From Ziplock Snakes to Velcro^{TM} Surfaces

W. Neuenschwander*, P. Fua†, G. Székely*, and O. Kübler*

* Communication Technology Laboratory
Swiss Federal Institute of Technology
ETH
CH-8092 Zurich, Switzerland
Ph.: +41-1-632 5302, Fax: +41-1-632 1199
e-mail: waneu@vision.ee.ethz.ch

† SRI International
Artificial Intelligence Center
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
Ph.: (415)-859 4480, Fax: (415)-859 3735
e-mail: fua@ai.sri.com

Abstract

The use of energy minimizing deformable models in various applications has become very popular. The issue of initializing such models, however, has not received much attention although the model's performance depends critically on its initial state. We aim at obtaining good convergence and segmentation properties from a minimum of a priori information.

We present a new approach to segmentation of 2– and 3–Dimensional shapes that initializes and then optimizes a deformable model given only the data and a very small number of 2–D or 3–D seed points respectively. This is a valuable capability for medical, robotic and cartographic applications where such seed points can be naturally supplied. In effect, the 2–D "snake" and the 3–D surface model are clamped onto the object boundary in manner reminiscent of a ziplock or velcro being closed.

We develop the method's mathematic framework and show results using 2–D cartographic data. Preliminary results in 3–D using volumetric medical data are shown as well.

1 Introduction

In recent years, deformable models have emerged as a very powerful tool for semiautomated object delineation and surface modeling as well as for 2– and 3–Dimensional image segmentation in applications as diverse as medical imaging, graphics, robotics or terrain-modeling.

The 2–D models, known as "snakes" have been originated by Terzopoulos, Kass, and Witkin (Terzopoulos *et al.*, 1987; Kass *et al.*, 1988) and have since given rise to a large body of literature (Fua and Leclerc, 1990; Cohen, 1991) among many others, that explores theoretical and implementation issues as well as new applications.

In most of these papers, however, it is assumed that the initial position of the snake is relatively close to the desired solution. In effect, initialization here amounted to a painstaking almost complete, manual delineation of the desired outline. While this is a reasonable assumption for applications such as motion tracking (Bascle and

Deriche, 1993) where the effort is limited to the first image of a whole sequence, it is ineffective for delineating complex objects from scratch.

Our efforts are aimed at alleviating the often repetitive task practitioners face when segmenting images. In particular, we aim at eliminating the need to outline the desired structure very precisely. In our implementation, the user needs only to supply a few discrete points through which the contour must pass; the system then propagates the information along the contour starting from these points. As a result, the snake is progressively clamped onto an image contour so that it smoothly connects those points and has the right orientation at their locations. This behavior is analogous to the closing of a ziplock, hence the name of our snakes. As illustrated in Figure 1, considerably fewer control points are needed than for conventional implementations which we refer to as *traditional snakes*.

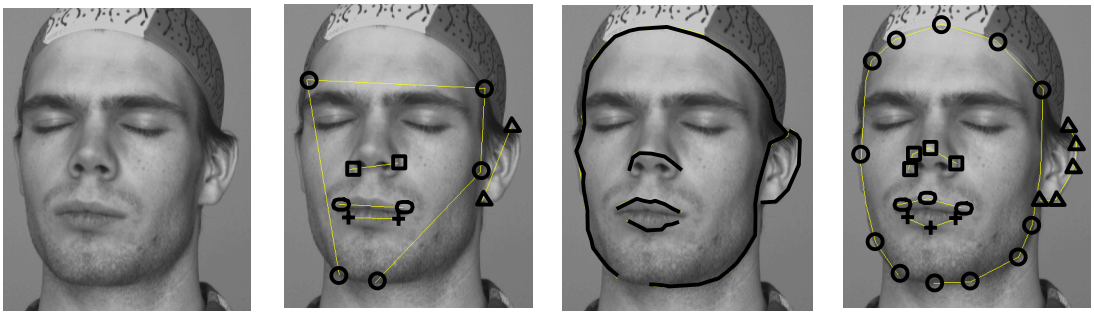


Figure 1: Outlining facial features. (a) A face image with low contrast contours. (b) Five sets of initial points, each denoted by a different symbol. Four of them are pairs of endpoints while the fifth is shown as a set of circles. (c) The contours delineated by Ziplock snakes. (d) The initial delineations that must be supplied to achieve the same result using traditional snakes.

In the same spirit, we present a method that allows a user to initialize and then optimize a 3–D surface model by supplying only a very small number of 3–D seed points and corresponding surface normals. This is a valuable capability because there are many applications, such as *Segmentation of 3–D shapes from volumetric data, Incremental construction of a world model by a mobile robot, Construction of composite models for high-resolution cartographic modeling* in which imposing initial conditions in this manner is both easy and natural.

Our technique relies on classical elastic models that are represented as triangulated meshes and deform themselves to minimize an objective function (Terzopoulos *et al.*, 1987). This method extends our 2–D ziplock approach (Neuenschwander *et al.*, 1994) to the 3–D domain. We show that a small set of 3–D seed points provides sufficient boundary conditions to solve the differential equation that governs the model's behavior in closed form, assuming that the data component of the objective function remains constant. As a result, it becomes possible to instantiate the model using these points alone by initially ignoring the data term and then propagating image information along the surface by progressively "turning it on". The triangulated mesh behaves like a piece of Velcro that is progressively being clamped onto the surface of interest, hence the name of our deformable surfaces.

2 Ziplock Snakes

The original snakes (Kass *et al.*, 1988) are modeled as time-dependent 2–D curves defined parametrically as $\vec{v}(s,t) = (x(s,t), y(s,t))_{0 \le s \le 1}$, where s is proportional to the arc length, t the current time, and x and y the curve's image coordinates. The snake deforms itself as time progresses so as to minimize an image potential $E_I(\vec{v})$, with $E_I(\vec{v}) = -\int_0^1 P(\vec{v}(s,t))\, ds$ where $P(\vec{v}(s,t))$ is a function of the image. One typical choice is to take $P(\vec{v}(s,t))$ to be equal to the magnitude of the image gradient, that is $P(\vec{v}(s,t)) = |\nabla I(\vec{v}(s,t))|$, where I is either the image itself or the image convolved by a Gaussian kernel.

Whatever the choice of P, $E_I(\vec{v})$ is typically not a convex functional. To overcome this problem, Terzopoulos (Terzopoulos *et al.*, 1987) has proposed to introduce a regularization term $E_D(\vec{v})$ that is convex and to minimize a total energy term $E(\vec{v})$ that is the sum of $E_I(\vec{v})$ and $E_D(\vec{v})$. Using the elastic rod model, $E_D(\vec{v})$ is taken to be

$$E_D(\vec{v}) = \frac{1}{2} \int_0^1 \alpha(s) \left| \frac{\partial \vec{v}(s,t)}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 \vec{v}(s,t)}{\partial s^2} \right|^2 ds \,, \tag{1}$$

where $\alpha(s)$ and $\beta(s)$ are arbitrary functions that regulate the curve's tension and rigidity. In the implementation described below $\alpha(s)$ and $\beta(s)$ are taken to be constant and supplied by the user. We have shown previously (Fua and Leclerc, 1990) that constant α and β can be chosen in a fairly image-independent way.

From variational calculus it is well known that if $\vec{v}(s)$ minimizes $E = E_D + E_I$ and is sufficiently regular, that is at least $C^4(0,1)$, then it must be a solution of the set of two coupled Euler differential equations

$$-\alpha \frac{\partial^2 v(s,t)}{\partial s^2} + \beta \frac{\partial^4 v(s,t)}{\partial s^4} = -\nabla P(v(s,t)) \tag{2}$$

where $v(s,t)$ stands for either $x(s,t)$ or $y(s,t)$. Note that, in order for this equation to have a unique solution, one must specify boundary conditions such as the values and derivatives of $v(s,t)$ for $s = 0$ and $s = 1$.

Discretizing Equation 2 using finite differences yields the linear system

$$K \cdot V = F_V \tag{3}$$

where V stands for the vector of either x or y coordinates, K is the stiffness matrix and F_V are the derived image forces. K is *not* invertible and these equations cannot be solved directly.

2.1 Initialization and Optimization using Boundary Conditions

To improve upon the snakes' convergence properties, we now introduce our ziplock optimization mechanism. The user is expected to specify endpoints in the vicinity of a clearly visible edge segment, which implies a well-defined edge direction.

To successfully optimize our snake, we start from an initial position that is approximately correct in the neighborhood of the endpoints. The easiest way to achieve

this result is to solve the homogeneous equations that correspond to the system of Equation 2

$$-\alpha\frac{d^2v(s)}{ds^2}+\beta\frac{d^4v(s)}{ds^4}=0 \tag{4}$$

where v stands for either $x(s)$ or $y(s)$ and $0 \leq s \leq 1$ and has to fulfill the four boundary conditions: $v(0)=\hat{v}_0$, $v(1)=\hat{v}_1$, $v'(0)=\hat{v}'_0$, $v'(1)=\hat{v}'_1$. By construction, this solution has the specified tangent at the endpoints $\hat{v}_0, \hat{v}_1$ and is close to the right answer near these points.

While the tangent direction at the endpoints can be computed, its orientation cannot be determined therefore the interface provides the user with the possibility of flipping the orientation at both ends if necessary. By fixing the curve's endpoints and giving the curve's tangent at those points, the system of Equation 3 is reduced by 4 entries (see appendix of (Neuenschwander *et al.*, 1995)) to

$$K^* \cdot V^* = F^*_{V^*}, \tag{5}$$

where V^* stands either for the reduced X^*- or Y^*-vector, and K^* is a pentadiagonal stiffness matrix that *is* invertible. Of course, since $F^*_{V^*}$ depends on the snake's current position, the system is only semilinear and cannot, in general, be solved in closed form.

We start the optimization of the energy term by defining the initial snake as the solution of the homogeneous differential system of Equation 4. At this stage the snake "feels" absolutely no external potential forces. During the ongoing iterative optimization process the image potential is turned on progressively for all the snake vertices, starting from the fixed extremities. We distinguish between *passive* and *active* snake nodes, depending on whether the potential force field is turned on for that vertex or not.

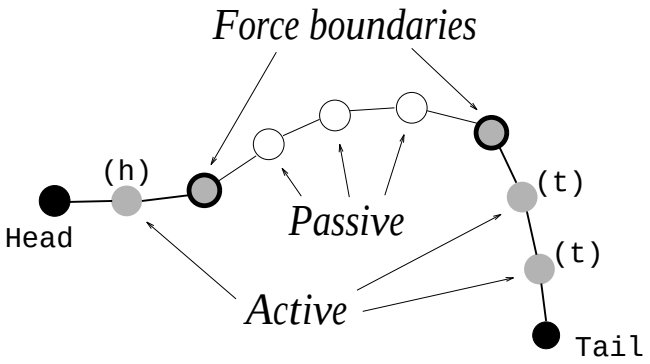


Figure 2: Schematic Ziplock Snake during optimization. A Ziplock Snake, fixed at `Head` and `Tail`, consists of two parts the *active* and the *passive* vertices. These areas are separated by moving *force boundaries*. The *active* part of the snake is divided up into two segments marked as `(h)` and `(t)` respectively.

As illustrated by Figure 2, we define the *force boundaries* as the location of the vertices farthest away from the endpoints that feel the image forces. These boundaries approach each other during the ongoing optimization process by moving forward.

The simplest way to optimize the snake would be to gradually move the force boundaries from the snake's head and tail towards its center and solve Equation 5 for each new position. However, because Equation 5 is semilinear, it cannot be solved in one single step. To enforce stability we introduce a viscosity term $\gamma(s,t)$ similar to the one used by traditional snakes and iteratively solve the Equation

$$(K^* + \gamma_t I) \cdot V_t^* = \gamma_t V_{t-1}^* + F_{V^*}^*\Big|_{V^* = V_{t-1}^*} \quad . \tag{6}$$

The initial coordinate vector $(X_0^*, Y_0^*)^T$ is the solution of Equation 4. The viscosity $\gamma(s,t)$ is initialized as $\gamma(s,0) \equiv 0$, $s \in [0,1]$ and recomputed each time the force boundaries move so that the initial displacement of each vertex is on the average of a given a-priori magnitude, typically 1 pixel, see (Fua and Leclerc, 1990). Each force boundary is moved individually but at most one vertex per iteration. A force boundary is moved only if the average motion of the *active* part of the snake, feeling the image forces and lying between the force boundary and the head `(h)` or tail `(t)`, respectively, is below a fraction of a pixel. The two different parts are marked with `(h)` and `(t)` in Figure 2.

The potential of Ziplock Snakes has already been illustrated in Figure 1. In Figure 3 we use Ziplock Snakes to outline various features of a house in an aerial image. Using our snakes we can delineate rather complex object boundaries using a very small number of seed-points.

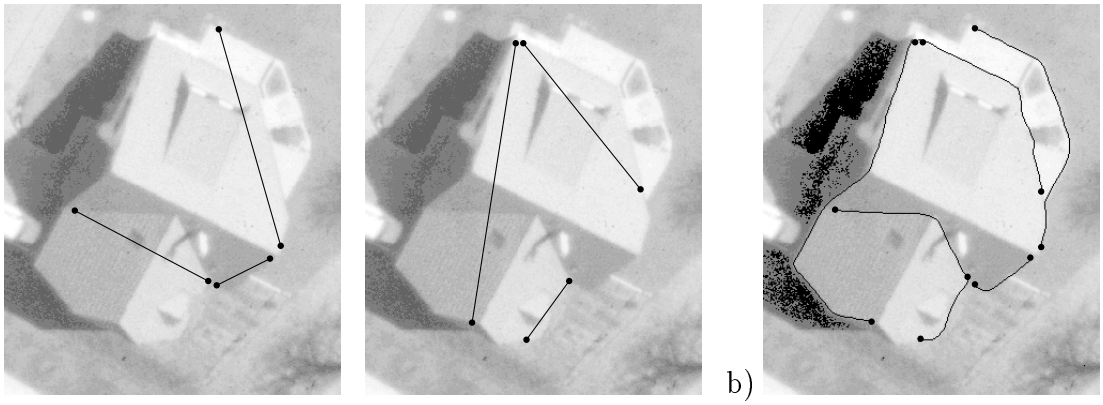


Figure 3: Delineating man-made objects in aerial images. a) the two figures show complementary sets of pairs of seed-points used to initialize Ziplock Snakes. b) Outlined features by Ziplock Snakes. Courtesy of the Institute of Geodesy and Photogrammetry, ETH.

2.2 Extension to Ribbon Snakes

Following (Fua and Leclerc, 1990), we have implemented a tool for interactive road delineation, called *ribbon snake*. The model vector $\vec{v}$ is augmented by a third component, the varying width $w(s,t)$ of the road. The expression for the deformation energy (1) still holds for $\vec{v}(s,t) = (x(s,t), y(s,t), w(s,t))^T$, where the width is subject to "tension" and "rigidity" constraints as the two coordinate components. The ribbon forms the center of the road, while the assigned width defines two curves

that are the actual deforming road boundaries. Note that the image information is taken into account along these two curves only. The sequence in Figure 4 depicts the initialization and the subsequent "Ziplock"-optimization of a ribbon snake model. Each ribbon is initialized by the two corresponding endpoints and the road direction, which is the average direction of the left and right road boundary.

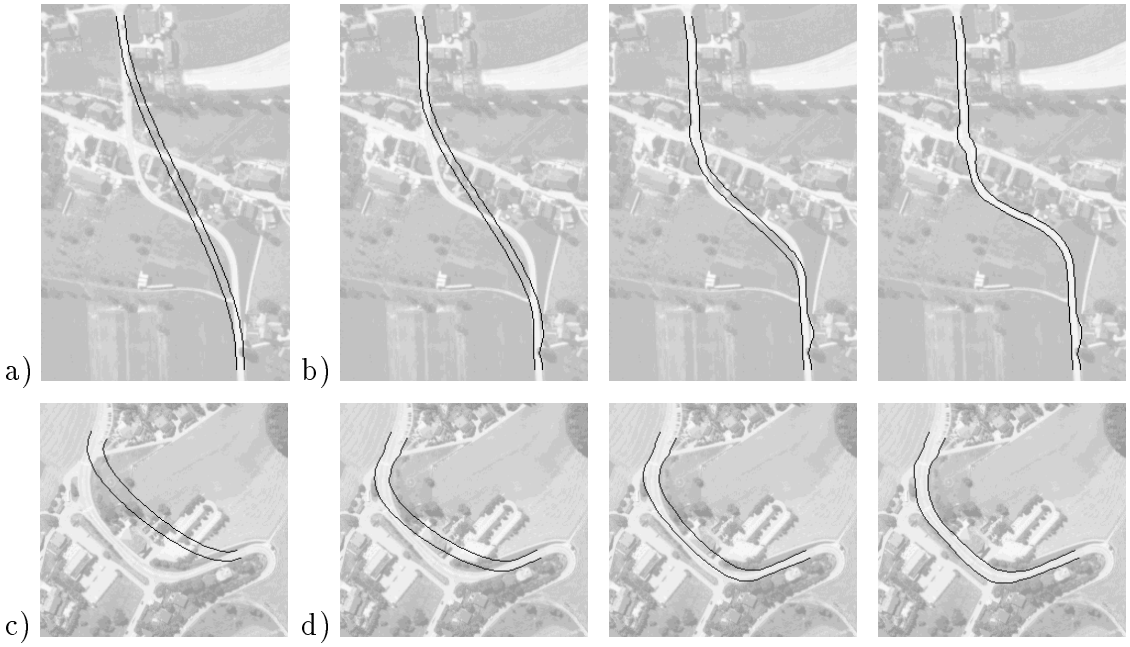


Figure 4: Road delineation in aerial images. a) & c) Initial ribbon snakes defined by the two fixed endpoints and the corresponding road direction. b) & d) The three images show intermediate and final results using the presented "Ziplock"-optimization mechanism. Note, that the ribbon snake can pass over other significant image features due to the fact that the image forces are not active between the force boundaries.Courtesy of the Institute of Geodesy and Photogrammetry, ETH and the Schweizer Landestopographie, Bern.

3 Deformable Velcro^{TM} Surfaces

Velcro surfaces as well as Snakes belong to the same framework of deformable models proposed by Terzopoulos (Terzopoulos *et al.*, 1987). The physical interpretation uses basics of elasticity theory and regards deformable surfaces as a weighted combination of membrane and thin plate. A generalized deformable surface model is defined as $\vec{v}(\boldsymbol{\omega},t) = (x(\boldsymbol{\omega},t), y(\boldsymbol{\omega},t), z(\boldsymbol{\omega},t))$ where $\boldsymbol{\omega} \in \Omega \subset I\!R^2$ is a suitable parametrization, t the current time, and x, y and z are the corresponding coordinate functions of the surface. The surface deforms itself so as to minimize its image potential energy $E_I(\vec{v}) = -\iint_\Omega P(\vec{v}(\boldsymbol{\omega}))d\boldsymbol{\omega}$ where P is the same potential function as derived in Section 2.

In the case of deformable surfaces, the convex regularization term $E_D(\vec{v})$ we use has the following form:

$$E_D(\vec{v}) = \iint_\Omega \tau(\boldsymbol{\omega}) \left[\left| \frac{\partial \vec{v}}{\partial \omega_1} \right|^2 + \left| \frac{\partial \vec{v}}{\partial \omega_2} \right|^2 \right] + \rho(\boldsymbol{\omega}) \left[\left| \frac{\partial^2 \vec{v}}{\partial \omega_1^2} \right|^2 + 2 \left| \frac{\partial^2 \vec{v}}{\partial \omega_1 \partial \omega_2} \right|^2 + \left| \frac{\partial^2 \vec{v}}{\partial \omega_2^2} \right|^2 \right] d\boldsymbol{\omega} \quad (7)$$

where $\rho(\boldsymbol{\omega}) = 1 - \tau(\boldsymbol{\omega})$ and $\boldsymbol{\omega} = (\omega_1, \omega_2)$. In the implementation described below, we take the surface tension parameter, τ, to be a constant between 0 and 1 supplied by the user. The Euler differential equation according to the optimization problem of the functional $E(\vec{v}) = E_I(\vec{v}) + E_D(\vec{v})$ can be written as

$$-\tau\,\Delta\boldsymbol{v} + (1-\tau)\Delta^2\boldsymbol{v} = -\frac{\partial P}{\partial \boldsymbol{v}} \tag{8}$$

where $\boldsymbol{v}$ stands for either x, y or z. Note, these differential equations do not have a unique solution in the absence of boundary conditions, whose incorporation is crucial for a successful segmentation.

We use the same notations as introduced in Section 2 for the 3–D framework and refer to equations derived in 2–D. Note, however, that equivalence is restricted to the notation while these expressions differ in their detailed mathematical structure.

In the context of segmentation of anatomical organs in 3-D medical image data, we assume that the surface is topologically equivalent to a sphere. We tessellate the surface into triangular patches and therefore define the discrete model by a set of nodes. The user manually supplies a few "anchor-points" and supervises the tessellation by specifying the triangular facets and their refinement. Using finite differences, the governing Equation 8 becomes the linear system

$$K \cdot V = F_V \tag{9}$$

where now V stands for the vector of either x, y or z coordinates, K is the surface's stiffness matrix and F_V are the derived image forces. As it is the case for Snakes, the matrix K is *not* invertible and these equations cannot be solved directly.

3.1 Initialization and Optimization Using Boundary Conditions

First we address the acquisition of boundary conditions and their use to effectively initializing the deformable surface model. We then discuss the actual optimization.

Visual inspection of 3–D data sets simultaneously from the three cardinal directions is supported by appropriate user interfaces and allows to identify surface points interactively. The normal vectors are taken as the gradient's direction of the potential field at the selected points. This process yields a set $\mathcal{B}$ of boundary conditions defined by an arbitrary number m of 3–D anchor-points P_i ($4 \leq i \leq m$, non-coplanar) and the corresponding surface normals $\vec{n}_{P_i}$. Therefore $\mathcal{B}$ can be written as

$$\mathcal{B} = \{\vec{v}_1 = P_1, N_{\vec{v}_1} = \vec{n}_{P_1}\ ;\ \vec{v}_2 = P_2, N_{\vec{v}_2} = \vec{n}_{P_2}\ ; \ldots ; \vec{v}_m = P_m, N_{\vec{v}_m} = \vec{n}_{P_m}\} \tag{10}$$

To successfully optimize the surface, we must start from an initial shape that is approximately correct in the neighborhood of the selected anchor-points. This result can be achieved by solving the homogeneous equations that correspond to the system of Equation 8

$$-\tau\,\Delta\boldsymbol{v} + (1-\tau)\,\Delta^2\boldsymbol{v} = 0 \tag{11}$$

where $\boldsymbol{v}$ stands for either x, y or z. The initial surface we compute is a solution of Equation 11 that satisfies the set $\mathcal{B}$ of boundary conditions defined above. By

construction, this solution will pass through the anchor-points and have the specified normal vectors there, and be close to the final surface near these points.

The semilinear system of Equation 9 is reduced by incorporating boundary conditions (set $\mathcal{B}$) to

$$K^* \cdot V^* = F^*_{V^*}, \tag{12}$$

where now V^* stands either for the reduced X^*-, Y^*- or Z^*-surface vector, and K^* is the sparse stiffness matrix that *is* now invertible.

We start the optimization of the energy term by defining the initial surface as the solution of the homogeneous differential system of Equation 11. At this stage the surface "feels" absolutely no external potential forces. During the ongoing iterative optimization process the image potential is turned on progressively for all the surface nodes, starting from the seed points. We distinguish between *passive* and *active* surface nodes, depending whether the potential force field is turned on for that vertex or not. We define the expanding *force fronts* as the location of the nodes farthest away from their corresponding central seed point that feel the image forces. These fronts approach each other during the ongoing optimization process and propagate all over the surface by moving forward like concentric wave fronts. This propagation is illustrated in Figure 5.

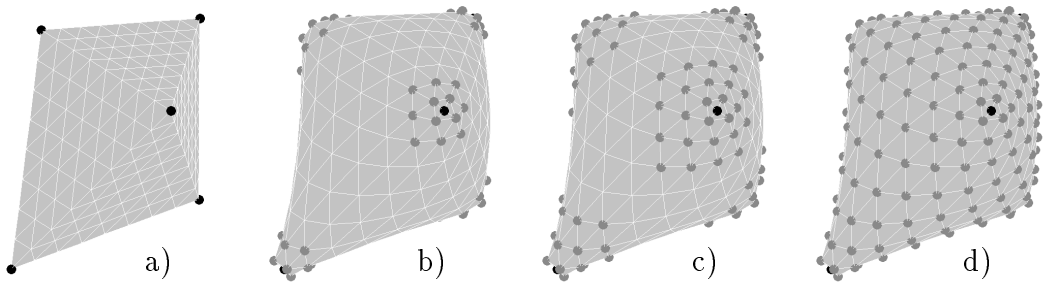


Figure 5: Propagation of the force boundaries. a) First polyhedral initialization performed by the user. b)-d) Solution of the Euler differential equation. At every seed point a force front expands on the surface. The black dots denote the various fix points while the light shaded vertices denote the active mesh nodes. Note, a case without image forces was used for this illustration.

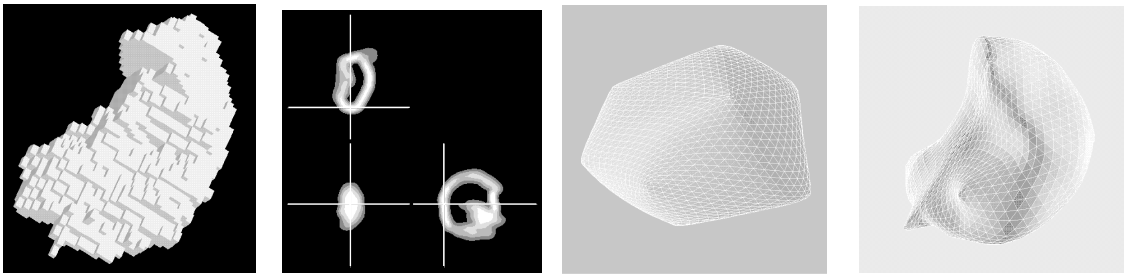
As explained in Section 2, to enforce stability we introduce a viscosity term $\gamma(\boldsymbol{\omega}, t)$ and iteratively solve the equation

$$(K^* + \gamma_t\, I) \cdot V^*_t = \gamma_t\, V^*_{t-1} + F^*_{V^*}\Big|_{V^* = V^*_{t-1}} . \tag{13}$$

where now the initial coordinate vector $(X^*_0, Y^*_0, Z^*_0)^T$ is the solution of Equation 11. The viscosity $\gamma(\boldsymbol{\omega}, t)$ is initialized as $\gamma(\boldsymbol{\omega}, 0) \equiv 0\ , \boldsymbol{\omega} \in \Omega$ and recomputed each time the force boundaries move.

We illustrate the segmentation process on the example of the putamen, a relatively small nucleus of the deep gray matter of the brain. The original binarized dataset is shown in Figure 6a). First the user has to provide a reasonable number of anchor-points, which will be used to generate the initial approximation of the

surface. At the same time they serve as seed points for the force front propagation of the surface evolution. We have built a special tool, which serves as a 3–D cursor in a gray-valued field, and is illustrated by Figure 6b). The tool simultaneously updates the three orthogonal planes that cross each other at the actual 3–D cursor position, which is shown as cross-hair on each of the planes.



a) b) c) d)

Figure 6: a) The original dataset used as an example to illustrate the segmentation process. b) Tool for the selection of fix points in the dataset to be segmented. c) Initial surface resulting from the force-free solution of the homogeneous system of Euler-Lagrange differential equations, and its refined triangulation. d) The final result of the surface evolution, governed by the progressive turning on of the image potential forces. Note, that the resulting surface is smooth, but features which are selected by the user as anchor-points are preserved.

The Delaunay-tetrahedralization (Boissonnat, 1984) of the selected anchor-points is then computed and "sculpted" interactively by successive deletion of Delaunay tetrahedra from the convex hull, until all selected points lie on the surface. Finally the remaining tetrahedra are used to define an initial triangulation that is refined by a number of recursive subdivisions—4 in the example presented here resulting in 1794 vertices and 3584 facets—and used to compute the unperturbed initial surface, as shown in Figure 6c). This solution is then optimized according to the procedure proposed in this section. Figure 6d) shows the result of this surface evolution.

4 Conclusion and Future Work

We have proposed a snake-based approach to semiautomated delineation that allows a user to outline an open contour by specifying only very distant endpoints and allowing the computer to propagate edge information from the extremities toward the center. In other words, we have proposed a natural initialization procedure that is completely in line with both the practitioner's task and the mathematical problem so that the "expert user" (Kass *et al.*, 1988) of the original snake papers need not be that much of an expert anymore. Ziplock snakes have been ported into the RADIUS Common Development Environment (Mundy *et al.*, 1991) where they can be used to optimize or connect curves and ribbons in a semiautomated fashion while taking advantage of RCDE's extensive editing capabilities.

We have generalized the 2–D method to a 3–D procedure that allows initializing and optimizing a 3–D surface model given only a very small number of 3–D

seed points and corresponding surface normals. This is valuable in semiautomated applications—such as medical ones—where seed points can be supplied manually by the user with reasonable ease. The user can influence the level of detail in the final representation by choosing the most appropriate force field, adjusting the tension parameter, and by providing more or less seed points for any given part of the surface.

The capability presented here should also be valuable for fully automated applications. Vision algorithms—such as the so-called "shape from X" methods—often provide high-quality results in some parts of a scene but may be unreliable in others. Our method has the potential to allow the use of the most reliable surface patches as anchors and the propagation of information to other parts of the surface.

Further development of deformable Velcro surfaces will primarily address the implementational and the user interface level. The governing objective is to develop the mathematical theory, the implementation, and the user interface of a tool for the segmentation of 3–D objects without requiring expert Computer-Vision knowledge for its use.

References

Bascle B. and R. Deriche (1993). Stereo Matching, Reconstruction und Refinement of 3D Curves Using Deformable Contours. In *International Conference on Computer Vision*, pp. 421–430, Berlin, Germany.

Boissonnat J.-D. (1984). Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, Vol. 3,pp. 266–286.

Cohen L.D. (1991). On Active Contour Models and Balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 53,No. 2,pp. 211–218.

Fua P. and Y.G. Leclerc (1990). Model Driven Edge Detection. *Machine Vision and Applications*, Vol. 3,pp. 45–56.

Kass M., A. Witkin, and D. Terzopoulos (1988). Snakes: Active Contour Models. *International Journal of Computer Vision*, Vol. 1,No. 4,pp. 321–331.

Mundy J.L., R. Welty, L. Quam, T. Strat, W. Bremmer, M. Horwedel, D. Hackett, and A. Hoogs (1991). The RADIUS Common Development Environment. In *Proc. of AIPR, Washington, DC.* (also *in Proc. of DARPA Image Understanding Workshop*, San Diego, California, 1992.).

Neuenschwander W., P. Fua, G. Székely, and O. Kübler (1994). Initializing Snakes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 658–663.

Neuenschwander W., P. Fua, G. Székely, and O. Kübler (submitted, 1995). Ziplock Snakes. *International Journal of Computer Vision*, Vol. .

Terzopoulos D., J. Platt, A. Barr, and K. Fleischer (1987). Elastically deformable models. *Computer Graphics*, Vol. 21,No. 4,pp. 205–214.