User-Defined B-Spline Template-Snakes

Tim McInerney[1,2] and Hoda Dehmeshki[1]

[1] Dept. of Math, Physics and Computer Science, Ryerson Univ., Toronto, ON M5B 2K3, Canada

[2] Dept. of Computer Science, Univ. of Toronto, Toronto, ON M5S 3H5, Canada

Abstract. We combine a new user initialization process with a B-spline snake to create a model with the properties of a deformable template. This 'template' snake can be constrained by its control polygon and is initially extremely close to, and similar in shape to, the target anatomical structure. The initialization process acts as almost a pre-segmentation and labelling step, making the snake's task much simpler and hence more likely to succeed in noisy images without subsequent user editing. By imposing an order on the initialization process, the user is able to transfer knowledge of global shape, symmetry, landmark position etc. to the model. We apply our snake to the segmentation of 2D medical images.

1 Introduction

Rapid and accurate segmentation of anatomical structures from medical images is a persistent problem that continues to impede the timely analysis of these structures. Robust, fully automatic segmentation systems have proved extremely challenging to develop. Consequently, a more immediate and significant impact on MIA may be realized by optimizing the capabilities of semi-automatic techniques, to the point where only a small amount of time and labor is required to process complex data sets. To achieve this goal, the recognition capabilities of the human expert must be fully exploited. Semi-automatic techniques that assist the human expert in the extraction of the structures must be designed to not only be fast and intuitive, but also permit the interactive transfer of structure shape and appearance knowledge from the expert in order to ensure segmentation accuracy, robustness and reproducibility with a minimal user editing phase. This is especially important when processing a large number of image slices from a volume image or a time series, or when processing very noisy images.

Active contour models (Snakes) and their variants have become widely popular in medical image segmentation and are still intensively applied and researched. The difficult challenge in improving these techniques is to develop more effective user initialization mechanisms, along with control mechanisms that can guide the optimization-driven segmentation process at an appropriately high level of abstraction [1]. One way to achieve this is to have the human expert recognize landmarks and other critical shape features and transfer this

R.E. Ellis and T.M. Peters (Eds.): MICCAI 2003, LNCS 2879, pp. 746–753, 2003.

information in such a way that the snake is explicitly 'aware' of where it is in the image, how its 'parts' are arranged, and what structure it is segmenting.

We apply some of these ideas in a semi-automatic segmentation context by prescribing custom snake initialization processes for each anatomical structure. The intuitive and general initialization process makes use of simple line primitives that are quickly drawn across the target structure at critical points in a pre-specified order. These line primitives are then used to construct a control polygon of a finite element B-spline snake [2]. By taking advantage of the properties of B-splines, we are able to create a model more like a template - a snake constrained by its control polygon that is initially extremely close to and similar in shape to the target structure. The initialization process acts as almost a pre-segmentation and labelling step, making the model's job much simpler and hence more likely to succeed without user editing. By drawing lines in a pre-specified order *across* the target structure, the human expert is able to transfer knowledge of structure shape, image appearance etc. to the model. This information can then be utilized by a high-level snake fitting algorithm. Finally, the recognition and identification of critical shape features by the expert also provides key information to subsequent shape analysis.

2 Motivation and Background

The classical snakes model, introduced by [3], is typically initialized by tracing a rough curve near the target boundary. This process is somewhat tedious and error prone, and often results in the snake latching onto spurious or neighbor structure boundaries. A correction step is then required to pull the snake off the incorrect boundaries into the correct position. Furthermore, in noisy regions the user is required to impose additional constraints, in the form of 'pin' points for example. These problems are even more apparent when processing a number of slices in a volume image or a time series.

Since an accurate initialization is needed in order for the snake to lock onto the correct image features, researchers have been actively investigating techniques to mitigate the sensitivity of snakes to their initialization. Among these techniques is the use of an inflation force [4], gradient vector flow fields [5], and the use of automatic snake element subdivision methods [6] [7][8]. These techniques can work well if the image feature map is relatively clean. However, most clinical images are noisy, contain many uninteresting edges, or texture is present. Hence, these more automated techniques do not work as expected and are sensitive to parameter settings. Livewire is a recently proposed interactive boundary tracing technique [9,10]. While an effective and efficient method for many objects, it is still fundamentally tracing-based and may require considerable user interaction and user concentration for noisy clinical images. In addition, Livewire is not as amenable for segmenting multiple image slices in a time series or in volume images or utilizing more automatic high-level fitting algorithms as are snakes, and its segmentation editing semantics are limited.

The most robust model-based techniques are arguably deformable template models, which are typically designed to be fully automatic and carefully handcrafted or trained for one anatomical structures and image modality. These models incorporate some form of prior information about object shape and/or object image intensities [11][12]. The success of these models often lies in how effectively they utilize higher-level object shape and image appearance information that manifests itself at multiple scales and locations with respect to the object. General interactive techniques such as snakes and livewire typically utilize small scale, local boundary information due primarily to their user initialization and manipulation processes.

3 User-Defined B-Spline Template Snakes

In order to optimize the performance of semi-automatic models, they must fully exploit user image interpretation. Initialization processes that use tracing-like actions around the object boundary are inherently limited in this ability. In this paper, we introduce a new initialization process, coupled with the power of a B-spline curve, to create a semiautomatic snake that has the properties of a deformable template. The snake is efficiently initialized such that it is very close in shape and position to the target object and such that it knows its position with respect to the object. This 'template' snake is built on top of an existing powerful and general snakes package [2], hence there is no need to use predefined, restrictive shape representations like superquadrics. Furthermore, B-spline snakes have many desirable properties. They are a compact, parameterized model with a control polygon which can be used for global deformation control and customized deformation handles.

3.1 Finite Element B-Spline Snakes

In this section, we briefly review the formulation of finite element B-spline snakes. For details, we refer the reader to [2].

A snake is a time-varying parametric contour $\boldsymbol{v}(s,t) = (x(s,t), y(s,t))^\top$ in the image plane $(x,y) \in \Re^2$, where x and y are coordinate functions of the parameter $s \in [0, L]$ and t is time. In a finite element formulation, the parametric domain $0 \leq s \leq L$ is partitioned into finite sub-domains, so that the snake contour is divided into "snake elements". Each element e is represented geometrically with shape functions $\boldsymbol{N}(s)$ involving shape parameters $\boldsymbol{u}^e(t)$. The shape parameters of all the elements are collected together into the snake parameter vector $\boldsymbol{u}(t)$. This leads to a discrete form of the equations of motion, which govern the shape of the dynamic contour, as a system of second-order ordinary differential equations in $\boldsymbol{u}(t)$:

$$\mathbf{M}\ddot{\boldsymbol{u}} + \mathbf{C}\dot{\boldsymbol{u}} + \mathbf{K}\boldsymbol{u} = \boldsymbol{F}, \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix, and $\boldsymbol{F}$ is the external force vector. The external forces consist of image forces and user constraint forces. The image forces are typically the negative gradient of some image potential function $-\nabla P_I(\boldsymbol{v})$. The user may guide the dynamic snake via time-varying interaction forces $\boldsymbol{f}(s,t)$, typically applied via an input device, driving the snake out of one energy minimizing equilibrium and into another, or these forces may also be derived from other user-initiated constraints. The stiffness matrix $\mathbf{K}$ is assembled from element stiffness sub-matrices $\boldsymbol{K}^e$ that depend on the shape functions $\boldsymbol{N}$ (the matrices $\mathbf{M}$, $\mathbf{C}$, and the vector of nodal external forces $\boldsymbol{F}$ are assembled in a similar way and also depend on $\boldsymbol{N}$).

An analytic form of the external forces is generally not available. Therefore, Gauss-Legendre quadrature may be employed to approximate the value of the integral for the element external force vector $\boldsymbol{F}^e$. For element e_i we have

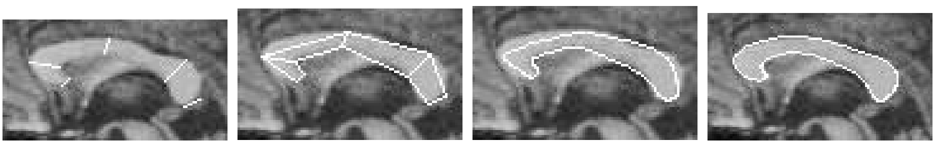
$$\boldsymbol{F}^{e_i} = l \sum_j \rho_j \boldsymbol{N}_h(\xi_j)^\top (-\nabla P_I(\boldsymbol{v}(\xi_j))), \tag{2}$$

where ξ_j and ρ_j are the jth Gaussian integration point and its corresponding weighting coefficient, respectively, and l is the element parametric length.

For B-spline shape functions, $\boldsymbol{v}(s)$ is constructed as a weighted sum of N_B basis functions $B_n(s)$, $n = 0, ..., N_B - 1$ as follows: $\boldsymbol{v}(s) = \boldsymbol{B}(s)^\top \boldsymbol{Q}$, where $\boldsymbol{B}(s) = [\boldsymbol{B}_0(s), ..., \boldsymbol{B}_{N_B-1}(s)]^\top$, $\boldsymbol{Q} = [\boldsymbol{p}_0, ..., \boldsymbol{p}_{N_B-1}]^\top$ and $\boldsymbol{p}_i$ are the control points. A B-spline span serves as an element in our finite element formulation. Consequently, we determine the nodal variables (*i.e.,* snake shape parameters - the position of the control points in a B-spline snake), the shape matrix, and the assembling matrix associated with a span.

3.2 B-Spline Template Snake Initialization Process

The new initialization process is simple but very effective. The user uses a mouse or pen input device to add cross-sectional lines to the target structure. A point is 'clicked' on one side of the object boundary and a line is stretched and rotated interactively to a point on the opposite boundary. For end cap regions of objects, the user draws lines approximately tangent to the region.



(a) (b) (c) (d)

Fig. 1. Example of initialization process. In (a) the user enters lines, starting from the left side of the corpus callosum and proceeding to the right. In (b) the B-spline control polygon is shown, (c) initial B-spline snake, (d) segmentation result.

The user adds lines in a prescribed order for a particular object, and in prescribed critical locations, such as landmark points. For example, to segment the corpus callosum (CC), the user starts at the end cap region near the rostrum (extreme left). A small line, tangent to the end cap region is drawn (Fig. 1a). The user then identifies the genu and draws a line across the CC in this region. The user then draws two additional lines, where the first line roughly divides the CC into half and the next line demarcates the splenium region. The splenium is then identified and a line is drawn tangent to the splenium end cap region. The user may also then use additional lines to demarcate the fornix if it appears attached to the CC. This process, once learned, is fast and intuitive. Drawing cross-section lines is natural and is less tedious than tracing around an object. Figure 1(b) shows the resulting control polygon, Fig. 1(c) shows the initial B-spline snake, while Fig. 1(d) shows the final segmentation.

Once the prescribed lines are drawn (or alternatively, as the lines are drawn), the algorithm uses them to automatically construct a customized control polygon, and displays the resulting B-spline curve. The user may also 'click' on control polygon edges or control points, and then add new lines or control points. The control polygon is updated and the new curve displayed. These new lines and points may be added during initialization or while the snake is running. Using this simple but effective process, the user recognizes critical points and regions in a specified order, and transfers this knowledge to model. Knowledge of global shape, such as width, is transferred, and the template snake is 'aware' of it's position with respect to the object. The snake is parameterized using optimally-placed minimum number of degrees of freedom (DOF). Thus, the model is more like a deformable template than a local snake model - it is less sensitive to noise and more amenable to propagation to subsequent image slices in a volume image or time series. Unlike a traditional deformable template model however, it is constructed and positioned by the user rather than preconstructed and automatically initialized by the segmentation system.

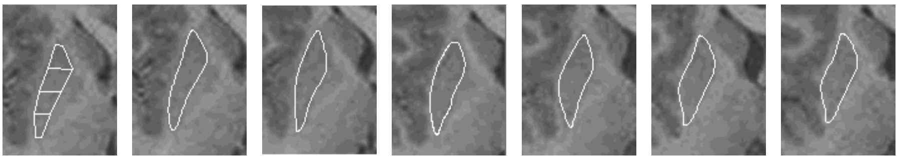
3.3 User-Customizable External Image Forces and Constraints

The initialization process results in each snake element or span roughly corresponding to a specific object boundary segment. This information is used to construct user-programmable object-specific external image forces. Among the features we have implemented are the following:

- The number of points in each element, which we term snake points, at which to compute image forces can be specified. For example, the number of snake points can be matched to the image resolution, so that there is roughly one snake point for each pixel along the element. The forces computed at a snake point are then distributed to the corresponding control points (2). This feature makes the snake less sensitive to noise or spurious image edges.
- For each snake element, a search along the normal direction at a snake point is carried out, for a small, user-specified distance (typically only two or three pixels). The search criteria can be set for the strongest edges or edges with a specific magnitude. Since the initial model is close in shape to the object, edge direction can also be used if desired. Local image intensity statistics along and around the user input lines can also be gathered and used to set local image thresholds. If a matching edge point is found, a spring force is applied to attract the snake point to it. If no matching edge is found (in the

case of a boundary gap or noise), this point does not contribute to the image forces.

- The end points of the user input lines can be used very effectively as soft 'pin' constraints. Points on the snake closest to these boundary points are attracted to them by a spring force.
- The B-spline control polygon is a coarse approximation of the curve and hence a coarse approximation of the target object boundary (Fig. 1b). It is therefore a convenient frame upon which to build global model deformation control. With the push of a button, the control points can be connected with springs (including control points on opposite sides of the polygon), and these spring constraint forces can be included on the right hand side of equation (1). The control polygon acts as a spring-mass lattice, constraining the global shape or symmetry of the snake (Fig. 2). Many useful user-definable spring constraint arrangements are also possible.



(a) (b) (c) (d) (e) (f) (g)

Fig. 2. Segmenting the putamen from MR volume image. In (a) the user enters lines in the first image and turns on the control polygon springs to act as global shape constraints. In (b) to (g) the snake 'tracks' the putamen.

4 Experimental Results

We have applied our B-spline template-snake to several 2D images. In Fig. 1 we show the initial snake and the final segmentation of the CC for several 2D mid-sagittal MR brain images. The algorithm has been tested on 26 CC images, using five user input lines each. The average error when compared against expert manual segmentations is 0.6 pixels, where the error is defined as the shortest distance between the snake points and the expert segmented boundaries. This error can be reduced by using additional program-added degrees of freedom or by additional user input lines for some CC's. Note that once the input lines are entered, no further user editing of the snake is needed. In Figure 4 we show the initial user lines, the resulting initial snake and the final segmentation of the arm bone in an x-ray image. This image is very noisy, especially where the two bones overlap. There are many large gaps in the edges of the bone boundary and many spurious edges inside the bone. Notice how, with only a few input lines, the initial snake is almost the same shape as the bone. The model ignores edges that are not of a specific magnitude and direction.

Figure 2 demonstrates the use of the global shape constraints. The goal is to segment the putamen from several slices of an MR image volume. The putamen is adjacent to the gray matter and to the globus pallidus - both with highly similar intensity to the putamen. Consequently, there are many large gaps in the putamen boundary, and texture in the interior. The user enters lines in the first image and turns on the control polygon spring constraints (Fig. 2(a)). The snake is able to successfully 'track' the putamen in neighboring slices with no user editing. Without the shape constraints, many user interventions are required to correct the segmentation. If a large shape change occurs between slices, the constraints springs also provide more efficient user editing of the snake.

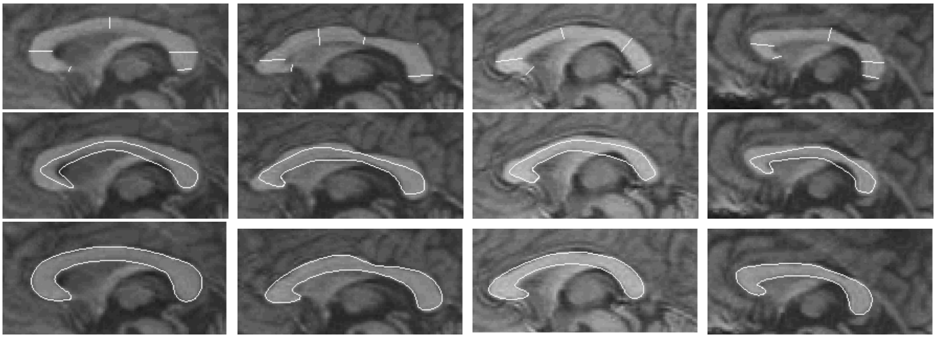
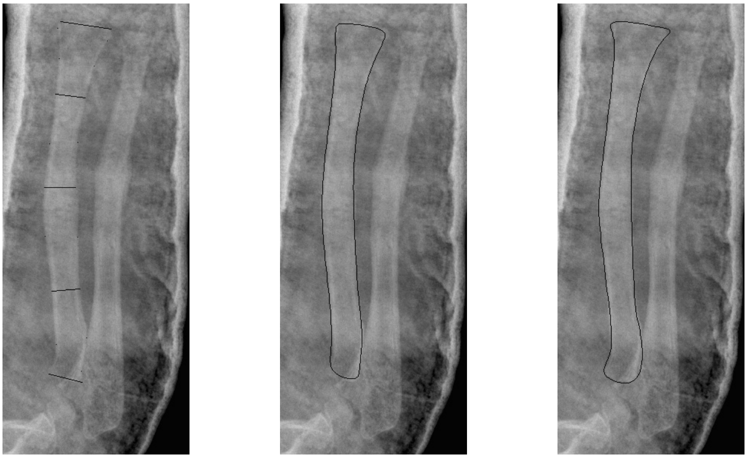


Fig. 3. Example corpus callosum segmentations using a B-spline snake. The first row shows the user input lines, the second row shows the initial B-spline snake and the final row shows the result.

5 Conclusion

We have created a user-definable deformable template model using a B-spline snake. A simple but effective and efficient initialization process, coupled with the properties of a B-spline, enables the construction of a snake that is extremely close to and similar in shape to the target anatomical structure. This allows the user to create customized external forces and utilize custom fitting algorithms, ensuring a more robust and automatic segmentation result. The B-spline control polygon provides a framework for imposing global shape constraints. We are also moving towards a NURBS-based snake, which will allow us to use control points as well as their associated weights, and nonuniform subdivision to create even more accurate initial template snakes. Future work involves the design of customized model fitting algorithms that will minimize the number of user lines required for initialization and the extension of the technique to 3D using deformable Doo-Sabin and NURSS subdivision surface models.



(a) (b) (c)

Fig. 4. Segmenting a bone from a noisy x-ray image. (a) user input lines (b) initial B-spline snake (c) segmentation result.

References

1. McInerney, T., Hamarneh, G., Terzopoulos, D.: Deformable organisms for automatic medical image analysis. Medical Image Analysis **6** (2002) 251–266
2. Liang, J., McInerney, T., Terzopoulos, D.: United snakes. In: Proc. Seventh International Conf. on Computer Vision (ICCV'99), Kerkyra (Corfu), Greece (1999)
3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1** (1988) 321–331
4. Cohen, L., Cohen, I.: Finite element methods for active contour models and balloons for 2D and 3D images. IEEE Trans. on PAMI **15** (1993) 1131–1147
5. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing **7** (1998) 359–369
6. Lobregt, S., Viergever, M.: A discrete dynamic contour model. IEEE Trans. on Medical Imaging **14** (1995) 12–24
7. McInerney, T., Terzopoulos, D.: T-snakes: Topology adaptive snakes. Medical Image Analysis **4** (2000) 73–91
8. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. Fifth International Conf. on Computer Vision (ICCV'95), Cambridge, MA, June, 1995, Los Alamitos, CA, IEEE Computer Society Press (1995) 694–699
9. Mortensen, E.N., Barrett, W.A.: Interactive segmentation with intelligent scissors. Graphical Models and Image Processing **60** (1998) 349–384
10. Falcão, A., Udupa, J., Samarasekera, S., Sharma, S.: User-steered image segmentation paradigms: Live wire and live lane. Graphical Models and Image Processing **60** (1998)
11. Staib, L., Duncan, J.: Boundary finding with parametrically deformable models. IEEE Trans. on PAMI **14** (1992) 1061–1075
12. Cootes, T., Hill, A., Taylor, C., Haslam, J.: The use of active shape models for locating structures in medical images. Image and Vision Computing **12** (1994) 355–366