SketchSurfaces: Sketch-Line Initialized Deformable Surfaces for Efficient and Controllable Interactive 3D Medical Image Segmentation

Meisam Aliroteh and Tim McInerney

Departments of Computer Science and Electrical Engineering, Ryerson University, Toronto, ON, Canada, M5B 2K3

Abstract. We present an intuitive, fast and accurate interactive segmentation method for visualizing and analyzing 3D medical images. Our method combines a general deformable subdivision-surface model with a novel sketch-line user initialization process. The model is simply and precisely initialized with a few quick sketch lines drawn across the width of the target object on several key slices of the volume image. The smooth surface constructed using these lines is extremely close to the shape of the object boundary, making the model's task of snapping to this boundary much simpler and hence more likely to succeed in noisy images with minimal user editing. Our subdivision surface based deformable model provides a foundation for precise user steering/editing capabilities and all of the simple, intuitive user interactions are seamlessly integrated with advanced visualization capabilities. We use our model to segment objects from several 3D medical images to demonstrate its efficiency and accuracy.

1 Introduction

Segmentation of 3D and 4D images remains one of the major challenges in medical image visualization and analysis. Currently, semiautomatic methods are potentially the most effective approach since they combine the efficiency aspects of automatic segmentation with human expert guaranteed robustness. However, despite the large number of 3D semiautomatic segmentation methods developed over the past decade, no one technique has yet been universally adopted in clinical practice; as a result, manual delineation or simple pixel-based tools are still heavily used. It is our contention that the lack of adoption of these techniques is in part related to the incomplete integration of simple, intuitive, and consistent interaction capabilities, with the necessary precision and power, into the segmentation work-flow.

Semi-automatic 3D segmentation techniques can fail in noisy images and this failure often results in time-consuming and tedious user intervention. With many of the current techniques [1], [2], [3], [5], [6], once the algorithm is initiated the ability to steer it is limited and/or the ability to edit it is restricted to a separate post processing phase, often with a separate tool-set and/or user actions. That is, while the mathematical formulations and numerical algorithms of these methods enable simple initialization and complex shape extraction, the cost may be incurred on the "back-end" of the segmentation process. Other techniques [7], [8] do provide interactive steering capabilities but editing facilities are not well integrated, the required tracing actions can be tedious, and the 2D nature of the underlying algorithm often requires considerable user concentration for a 3D data set.

The deficiencies of user control functionality, simplicity of use, and the lack of interaction precision and intuitiveness of current methods have led us to explore an alternative research direction in 2D [9], creating interactive active contour models suitable for segmenting medical images which exhibit a significant amount of noise. In this paper, we extend this idea to 3D segmentation. More specifically, our goals are to provide a highly accurate initialization of an active surface model using fast, simple, user actions that require little concentration. After initialization, it should be visually apparent to the user what the resulting segmentation will look like. In addition, we seek a robust, noise-insensitive model that provides the foundation for precise, efficient, and intuitive steering and editing capabilities, and a method that requires little or no parameter tweaking and/or mode changes. Most importantly, all the interaction capabilities must be seamlessly "woven" together providing simple, consistent user interactions and complete user control throughout the entire segmentation process.

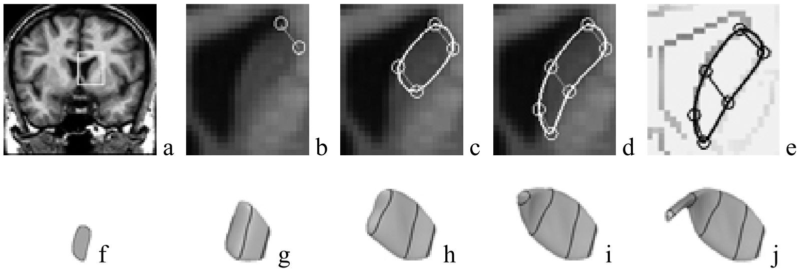

Fig. 1. SketchSurfaces segmentation of the right caudate nucleus from an MR volume image. The user positions and orients an image slice plane (a), and then sketches a few lines across the width of the caudate, creating a contour that approximates the caudate boundary (b-d). The user may also use an edge detected view (e). The user then proceeds to another slice plane and repeats this sketching process. After each sketch process, the previous contours are automatically connected and converted into a subdivision surface (f – j). The dark curves in f – j highlight the sketch plane locations.

To achieve our goals, we present *SketchSurfaces* - a highly controllable active surface model with a simple formulation that is coupled with a novel sketch-based 3D initialization method (Fig. 1). SketchSurfaces offer easy, consistent transitions between initializing, fitting, steering, editing, visualizing, and zooming. The user is able to stop the segmentation at any time, examine the current initial surface model or partial segmentation result, zoom in and out, and make precise corrections before continuing, all with simple mouse or stylus actions. In the remainder of this paper, we describe SketchSurfaces and its interaction model in more detail. We apply it to several MRI brain data sets to demonstrate its effectiveness and make a brief comparison of its efficiency and accuracy with other techniques.

1.1 Background

Over the past decade or more, several model-based 3D interactive segmentation techniques have been developed. For many of the methods, the research focus was on the model itself, the user was required to specify various un-intuitive parameters, and the interaction facilities were added later.

For example, one popular deformable surface model technique is commonly known as a deformable "balloon" model [1], [2], [3], [5], [6]. These models can "inflate" and take on the shape of the target object and some are also able to dynamically change their topology [1], [2], [5]. The inflation forces significantly increase their shape capture range and they can be simply initialized using, for example, one mouse click to create a small spherical seed model inside the target object. Balloon models work best when the image feature map is relatively clean and homogeneous. However, clinical images are noisy, contain many uninteresting edges and regions of low contrast, contain gaps in the object boundary, or exhibit a complex texture. Hence, these more automatic techniques may not generate the expected result - the added automation does not come without a cost. Some interactive control (steering capability) over the model is lost and the gaps in the object boundaries may allow the model to leak through, requiring user intervention in the form of barriers. Another common problem is the balloon may not completely inflate into all regions of the target object and mechanisms must be provided so that the user can force the model into these regions. Finally, editing in these techniques is typically performed in a post processing phase, often using a separate tool-set. More recently, researchers have developed semiautomatic volume/region painting methods [10], [11], [12], [13]. Here the user gives loose hints to the algorithm by painting rough strokes or seeds on the object and several strokes on the background. An optimization algorithm uses these inputs hints to extract the actual object boundary. In noisy images, these “Graph Cut” techniques can share similar problems with the balloon models and the user needs to provide more foreground/background strokes or edit with a separate tool-set. On the other hand, both the balloon and Graph Cut methods may be used to segment very complex-shaped objects, such as the skull or the brain cortical surface. SketchSurfaces would currently require too much user input to efficiently extract such structures.

Another popular technique is the extension of LiveWire to 3D [7], [8]. "LiveWire" or "Intelligent Scissors" [14], [15], is an interactive 2D boundary tracing tool which provides the user with the ability to steer the segmentation. Although faster and more reproducible than manual tracing, these techniques can still demand a large amount of concentration from the user, especially in noisy 3D images. There is never a perfect match between the features used by the LiveWire algorithm and the desired object boundary. As a result, the user often must control the mouse carefully. If a mistake is made, the user’s only option is to "backtrack" and try again. In addition, tracing around the entire object using a mouse can be fatiguing. Finally, 3D LiveWire is fundamentally an extended 2D technique – when segmenting a 3D image, the 2D contour-based nature of the algorithm forces the user to pay careful attention to how it is applied and to mentally reconstruct the 3D shape of the object.

2 SketchSurfaces

Our method uses an active surface model and is 3D by nature. It is constructed using a subdivision surface [16], providing the robustness against noise of a finite-element based model, the broadest possible shape coverage, a foundation for simple and precise editing, inherent smoothness, and a natural hierarchical organization.

We have carefully integrated a novel sketch line initialization process that allows the subdivision surface to be quickly initialized with a few rough sketch lines drawn across the width of the target object in several key image slices. The result is an initial model that is extremely close in shape to the target object, making the deformable surface model's task of snapping to the object boundary much simpler and hence more likely to succeed in noisy images with minimal user editing. It also minimizes the effect of the model's elasticity parameters. Furthermore, it allows the model to primarily rely on image edge information and (optionally) expected edge transitions (e.g. bright to dark) so that it is generally applicable to many common imaging modalities without the need for modality-specific parameters. Finally, no tedious tracing actions are needed - only simple short sketch lines on a few slices are required. Steering and editing the model can take place at any time during the segmentation process and anywhere on the model. In other words, we have attempted to design the method with the 3D user interaction model as our focus. In the following sections, we describe the construction of our subdivision surface model as well as our interaction model.

2.1 Subdivision Surfaces

The success of interactive shape-model based segmentation techniques, such as active contours or active surfaces, is still heavily dependent upon the generality, controllability, and simplicity of the underlying shape representation scheme. We propose the use of subdivision surfaces (and curves in 2D) [16] as a very general shape representation basis for deformable models. The underlying idea behind subdivision methods [16] is very simple - using geometric algorithms to progressively subdivide a control polygon (or control mesh in 3D). Repeated subdivision leads to a hierarchy of increasingly refined models which approach a smooth *limit curve* (*limit surface* in 3D). The vertices of the control mesh (the control points) provide a convenient, intuitive, and precise mechanism to control the shape of the limit surface. As the control points are repositioned, a new limit surface is computed in real time, allowing the user to precisely deform the surface (Fig. 2).

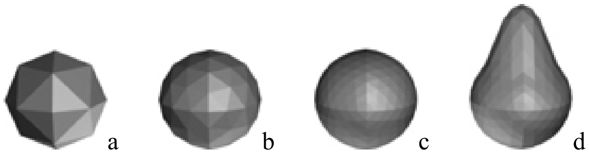

Fig. 2. Example of a simple subdivision surface and the subdivision process: (a) control polygon, (b) one level of subdivision, (c) two levels, and (d) modified subdivision surface after the user moves a control point.

2.2 Constructing a Deformable Subdivision-Surface Model

To construct a deformable surface model using a subdivision surface, we use the vertices of the coarsest level of the surface (i.e. the control points) as the degrees of freedom (d.o.f.), and the finest level vertices as "sensors". The idea is to use a sufficient number of control points and a sufficient level of subdivision such that there is roughly one sensor point for each boundary voxel of the target object, making maximal use of all available image information. External image forces are computed at these sensor points and then (using the weights of the subdivision mask of the points) distributed among the control points (i.e. a control point that is closer to a particular sensor point will receive proportionately more of the computed force than a control point further away). The deformable subdivision surface model is then formulated numerically using the simple explicit scheme described in detail in [5] and the positions of the control points are updated. New external forces are calculated and the process is repeated until an accurate solution is reached.

Our subdivision surface shares similarities with the model developed in [4]. However, their model was developed as a standard deformable balloon and uses a complex finite-element formulation. We use a simple finite difference formulation (although with a newer, more stable numerical update procedure). In addition, rather than the traditional image gradient magnitude external forces, we use spring forces between a sensor point and an edge voxel as our external forces. For each sensor point on the model, a search along its normal is carried out for a small, user-specified distance (typically only two or three voxels). If a matching edge voxel is found, a spring force is applied to attract the sensor point to it. If no matching edge is found (due to boundary gaps for example), this sensor point does not contribute to the image forces. This type of external force produces very predictable and stable behavior, with no "leaking" issues commonly associated with balloon inflation forces.

3 SketchSurfaces Interaction

SketchSurfaces presents two windows to the user. One window is used as a contextual view and consists of a 3D image slice plane that can be oriented as the user chooses. The user can intuitively push or pull on this plane to navigate through the volume and the display of the 3D deformable surface model is controlled with a key. The other window presents the same slice plane in 2D (i.e. looking down at the plane in the negative direction of its normal) and allows the user to focus on a slice of the target object with the cross-section of the surface model overlaid. Without this 2D view, the user will encounter situations where they would have to tediously rotate the entire scene in the 3D window to bring the content of the 3D slice plane into view. The user can initialize, fit, edit, steer, or zoom in on the model in either window.

3.1 Initialization

A critical and often overlooked phase of the interactive shape-model based segmentation process is the initialization of the model. Explicit, parametric deformable models such as SketchSurfaces typically minimize energy using local optimization techniques. Consequently, efficient convergence to the desired segmentation result is heavily dependent upon careful initial model placement.

As discussed in Section 1.1, the most common method of initializing a deformable surface model is to create a small spherical model from a user-defined seed point. Inflation forces are then used to drive the model towards the object boundary. The problem with this approach is the model needs to inflate a considerable distance to reach the object boundary so it is not immediately visually apparent whether the segmentation will succeed. This approach creates model steering issues as well as a separation between the inflation phase of the segmentation and the editing phase.

User fatigue is an important consideration in any interactive design and analysis task. For this reason, sketching actions are being actively researched for many of these tasks in an effort to reduce fatigue and user concentration and to "amplify" user input action, thereby maximizing productivity. In the context of interactive segmentation, the idea behind sketching is to allow the user to provide an accurate initialization for the surface model - which minimizes subsequent steering and editing - with a low degree of concentration and simple mouse movements. Our sketch-line initialization process [9] is a simple but effective technique that realizes this idea (Fig. 3). It does not require tedious tracing actions and can be performed with minimal concentration. Short sketch lines are quick, easy and comfortable to draw with a mouse or stylus.

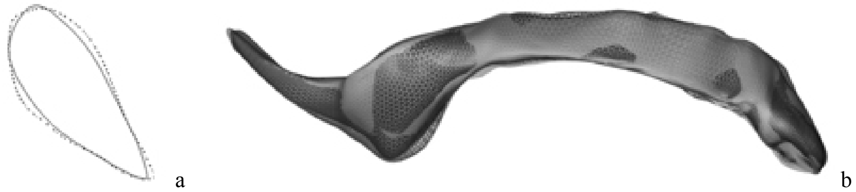

Fig. 3. Result of the sketch line initialization process for the right brain ventricle. By quickly sketching a few lines in several slices of an MR volume image, a highly-accurate initial surface is created. (a) cross-sectional contour of the initial model (solid curve) shown with a cross-section of the manually segmented ventricle (dotted curve). (b) initial surface model (light semi-transparent) shown with the manually segmented ventricle surface (dark wireframe).

The user begins the initialization process by first positioning the image slice plane at the far end of the target object and then sketching a few short lines across the width of the object (Fig. 1, b-d) on this plane (which we term a *sketch plane*). Drawing these lines roughly perpendicular to the object boundary will result in an initial surface model that is approximately locally aligned with the boundary. The user then pulls the image slice plane forwards and again sketches a few lines across the object. A subdivision surface model is progressively constructed in real-time from these sketch lines using the quick-hull algorithm and the subdivision process (Fig. 1, f – j), and the cross-section of the model is instantly displayed on the image slice. In this way, the user is given immediate visual feedback of the surface model construction and can observe the initialization accuracy. The user repeats this pulling and sketching process several times until a sketch plane is close to the near-end of the object.

The user can view the 3D surface model as it is constructed and make corrections if desired by repositioning control points and/or by adding new sketch lines. Additional sketch planes can also be inserted. The number of sketch planes required in order to generate an accurate initial surface model is dependent upon the shape of the target object. For example, in our experiments with the Caudate Nucleus, we typically used five sketch planes, with one close to each end of the Caudate Nucleus and three evenly distributed in between. Furthermore, the number of sketch lines required on each sketch plane is dependent on the complexity of the shape of target object cross-section (usually only two or three sketch lines are required for the Caudate Nucleus).

The user typically performs a few trials, with a different number and position of sketch planes, until an accurate initial surface model is obtained. We have found that it is best to begin with two/three sketch planes (one close to the each end of the target object and optionally, one in the center), examine the initial surface, fit the model and examine the segmentation result, and then add intermediate sketch planes until the desired accuracy is obtained. This pattern of sketch planes and sketch lines can then be recorded and redisplayed when segmenting the target object in all subsequent volumes. Note that the pattern of sketch planes is only a rough guide – the algorithm is fairly robust to small differences in the number/positions of the sketch planes. For large differences accuracy is impacted, resulting in increased user editing.

3.2 Fitting, Steering, and Editing

Editing a segmentation result typically implies making corrections to the segmentation after the algorithm has run to completion. Steering, on the other hand, implies guiding the segmentation process toward the correct result while the algorithm is running. The lines between fitting, steering, and editing is blurred when considering the segmentation work-flow using SketchSurfaces. Once the model has been initialized, the user presses a button to begin the fitting step and the surface quickly *snaps* to the object boundary (typically within one to two seconds). The fitting/snapping of the model to the object boundary is done in discrete steps rather than in a continuously deforming manner, allowing steering/editing to be performed at any time. We terminate the fitting step after a (user-adjustable) number of iterations have occurred. Because the initialization process creates a surface model very close to its final position, one fitting step is often sufficient to generate an accurate segmentation. Nevertheless, the user may repeat the fitting step as many times as desired.

Before, between, or after these fitting steps, SketchSurfaces can be precisely and intuitively controlled using various geometric editing actions. The initialization, fitting, editing, steering, and zooming are all performed within a single process, using only simple actions. The user is not forced to switch modes or select action items from a menu/panel. Examples of these SketchSurfaces editing actions are listed below:

1. *Before fitting*: The user can edit the initial surface by modifying the sketch plane contours. Actions include adding new sketch planes, repositioning the existing control points and sketch lines, and adding new control points by drawing new sketch lines or by breaking an existing sketch line into two pieces.
2. *After/between fitting steps*: Simple, precise control over the surface model position and shape is performed in 2D on an (2D/3D) image slice plane. Constraining user interactions to 2D significantly simplifies them, improves accuracy and efficiency, maintains user familiarity, and does not require a special input device.

   The user manipulates the image slice plane in either the 3D or 2D window and cuts the surface model in any desired orientation to generate a 2D subdivision curve. The control points of this curve are automatically created from the cross-section of the surface model's control mesh and the resulting curve shape closely matches the shape of the subdivision surface cross-section. Because the initialization and fitting of the surface model provides an accurate segmentation, the control points of the subdivision curve are often in a "good" position and fine-tuning can be performed simply by nudging these points, causing a section of the curve to be dragged into the desired position. We then construct “soft” constraint forces between the sensor points on the surface cross-section and corresponding points along the newly deformed subdivision curve. A quick refitting (snapping) of the surface model is performed to pull it towards the subdivision curve via the soft constraints. This process allows any number of arbitrarily oriented constraint forces to be added to the surface model and also takes advantage of the powerful editing capabilities of subdivision curves, avoiding imprecise and tedious editing using manual tracing. Curve control points can be fluidly dragged and precisely positioned and control points can be added or deleted by clicking on the curve etc. One potential drawback of this constraint mechanism is large changes in the curve may prevent the surface model from completely snapping to the curve points.
3. *Snipping*: Occasionally the initialization process may create a surface region that protrudes past the end of an object. We provide the ability to snip away this protrusion by simply positioning the 3D image slice plane such that the protrusion is on one side of the plane. The user simply presses a key to cutout this excess portion.
4. *Pinning/Unpinning*: The user can pin/unpin any control points of the surface model by simply double-clicking on a point or an edge joining two control points. A pinned control point will remain frozen regardless of any forces that are acting on it. This is a useful feature for segmenting objects in regions with no (or very weak) edges and is typically performed during sketch line initialization. The user takes more care to position and pin a point in its "final" position.

5. *Local subdivision*: New control points and sensor points can be added to the subdivision surface model in a *local* region where editing is occurring. Currently, this is explicitly initiated by the user. In future, this will be performed automatically.

3.3 SketchSurfaces Parameters

The goal of SketchSurfaces is the creation of a segmentation tool that requires no parameter tweaking or mode changing. Our initialization process and subdivision surface model goes a long way towards achieving this goal - default parameter settings are often adequate. Nevertheless, the user is able to set several parameters:

1. *Edge intensity control*: Maximum and minimum edge intensity threshold can be set and the model will search for edge intensities within this range. Canny edges are used as they are visually simpler to "see".
2. *Automatic insertion of internal control points*: The number of control points automatically added between the user-defined control points can be controlled. The default value is set to one and we have used this value in all of our experiments.
3. *Image edge search range*: Edge voxel search range can be controlled in both the positive and negative sensor point normal direction.
4. *Edge transition control*: This parameter can be used to activate a search for edges with either a bright-to-dark transition or a dark-to-bright transition, allowing the model to ignore edges with incorrect transitions thereby improving robustness.
5. *Surface model stiffness/stretchiness*: Due to the proximity of the Snake to the object boundary after initialization and the restricted search range of the external force, the default settings for these parameters are typically adequate.

4 Experimental Results and Discussion

We have successfully tested our approach on five MR brain data sets obtained from The Internet Brain Segmentation Repository (http://www.cma.mgh.harvard.edu/ibsr/). The data sets were interpolated to obtain cubical voxels, with interpolated dimensions of 256 x 256 x 205 and 256 x 256 x 192. We have tested our application on a PC with a CPU clock speed of 3.2 Ghz. We segmented three different objects and used the hand-segmented volumes available from the repository to calculate the accuracy of our model. The accuracy was measured using 2-sided Hausdorff distancing, which is a more strict measure of accuracy than measures such as volume overlap.

Table 1 and Figure 4 illustrate the results obtained from segmenting the right Caudate Nucleus from the five data sets. The same parameters have been used for all of the segmentations and the same number of sketch planes was used to construct the initial surface model. Table 1 also demonstrates a significant increase in the speed of the segmentation when compared to [1] and [8]. On average, we performed the initialization and fitting in about 36 seconds plus an additional 29 seconds to edit a few slices, resulting in a total segmentation time of about 65 seconds and sub-voxel accuracy. This can be roughly compared to 2.5 minutes for a well-known balloon method [1] and 3.5 minutes for a 3D LiveWire technique [8].

Table 1. Results of segmenting the right Caudate Nucleus from five data sets.

					Hausdorff dist. stat. between the extracted and expert seg. surfaces		
Exp	# of Sketch Planes	# of Control / Sensor Points	Seg. Time (sec)	Editing Time (sec)	Avg. Dist.	Max Dist.	# of (% of) sensor points with sub-voxel accuracy
1	5	79 / 1219	33	20	0.308	1.81	1172 (96.1%)
2	5	87 / 1343	35	42	0.481	2.25	1224 (91.1%)
3	5	92 / 1412	37	24	0.444	2.11	1294 (91.6%)
4	5	87 / 1347	36	36	0.422	1.93	1257 (93.3%)
5	5	84 / 1284	37	24	0.373	1.36	1243 (96.8%)

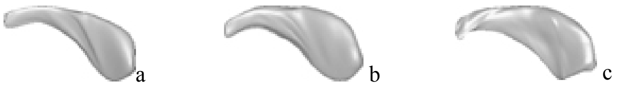

Fig. 4. Example segmentation result for the right Caudate Nucleus from MR volume image. (a,b) initial and fitted subdivision surface and (c) hand-segmented surface.

Table 2 and Figure 5 illustrate the results obtained from segmenting the right ventricle from the five data sets. Once again, the same parameters have been used for all of the segmentations. Moreover, with the exception of experiment 4, approximately the same numbers of sketch planes and control points have been used. Due to a significant difference in the shape of the ventricle in the data set for experiment 4, an extra sketch plane was inserted. On average, we performed the initialization and fitting in 41 seconds plus an additional 9 seconds to edit one slice, resulting in a total segmentation time of about 50 seconds with sub-voxel accuracy. Again, this result can be roughly compared to 7.5 mins. for a balloon [1] and 5.5 mins. for 3D Live-Wire [8].

Table 2. Results of segmenting the right ventricle from five data sets.

					Hausdorff dist. stat. between the extracted and expert seg. surfaces		
Exp	# of Sketch Planes	# of Control / Sensor Points	Seg. Time (sec)	Editing Time (sec)	Avg. Dist.	Max Dist.	# of (% of) sensor points with sub-voxel accuracy
1	7	132 / 2052	40	14	0.460	2.07	1905 (92.8%)
2	8	153 / 2373	39	8	0.393	1.83	2290 (96.5%)
3	8	150 / 2310	41	7	0.412	2.43	2160 (93.5%)
4	9	169 / 2629	43	12	0.336	2.15	2500 (95.1%)
5	8	153 / 2373	42	5	0.357	1.62	2297 (96.8%)

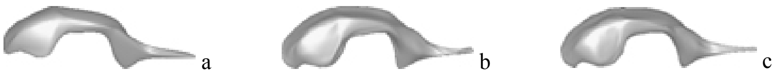

Fig. 5. Example segmentation result for the right brain ventricle. (a, b) initial and fitted subdivision surface model and (c) the hand-segmented ventricle surface.

Finally, we also segmented the right putamen from one volume image. The shape of the putamen is more complex than that of the caudate, requiring 6 sketch planes instead of 5 to construct the initial surface model and one or two more sketch lines per sketch plane were required to create accurate contours. In addition, the putamen is a very noisy object and there are large regions where there are no edge voxels and where the user must interpret the location of the boundary. These factors demand precise segmentation control. With SketchSurfaces, the initialization and fitting were performed in 65 seconds plus an additional 8 seconds to edit one slice, resulting in a total segmentation time of about 73 seconds with sub-voxel accuracy (avg. distance 0.484 voxels, max. dist. 2.07 voxels, 90.3% of sensor points with sub-voxel accuracy).

From our experiments, we observed that steering our model (specifically, editing before fitting) can be less time consuming than editing after the fitting step(s). For example, on several sketch planes there are regions of the caudate nucleus without any visible object boundaries in the edge-detected image. In these regions, the sketch lines were drawn more carefully and several control points were pinned. Due to the nature of the initialization technique and the flow of our segmentation method, this pinning action requires only a very slight increase in effort but realizes considerable gains in overall efficiency.

6 Conclusion

Optimizing the performance of 3D semiautomatic medical image segmentation methods for noisy volume images requires a minimal number of fast, simple and fatigue-free interactions, intuitive, flexible, and precise user steering and editing capabilities. Furthermore, all of these capabilities must be seamlessly integrated into the segmentation work-flow so that they are available at any time and presented to the user using a consistent interface. We believe the combination of sketching input lines across an object, a powerful subdivision-surface based deformable model, with subdivision curve editing flexibly derived from arbitrarily oriented cross-sections of this model, results in a tool that meets these requirements and is effective for many segmentation tasks that may not be processed as efficiently with other techniques. Currently our method is optimized for simple to moderately complex-shaped objects. We are extending the method so that objects with significant protrusions and/ holes may be segmented. For very complicated objects (e.g. brain cortical surface) our method may require too much user input and we are exploring the idea of merging SketchSurfaces with a balloon model. We are also investigating techniques to automate sketch plane placement, allowing even more efficient and simple volume image segmentation.

References

1. Yushkevich, P., Piven, J., Cody Hazlett, H., Gimpel Smith, R., Ho, S., Gee, J., Gerig, G., "User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability," NeuroImage, Vol. 31, No. 3, 2006, pp. 1116–1128.
2. Pons, J.P., Boissonnat, J.D., "Delaunay deformable models: Topology-adaptive meshes based on the restricted Delaunay triangulation," In IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA, June 2007.
3. Montagnat, J., Delingette, H., "Spatial and temporal shape constrained deformable surfaces for 3D and 4D medical image segmentation," Technical report RR-4078, INRIA, 2000.
4. Mandal, C., Qin, H., Vemuri, B.C., "Dynamic Modeling of Butterfly Subdivision Surfaces," IEEE Trans. on Visualization and Computer Graphics, Vol. 6, No. 3, 2000, pp. 265-287.
5. McInerney, T., Terzopoulos, D., "Topology Adaptive Deformable Surfaces for Medical Image Volume Segmentation", IEEE Trans. on Med. Imaging, 18 (10), 1999, pp. 840-850.
6. Bredno, J., Lehmann, T.M., Spitzer, K., "A General Discrete Contour Model in Two, Three, and Four Dimensions for Topology-Adaptive Multichannel Segmentation," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, 2003, pp. 550-563.
7. Falcão, A.X., Udupa, J.K., "A 3D Generalization of User-Steered Live Wire Segmentation," Medical Image Analysis, Vol. 4, 2000, pp. 389-402.
8. Poon, K., Hamarneh, G., Abugharbieh, R., "Segmentation of Complex Objects with Non-Spherical Topologies from Volumetric Medical Images using 3D Livewire," In SPIE Medical Imaging, Vol. 6512-31, 2007, pp 1-10.
9. McInerney, T., Akhavan Sharif, M.R., "Sketch Initialized Snakes for Rapid, Accurate, and Repeatable Interactive Medical Image Segmentation", In 3rd Int. Symposium on Biomedical Imaging, Arlington, Virginia, April 2006, pp. 398-401.
10. Boykov, Y.Y., Funka-Lea, G., "Graph Cuts and Efficient N-D Image Segmentation," International Journal of Computer Vision, Vol. 70, No. 2, 2006, pp. 109-131.
11. Tzeng, F.Y., Lum, E.B., Ma, K.L., "An Intelligent System Approach to Higher-Dimensional Classification of Volume Data," IEEE Trans. On Visualization and Computer Graphics, Vol. 11, No. 3, 2005, pp. 273-284.
12. Falcão, A.X., Bergo, F.P.G., "Interactive Volume Segmentation with Differential Image Foresting Transforms," IEEE Trans. on Medical Imaging, 23(9), 2004, pp. 1100-1108.
13. Owada, S., Nielsen, F., Igarashi, T., "Volume Catcher", Proceedings of the 2005 symposium on Interactive 3D graphics and games, 2005, pp. 111-116.
14. Mortensen, E.N., Barrett, W.A., "Interactive segmentation with intelligent scissors", Graphical Models and Image Processing, Vol. 60, No. 5, 1998, pp. 349–384.
15. Falcão, A.X., Udupa, J.K., Samarasekera, S., Sharma, S., Hirsch, B.E., Lotufo, R. de A., "User-steered image segmentation paradigms: Live wire and live lane," Graphical Models and Image Processing, Vol. 60, 1998, pp. 233–260.
16. Zorin, D., Schröder, P., Sweldens, W., "Interpolating Subdivision for Meshes with Arbitrary Topology," Computer Graphics Proceedings (SIGGRAPH 96), pp. 189-192.