

SKETCH INITIALIZED SNAKES FOR RAPID, ACCURATE AND REPEATABLE INTERACTIVE MEDICAL IMAGE SEGMENTATION

T. McInerney, M.R. Akhavan Sharif

Ryerson University
Dept. of Computer Science
350 Victoria St., Toronto, ON, Canada M5B 2K3

ABSTRACT

We combine a pen and pressure-sensitive tablet input device, and a sketch-based user initialization process, with a general subdivision-curve Snake to create an intuitive, fast, accurate, interactive model-based segmentation method. Using the pen input device, the Snake is quickly and precisely initialized with a series of sketch lines such that it is extremely close to the position and shape of the target object boundary, making the Snake’s task much simpler and hence more likely to succeed in noisy images with minimal user editing. The user may also use pen pressure levels to easily impart knowledge of object edge strength to the model, and the low degree-of-freedom subdivision curve Snake provides powerful control and editing capabilities. We apply our Snake to the segmentation of several 2D medical images.

1. INTRODUCTION

Robust, fully automatic medical image segmentation systems have proved extremely challenging to develop. Consequently, semiautomatic techniques, such as Snakes [1] and their variants have become widely popular. Many improvements in the application of these models to images have been proposed, especially models that mitigate the well-known initialization sensitivity problem, creating the ability to perform “one-click” initialization and allowing some models to flow into complex shapes and change their connectivity[2, 3, 4, 5, 6]. These improvements have proved effective for a significant set of segmentation tasks, especially involving very complex-shaped objects or numerous small objects. However, these models tend to fail in noisy images and require significant user intervention. Unfortunately, the mechanisms and formulations that enabled the simple initialization and complex-shape extraction were often achieved at the expense of capabilities to perform interactive steering, high-level control and editing, and the ability to impose global shape constraints. Over the past several years, we have been exploring an alternative research direction [7, 8] to complement the “flow”-type models and create tools for noisy medical images and image sequences. In particular, our goal is to minimize the time and labor required to interactively construct and initialize a Snake, provide powerful interactive steering and editing capabilities, while retaining the ability to support high-level controllers and incorporate shape constraints.

In [9], Olabarriaga and Smeulders suggested several guiding design principles be followed to produce efficient interactive segmentation methods that generate accurate and repeatable results: integrating the computational and user interaction components into one process, using pictorial input to the computational part, initializing the segmentation method with key information which will lead the

method to an accurate result more quickly, allowing the user to control and steer the method throughout the whole process, allowing the user to intuitively predict the impact of the interactions on the segmentation result, providing visual feedback of the effect of the interaction in real-time, emphasizing computation after each user interaction for optimal repeatability, and adding intelligent behavior to elevate the abstraction level of the interaction.

In this paper, we have attempted to adhere to these guiding principles. We describe several extensions, modifications and additions to our active contour model in [8] in an effort to provide precise, intuitive initialization and interactive abilities while maintaining the capability of high-level control and constraints. We use an inexpensive, off-the-shelf pressure sensitive pen and tablet input device, rather than a mouse, to allow easy, fast, precise positioning and sketching of points and lines. We utilize the pressure sensitivity to impart knowledge of object edge strength. We have also replaced the B-spline Snake with a new Subdivision-Curve Snake that is simpler and more general. This new subdivision curve snake can be initialized with a few sketch lines to be extremely close in shape to the target object. It is naturally multi-scale, allowing coarse to fine and other custom fitting schedules. We have also improved our general sketch-line initialization process, taking advantage of the pen input device features. We demonstrate our sketch-line based Snake on several 2D medical images and briefly discuss its performance.

2. SUBDIVISION CURVE BASED DEFORMABLE MODELS

The success of shape model-based segmentation techniques is still heavily dependent upon the flexibility, controllability, and simplicity of the underlying shape representation scheme. We propose the use of subdivision curves (and surfaces in 3D) [11] as a general shape representation basis for deformable models. We have developed a Java-based, open source, highly extensible Snakes system known as JESS[10] for the rapid construction of 2D segmentation systems, which include subdivision curve Snake implementations.

2.1. Subdivision Curves

The underlying idea behind subdivision methods [11] is the use of geometric algorithms to progressively subdivide a control polygon or mesh. Repeated subdivision leads to a hierarchy of increasingly refined models which approach the *limit curve* (*surface* in 3D). One such algorithm in 2D is based upon Chaikin’s[12] “corner cutting” method, where a new control polygon is generated by cutting the corners off the original one (Figure 1). Assume the n vertices of a control polygon are $\mathbf{P}^0 = \{\bar{p}_0^0, \bar{p}_1^0, \dots, \bar{p}_{n-1}^0\}$, where the superscript denotes the level of subdivision. Chaikin’s scheme creates

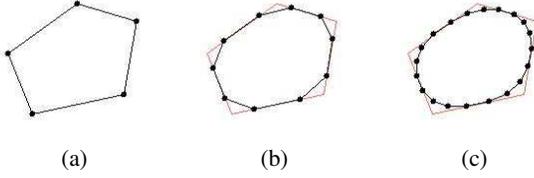


Fig. 1. Example of corner cutting subdivision to generate a subdivision curve. (a) Control polygon. (b) After one subdivision. (c) Two subdivisions.

two new vertices between each subsequent pair of vertices, say \vec{p}_i^k and \vec{p}_{i+1}^k , of the original polygon as: $\vec{p}_{2i}^{k+1} = \frac{3}{4}\vec{p}_i^k + \frac{1}{4}\vec{p}_{i+1}^k$ and $\vec{p}_{2i+1}^{k+1} = \frac{1}{4}\vec{p}_i^k + \frac{3}{4}\vec{p}_{i+1}^k$. This may be written in terms of a local subdivision operator as $\vec{p}^{k+1} = \vec{S}\vec{p}^k$.

The positions of the vertices at level $k+1$ can be expressed in matrix form as $\mathbf{P}^{k+1} = \mathbf{S}_k^{k+1}\mathbf{P}^k$. The global subdivision operator \mathbf{S}_k^{k+1} , which defines level $k+1$ in terms of level k , may be represented as a rectangular (and usually sparse) matrix. The entries of \mathbf{S}_k^{k+1} are defined by both the chosen subdivision scheme and the topology (but not geometry) of the control mesh at level k . The entries and size of the global subdivision operator change at each level.

Maillot and Stam [13] introduced a more general subdivision curve and surface. This model allows an edge to be subdivided into any number of pieces, creating the ability to tie the pixel resolution of the image more closely and naturally to the curve points of a Snake. The original control points are retained at each subdivision level, a very important feature if the control points are specified as landmark constraints. They also introduced a new parameter to allow the curve to be pushed back towards the control points, interpolating the control points if desired. This feature allows the user to initialize a Snake that is very close to the target object shape and to utilize the user-entered control points as Snake constraints. In this paper, we use Maillot’s and Stam’s algorithm for our subdivision curve Snake.

2.2. Subdivision-Curve Snakes

To construct a Snake using a subdivision curve, we use the vertices of the coarsest level control polygon as the degrees of freedom (d.o.f.), and the finest level vertices as “sensors”. Forces are computed at these sensor points and then distributed, using weights derived from the original subdivision rules, to the control points. The Snake is then constructed with the control points using either the simple explicit scheme [5] or using the well-known semi-implicit formulation in [1].

To distribute forces from sensor points at the finest level of the subdivision curve to the control points, we may use the transpose of the global subdivision operator: $(\mathbf{S}_k^{k+1})^T \mathbf{F}^{k+1} = \mathbf{F}^k$, where \mathbf{F} is the generalized external force vector. This process can also be carried out explicitly for each sensor point using the transpose of the local subdivision operator.

Subdivision-curve Snakes provide a robust, easily manipulated (using either user-derived forces or by directly editing the spline-like curve), smooth, low d.o.f. deformable model that integrates image feature information between control points. It is efficient, naturally multi-scale, and the control polygon provides a natural framework for additional constraints. To increase the accuracy of the segmentation, the user can simply use the vertices at the next level down the subdivision curve hierarchy as the d.o.f.’s. Alternatively, the user

may interactively select and subdivide a specific edge of the control polygon to generate a new control polygon, and rerun the efficient subdivision process.

3. SKETCH-BASED SNAKE INITIALIZATION

Our sketch-based initialization process [8] is simple but effective. We have extended this process to take advantage of a pressure sensitive pen and tablet. The user uses the pen input device to quickly sketch lines across the target object. 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 (Figure 2). If the object boundary exhibits strong edges near the line, the user need not sketch the line carefully - the algorithm can be set to automatically search for the strongest edge close to and perpendicular to the line. However, we have observed that the cursor can be positioned very precisely and quickly using the pen device, with little fatigue.

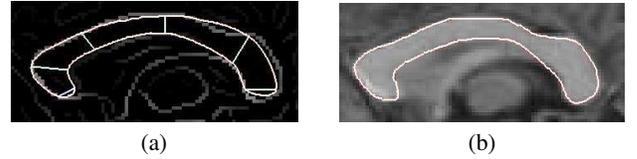


Fig. 2. Example corpus callosum (CC) segmentation using sketch lines and subdivision curve Snake. In (a) the user sketches a few lines in an edge detected image resulting in an initial curve extremely close to the shape of the CC. Note how the initial curve interpolates the line endpoints. In (b) the Snake instantly locks onto the boundary.

For regions of the object with missing or noisy edges, the user applies more pressure to the pen as the line is drawn. This indicates to the algorithm that the user is imposing a stronger constraint and should accept line endpoints as part of the object boundary (by creating a hard constraint point [7]). The user also takes more care in positioning the line, ensuring that the line endpoints are placed where the user has recognized the boundary to be. Thus, optimal use of human recognition capabilities are exploited. The user is able to easily and dynamically transfer knowledge of the object boundary to the algorithm as the Snake is constructed, maximizing the chance of Snake success and thereby minimizing or eliminating post user editing/interacting phase. This philosophy is similar to Live Wire [14], where the cursor speed is used to indicate weak and strong boundaries.

As each line is sketched, the algorithm uses it to automatically update the control polygon, and dynamically displays the resulting subdivision curve. Thus, the user is given immediate visual feedback of the Snake construction and can observe the accuracy of the initialization as it progresses. Typically the initial Snake is very close to the target object shape, ensuring the Snake will lock onto the correct boundary without editing. Nevertheless, the user may want to edit (or add to) the initial contour. This is done in two basic ways. Firstly, the user may “press” (i.e. apply a threshold pressure) on control polygon edges to “break” (subdivide) the edge, the new control point may then be dragged to its new position. The control polygon and curve are automatically updated and drawn as the control point is dragged. Secondly, if the cursor is placed on or near a control polygon edge, it becomes the active edge and is highlighted. Consequently, when a new line is sketched by the user, it is connected to

the control polygon via this active edge. These two processes allow a user to easily outline more complex shapes that may have significant protrusions or bumps (Figure 5).

Sketch lines and control points may be added during the initialization process or even while the snake is deforming, allowing the algorithm to be interactively steered. One of the main strengths of Snakes is the ability to use input-device derived forces to dynamically push or pull on a Snake (i.e. as it is deforming). The user may also place and move soft constraint “magnets” (or hard constraint “pin points”) which pull the closest snake point towards the magnet (or pin the selected closest snake control point). Unlike Live Wire, which only supports backtracking of the cursor to undo the last path segment, this type of steering may be performed at any part of the Snake. Finally, the user may bypass the Snake dynamics and edit the subdivision curve directly, using the powerful editing semantics of this spline-like representation. Editing may be performed at any level of the subdivision, allowing as local an edit as desired. All of these control mechanisms combine to provide intuitive and precise steering and editing capabilities.

4. SNAKE CUSTOMIZATIONS AND SHAPE CONSTRAINTS

As listed in [8], various Snake customizations are possible to improve robustness to noise. In fact, these customizations are now simpler to implement using the underlying subdivision curve, rather than a B-Spline, since we store an explicit representation of sensor points at every subdivision level. Due to space limitations, we will simply list these customizations: the number of sensor points per control polygon edge can be set so that the number of sensors roughly matches the image pixel resolution. The number of control points automatically added between user defined control points can be set. For each sensor point, a search along its normal is carried out for a small, user-specified distance (typically only two or three pixels). The search criteria can be set for edges with a specific magnitude and expected direction. If a matching edge pixel 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 noisy edge pixels), this pixel does not contribute to the image forces. These combined features allow a user to “tune” the Snake for a specific object and image modality and maximize segmentation performance.

One of our design goals was to not lose the ability to utilize high-level controllers or incorporate global shape constraints. To create a Snake with the properties of a deformable template model [8], global shape constraints can be incorporated by connecting control points on the control polygon using springs. With the push of a button, the control points can be connected with springs (including angle springs that maintain angles between adjacent Snake edges), and these spring constraints can be included as external forces. The control polygon acts as a spring-mass lattice, constraining the global shape or symmetry of the Snake. This feature is very useful when the model is tracking an object from slice to slice. Another mechanism to incorporate knowledge is to have the user sketch lines in a specific order for a particular object, and at specific critical locations such as landmark points. Thus, knowledge of global shape, such as width, is transferred to the Snake and the Snake is made ‘aware’ of its initial position with respect to the target object. This knowledge can be used in various ways. For example, optimal object-specific Snake parameterizations can be automatically generated (initially hand-generated) by adding d.o.f. in optimal locations between the user-specified control points. Another possibility is the construction of customized controllers that first fit segments of the Snake in areas

of known strong image feature evidence, then increasing the Snake stiffness and fitting the Snake in noisy regions.

5. EXPERIMENTAL RESULTS AND DISCUSSION

We have applied our subdivision-curve Snakes to several 2D images. In Fig. 3 we show the user sketched lines, initial snake and the final segmentation for an arm bone and a finger bone in noisy x-ray images. Some user lines in Figure 3(a) and (c) were drawn with significantly more pressure applied to the pen, resulting in the automatic creation of hard constraints (indicated by filled rectangles), forcing the end points of these lines to be fixed, with the Snake exactly passing through them. In Fig. 3(c), the active edge (the line at the bottom) was “broken” into two pieces by selecting it and then applying increased pressure with the pen. A new control point is created and this point was dragged into position. This control point was then selected and increased pressure applied to force the creation of a hard constraint. We typically switch back and forth between the edge map image and the input image to determine more easily where weak boundary areas occur. These x-ray images are very noisy, especially where two bones overlap. There are many large gaps in the edges of the bone boundary and many spurious edges inside the bone. As the initial Snakes are almost the same shape as the target bones, they are able to ignore edges that are not of a specific magnitude and direction. Occasionally a small edit is required to move a section of the final curve over slightly. We find that using the editing capabilities of the subdivision curve is sufficient to nudge the section into the correct position. In our opinion the editing abilities of the Snake and its subdivision curve representation provide much more flexible control than the simple backtracking provided by Live Wire.

Figures 4 and 5 show two more examples. In both of these examples, the user has sketched lines and then broken an edge (near the right bulge of the knee bone and at the bottom of the lung) in order to create an initial Snake almost the exact shape as the target objects. We have performed some preliminary experiments to compare our Snake to a Live Wire-like algorithm. In general, for the objects segmented in this paper, we have found that our system requires approximately *half* the number of user interactions to achieve an accurate segmentation than the Live Wire-like tracing algorithm we used. In addition, we have chosen five corpus callosum images (as in Figure 2) at random and repeated the segmentation five times for each image. Our Snake generated essentially (an average error of less than 1 pixel) the same result for each of the CC’s. Occasionally, a small nudge of a Snake curve section near the fornix was required to correct the segmentation.

6. CONCLUSION

Optimizing the performance of semi-automatic model-based segmentation methods for noisy images (and image sequences) requires fast, precise and fatigue-free initialization, powerful user control and editing capabilities, and the ability to exploit the image interpretation abilities of the human expert. The combination of sketching input lines across an object, a pressure sensitive pen and tablet input device, and a general subdivision-curve Snake results in a tool that meets these requirements, and is effective for many segmentation tasks that cannot be efficiently processed with other techniques. Future work involves packaging our entire suite of Snakes tools within the JESS framework to create a complete open-source segmentation package that can be applied to virtually any 2D medical image and object.

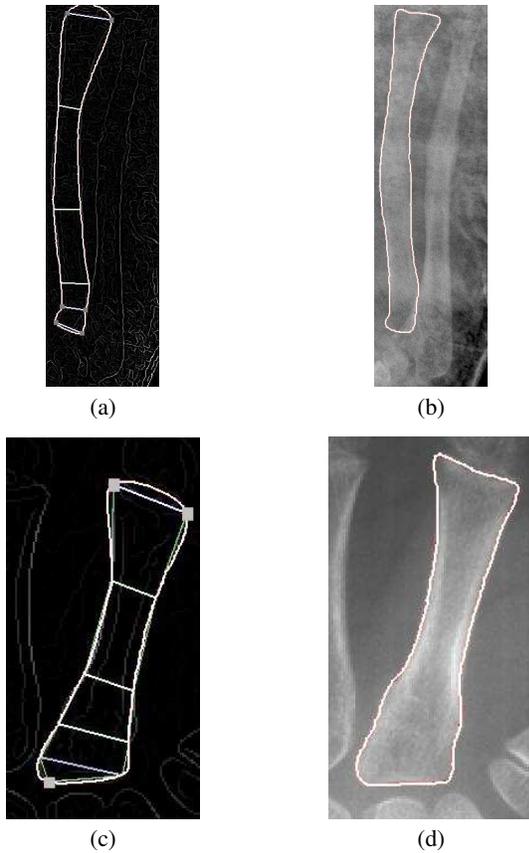


Fig. 3. Segmenting an arm bone from a noisy x-ray image. (a) user input lines and initial snake (b) segmentation result. (c)(d) segmenting finger bone from hand x-ray image. The rectangles indicate hard constraint points generated automatically from increased user pressure on the pen during line drawing.

7. REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[2] L.D. Cohen and I. Cohen, "Finite element methods for active contour models and balloons for 2D and 3D images," *IEEE Trans. on PAMI*, vol. 15, no. 11, pp. 1131–1147, November 1993.

[3] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[4] S. Lobregt and M.A. Viergever, "A discrete dynamic contour model," *IEEE Trans. on Medical Imaging*, vol. 14, no. 1, pp. 12–24, March 1995.

[5] Tim McInerney and Demetri Terzopoulos, "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, pp. 73–91, 2000.

[6] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *IJCV*, vol. 22, no. 1, 1997.

[7] J. Liang, T. McInerney, and D. Terzopoulos, "United snakes," *Medical Image Analysis*, In press.

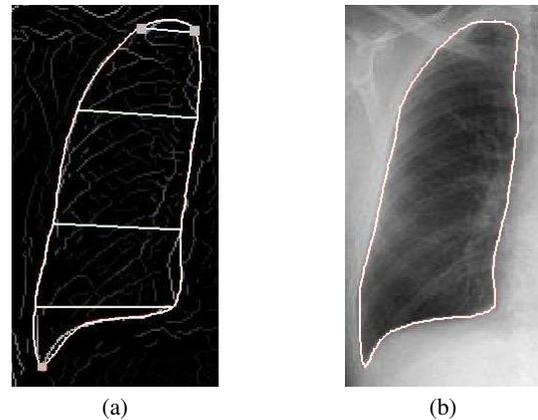


Fig. 4. Segmenting a lung from a noisy x-ray image. (a) user input lines, hard constraint points and initial snake (b) segmentation result.

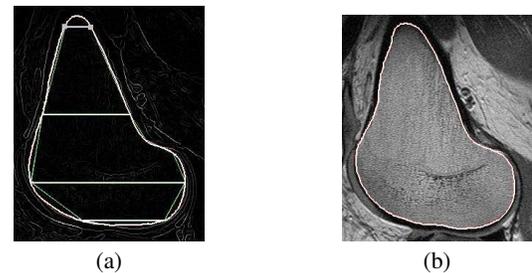


Fig. 5. Segmentation of knee bone from MR image. (a) control polygon generated from 4 input lines and two additional points added by breaking control polygon edge near the right hand bulge. (b) segmentation result.

[8] T. McInerney and H. Dehmshki, "User-defined B-spline template-snakes," in *Proc. Sixth International Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI 99)*, Montreal, Canada, November 2003, Springer.

[9] S.D. Olabbarriaga and A.W.M. Smeulders, "Interaction in the segmentation of medical images: A survey," *Medical Image Analysis*, vol. 5, pp. 127–142, 2001.

[10] T. McInerney, M.R. Akhavan Sharif, and N. Pashotanzadeh, "Java extensible snakes system," in *SPIE Int. Symposium, Medical Imaging*, San Diego, CA, February 2005.

[11] D. Zorin, P. Schröder, A. DeRose, L. Kobbelt, A. Levin, and W. Sweldens, "Subdivision for modeling and animation," in *SIGGRAPH 2000 Course Notes*. 2000.

[12] G. Chaikin, "An algorithm for high speed curve generation," *Computer Graphics and Image Processing*, vol. 3, pp. 346–349, 1974.

[13] J. Maillot and J. Stam, "A unified subdivision scheme for polygonal modeling," in *Proceedings of Eurographics*. 2001, vol. 20, Computer Graphics Forum.

[14] A.X. Falcão, J.K. Udupa, S. Samarasekera, and S. Sharma, "User-steered image segmentation paradigms: Live wire and live lane," *Graphical Models and Image Processing*, vol. 60, 1998.