered} E_{\mathrm{I}} = -\sum_{\mathbf{p}_i \in \mathcal{P}_k^\sharp} P(I) \mid_{\mathbf{p}_i(t)}, \quad E_{\mathrm{kin}} = \frac{1}{2} \sum_{\mathbf{p}_i \in \mathcal{P}_k^\sharp} \mu_i \left| \frac{\partial \mathbf{p}_i(t)}{\partial t} \right|^2, \quad E_{\mathrm{pot}} = E_{\mathrm{S}} + E_{\mathrm{I}} \\ E_{\mathrm{TS}} = \frac{1}{2} \sum_{\mathbf{p}_i \in \mathcal{P}_k^\sharp} \left[\mu_i \left| \frac{\partial \mathbf{p}_i(t)}{\partial t} \right|^2 - \delta_i^{(k)} \, \|\mathbf{p}_i(t) - \mathbf{p}_i(0)\|^2 - 2P(I) \mid_{\mathbf{p}_i(t)} \right] \end{gathered} \tag{5}$$

Since a Tamed Snake defined by (5) can be regarded as an energy conserving physical system, we introduce as usual the Rayleigh dissipation functional depending on the viscosity γ in order to stabilise the numerical simulation. Using variational calculus for the derivation of the Euler-Lagrange equations of motion, we end up with (6) describing the motion of all mass points $\mathbf{p}_i$ at the time t:

$$\forall \; \mathbf{p}_i \in \mathcal{P}_k^\sharp : \; \mu_i \frac{\partial^2 \mathbf{p}_i(t)}{\partial t^2} + \gamma_i \frac{\partial \mathbf{p}_i(t)}{\partial t} + \delta_i^{(k)} (\mathbf{p}_i(t) - \mathbf{p}_i(0)) = -\nabla_\perp P(I) \mid_{\mathbf{p}_i(t)} \tag{6}$$

In order to prevent the control vertices from drifting along the boundary, the gradient of the potential is projected in normal direction of the curve. This operation has been denoted by $\nabla_\perp$ in (6).

Since we dropped the regularisation term in the energy functional, there are no longer derivatives with respect to a parameterisation along the curve. The odd points of the curve are therefore not correlated during energy minimisation and, as a direct consequence, the physical system degrades to an uncorrelated particle system. For this reason, the governing differential equation (6) is only an ordinary second order differential equation which can be solved for each vertex $\mathbf{p}_i$ separately. Using backward differences and resolving for $\mathbf{p}_i^{[t]}$, a simple discretisation results for numerical simulation, as illustrated in (7), where the

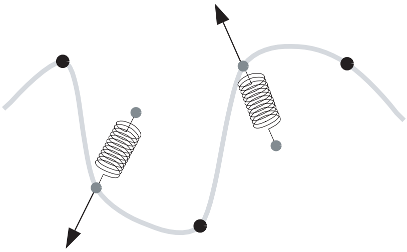

Fig. 6. Springs are attached to all odd vertices $\mathbf{p}_i \in \mathcal{P}_k^\sharp$.

symbol p_i represents either the x- or y-component of $\mathbf{p}_i$:

$$p_i^{[t]} = \frac{1}{(\mu_i + \gamma_i + \delta_i^{(k)})} \left[\mu_i (2p_i^{[t-1]} - p_i^{[t-2]}) + \gamma_i p_i^{[t-1]} + \delta_i^{(k)} p_i^{[0]} - \nabla_\perp P(I) \mid_{\mathbf{p}_i^{[t-1]}} \right] \tag{7}$$

Note, the equations for x- and y-components are semi-linear since they are coupled by the image term $P(I)$ and by the spring term. Fortunately, the latter can be decoupled by replacing the spring with two orthogonal springs in x- and y-direction, respectively.

3 Segmentation Results

Without any prior knowledge of the structure to be segmented, a Tamed Snake must be initialised by a small number of points defining a very rough outline of the shape. The (upper) left images in Fig. 7 and Fig. 8 depict possible initialisations for the segmentation of a bladder and a corpus callosum, respectively. The more points are provided by the human operator, the closer the vertices of the subsequently refined curves are to their correct boundary position. Although the number and position of the initial points are not decisive for the quality of the resulting segmentation, the initialisation has a strong impact on the additional manual editing required for achieving good results. This can be illustrated by the second and third image in Fig. 8, where the initial position of the vertices in the Splenium and the Genu of the corpus callosum must be adjusted manually to bring them into the local neighbourhood of their correct position.

The segmentation itself is a quick interplay between automatic edge tracking and manual manipulation. The optimisation of the boundary position for all odd vertices in the current working level stops as soon as the simulated physical system has reached a stable state. Afterwards, the operator can intervene manually if necessary or proceed to the next finer representation. This iterative refinement of the segmentation is illustrated in Fig. 7 and Fig. 8. The first five images depict the shapes with increasing refinement after edge optimisation and occasional manual corrections where necessary.

Good results can be achieved applying an exponential weighting function to the spring constants $\delta^{(k)} := \delta^{(0)} e^k$. Although the working range for automatic edge tracking consequently becomes very restricted on finer levels, it proved to be necessary to use such a strong weighting function in order to conserve a regular sampling of the curve. Nevertheless, since the prediction of the subdivision scheme is sufficiently precise on fine scales, the exponential weighting is not a restricting limitation of the framework.

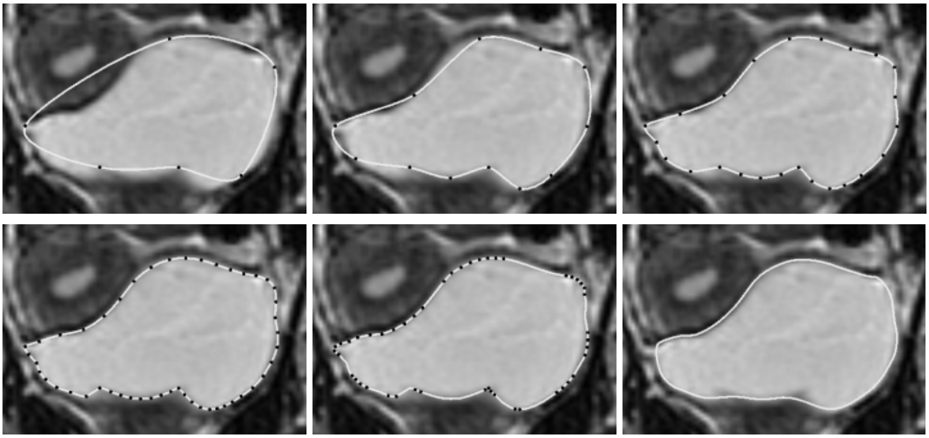

Fig. 7. Segmentation of a bladder with a Tamed Snake. The upper row and the two lower left images depict a Tamed Snake with increasing refinement after edge optimisation and occasional manual modifications. The lower right picture shows the best possible solution with a traditional Snake.

In order to compare the results with traditional approaches, the lower right image in Fig. 7 and the rightmost image in Fig. 8 show the segmentation results achieved with a traditional Snake. While the result for the bladder is, on a coarse scale, of equal quality as the Tamed Snake segmentation, the original Snake is shown to partially fail in the case of the corpus callosum. Although manual editing of the resulting polygonal shape is theoretically possible, it would be a tedious and time consuming task to adjust all the misdirected vertices. Equally time consuming is the manual modification of the bladder-Snake required to capture the intricate details obtained by the Tamed Snake at its finest scale.

4 Conclusion and Future Research

Tamed Snakes combine powerful editing semantics with traditional edge tracking methods so as to provide a robust segmentation tool. Although Tamed Snakes are, in the case of clear boundaries, not as fast and elegant as traditional Snakes, the advantages of better controllability and stronger editing semantics are, given

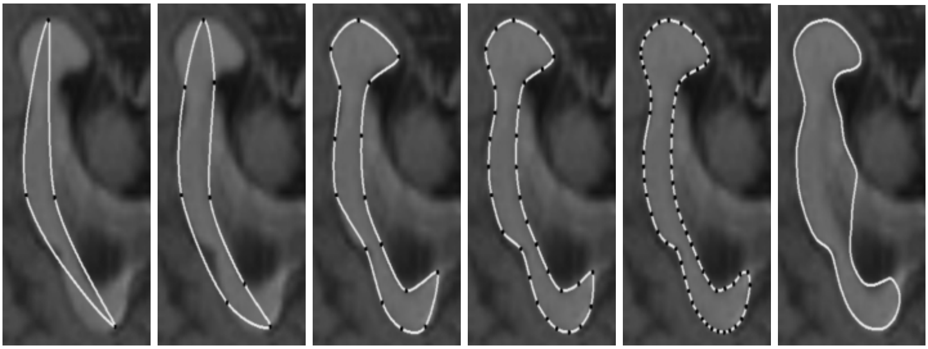

Fig. 8. Segmentation of a corpus callosum with a Tamed Snake. The rightmost picture shows the partial failure of a traditional Snake, in spite of a good manual curve initialisation.

all considerations, superior to all interactive segmentation algorithms we have tested up to the present day.

Further research has to be carried out in the field of model-based initialisations in order to reduce the necessary initial user interaction. Additionally, we plan to extend Tamed Snakes for the segmentation of three-dimensional medical datasets, where controllability of the shape is even more important than in the two-dimensional case.

References

1. Nira Dyn, David Levin, and John A. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4(4):257–268, 1987. 108
2. M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–233, 1981. 106, 107
3. David R. Forsey and Richard H. Bartels. Hierarchical B-spline refinement. In *Computer Graphics Proceedings 1988*, pages 205–212. ACM SIGGRAPH, 1988. 108
4. Steven J. Gortler and Michael F. Cohen. Hierarchical and variational geometric modeling with wavelets. In *1995 Symposium on interactive 3D Graphics*, pages 35–42, April 1995. 108
5. Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 106, 107, 111
6. F. Leitner, I. Marque, S. Lavallée, and P. Cinquin. Dynamic segmentation: Finding the edge with Snake splines. In *Curves and Surfaces*, pages 279–284, 1991. 107
7. Haihong Li. *Semi-automatic Road Extraction from Satellite and Aerial Images*. PhD thesis, Swiss Federal Institute of Technology Zürich, Switzerland, 1997. 107

8. C. Lürig, L. Kobbelt, and T. Ertl. Deformable surfaces for feature based indirect volume rendering. In *Proceedings of CGI 98*, pages 752–760, June 1998. 108
9. Chhandomay Mandal, Baba C. Vemuri, and Hong Qin. A new dynamic FEM-based subdivision surface model for shape recovery and tracking in medical images. In *Proceedings of MICCAI 98*, pages 753–760. Springer, October 1998. 108
10. E. N. Mortensen and W. A. Barret. Fast, accurate, and reproducible live-wire boundary extraction. *Proceedings of VBC 96*, pages 183–192, September 1996. 106, 107
11. Walter M. Neuenschwander. *Elastic Deformable Contour and Surface Models for 2-D and 3-D Image Segmentation*. PhD thesis, Swiss Federal Institute of Technology Zürich, Switzerland, 1996. 106, 107, 111
12. G. Székely, M. Bajka, J. Hug, M. Manestar, P. Groscurth, and U. Haller. Anatomical model generation for laparoscopic surgery simulation. In *The Second Visible Human Project Conference Proceedings*, October 1998. 106
13. Denis Zorin, Peter Schröder, and Wim Sweldens. Interactive multiresolution mesh editing. In *Computer Graphics Proceedings 1997*, pages 259–268, Los Angeles, August 1997. ACM SIGGRAPH. 108