

# Interactive Medical Image Segmentation with United Snakes

Jianming Liang<sup>1,2</sup>, Tim McInerney<sup>2,3</sup>, and Demetri Terzopoulos<sup>2</sup>

<sup>1</sup> Turku Centre for Computer Science

DataCity, Lemminkäisenkatu 14 A, 20520 Turku, Finland

<sup>2</sup> Department of Computer Science

University of Toronto, 6 King's College Road, Toronto, ON M5S 3H5, Canada

<sup>3</sup> Department of Math, Physics and Computer Science

Ryerson Polytechnic University, Toronto, ON M5B 2K3, Canada

**Abstract.** Snakes have become a standard image analysis technique with several variants now in common use. We have developed a software package called “United Snakes”. It unifies the most important snake variants, including finite difference, B-spline, and Hermite polynomial snakes, within the framework of a general finite element formulation with a choice of shape functions. Furthermore, we have incorporated into united snakes a recently proposed snake-like technique known as “livewire”, via a method for imposing hard constraints on snakes. Here, we demonstrate that the combination of techniques in united snakes yields generality, accuracy, ease of use, and robustness in several medical image analysis applications, including the segmentation of neuronal dendrites in EM images, dynamic chest image analysis, and the quantification of growth plates.

## 1 Introduction

Snakes (active contour models) [8] have met the challenge of extracting clinically useful information from medical images in numerous applications to a wide range of medical image analysis tasks, including segmentation, registration, tracking, and shape analysis [14,18]. Extensive research activity worldwide over the past decade has resulted in a host of snake variants (e.g., finite element snakes [5], B-snakes [15,3], Fourier snakes [19]) as well as snake-like “livewire” techniques [16,2,17,6]. The result is a confusing array of choices for the user.

There exists a need for definitive software which unites the best features of the various snake(-like) techniques in a portable, reusable package. To this end, we have developed *United Snakes* [11] in Java as a JavaBean (reusable Java software component). United snakes unify the most important snakes variants, including finite difference, B-spline, and Hermite polynomial snakes, in a comprehensive finite element framework, where any particular type of snake can be chosen by simply changing the finite element shape functions. Furthermore, united snakes combine snakes and livewire via a simple yet effective method for imposing constraints.

Hence, we offer a general purpose tool for interactive medical image analysis that provides more flexible control than its component techniques alone while reducing the need for user interaction. In the remainder of the paper, we will describe united snakes and demonstrate their generality, accuracy, ease of use, and robustness using clinical images from several different medical image analysis projects.

## 2 United Snakes

This section presents a non-mathematical description of united snakes. The relevant mathematical formulations are found in [11]).

### 2.1 Snakes

A snake is a dynamical system taking the form of a parametric contour in the image plane, whose shape is dictated by a potential energy comprising a contour deformation term and an image term. The final shape of the snake results when the dynamical system reaches equilibrium, corresponding to a minimum of the potential energy. Traditionally, a snake is initialized manually by the user or automatically through image processing operators with a rough approximation to an image feature of interest (edge, boundary, etc.). If the snake has been initialized appropriately, it will accurately localize and conform to the image feature. With a dynamic snake it is natural to incorporate user guidance via constraints in order to drive the snake out of one local energy minimum into another, and it is also possible to track moving objects in image sequences.

### 2.2 Finite Element Framework

Several variants of the original finite difference formulation [8] have been proposed in an effort to improve its performance in some respect (e.g. to decrease initialization sensitivity, increase the range of object shapes that can be modeled, decrease noise susceptibility, improve segmentation reproducibility for a class of objects, etc.). Finite element snakes [5,13], B-snakes [15,3], and Fourier descriptor snakes [19] are representative examples of these variants.

We have unified several snake variants within a finite element framework with different shape functions: Hermitian, B-spline, NURBS [20], Catmull-Rom, Bézier, Fourier, and Dirac delta. The shape functions generate different stiffness matrices and, in turn, yield different snake behaviors suitable for different tasks. For example, snakes that use B-spline shape functions are typically characterized by a low number of degrees of freedom, typically use polynomial basis functions of degree 2 or higher, and are inherently very smooth. Therefore, these “B-snakes” [15,3] can be effective in segmentation or tracking tasks involving noisy images and where the target object boundaries may exhibit significant gaps in the images. Alternatively, object boundaries with many fine details or rapid curvature variations may best be segmented by a snake that uses simpler shape

functions and many degrees of freedom such as the finite difference snake [8]. The unification of different shape functions in a single framework enhances the range of object modeling capabilities.

### 2.3 Initialization

An accurate initialization is generally needed in order for the snake to lock onto the image features. Therefore, 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 [5], the use of a chamfer distance map [5] and gradient vector flow [21]. 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 automatic techniques do not work as expected. For this reason, we are exploring an alternative direction—instead of attempting to automatically remove or decrease initialization sensitivity, we seek to increase the efficiency of interactive initialization. In particular, we enable the user to initialize snakes quickly and with minimal effort by exploiting the strengths of the livewire technique.

### 2.4 Livewire

Livewire is a recently proposed interactive boundary tracing technique [16,2,17,6]. Although it shares some similarities with snakes—it was originally developed as an interactive 2-D extension to previous stage-wise optimal boundary tracking methods—it is generally considered in the literature as a competing technique to snakes. Like snakes, the idea behind the livewire technique is to allow image segmentation to occur with minimal user interaction, while at the same time allowing the user to exercise control over the segmentation process. However, livewire realizes the idea differently from snakes.

In livewire, the user places an initial *seed* point near the boundary of the object of interest to begin the segmentation process. As the cursor, or *free* point is moved around, the current calculated boundary, the *livewire* or *trace*, is interactively displayed from the seed point to the free point. If the displayed trace is acceptable, the free point may be collected as an additional seed point, and the trace between the two adjacent seed points is frozen (locked). Each trace given by the livewire is the path with minimal cost between the two seed points resulting from the application of two-dimensional dynamic programming [6] or Dijkstra's graph search algorithm [17] with a local cost function. The local cost function assigns a lower cost to image features, such as edges, and consequently the livewire tends to stick to the object boundary guided with the seed points from the user. The resulting livewire boundaries are piecewise optimal (*i.e.* optimal between seed points), while the snake gives a global optimal solution over the entire contour.

## 2.5 Union of Snakes and Livewire

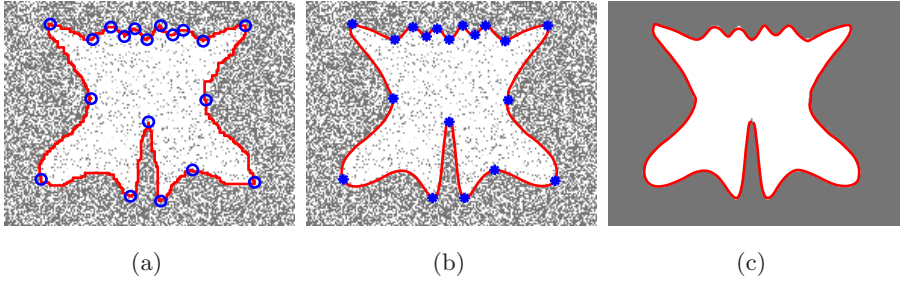
With livewire, the user has no control of the traces between seed points other than by backtracking. When the shape of the object boundary is complex, or when it is near other strong but uninteresting object boundaries, many seed points are needed in order to generate an acceptable result. Furthermore, when a section of the desired object boundary has a weak edge relative to a nearby strong edge, the livewire snaps to the strong edge rather than the desired weaker boundary. A method called *on-the-fly training* has been proposed to mitigate this problem [17]. However, the method relies on the assumption that the edge property is relatively consistent along the object boundary.

Livewire is fundamentally image-based. Thus, it cannot effectively bridge gaps where the desired object boundaries are missing, and the smoothness of the traces cannot be guaranteed. Therefore, it is desirable to allow the user to exercise control over the livewire traces between seed points, impose smoothness on livewire traces, and bridge gaps along object boundaries. This is what snakes are very good at doing. Snakes adhere to edges with sub-pixel accuracy and they may also be adjusted interactively as parametric curves with intuitively familiar physical behaviors. Moreover, unlike livewire, snakes have the power to track moving objects.

In most cases, however, livewire can quickly give much better results than casual manual tracing. Hence, the resulting livewire boundary can serve to quickly and effectively initialize a snake. The livewire seed points carry the user's prior knowledge of the object boundary. They can therefore serve as either hard or soft point constraints for the snake, depending on the user's confidence in the accuracies of the livewire seed points.

Because a livewire-traced initial object boundary is usually more accurate than a hand-drawn boundary, the incorporation of the seed points provided by the livewire trace as snake constraints results in a snake that very quickly locks onto the desired object boundary. If necessary, the user may correct errors inherited from the livewire-generated boundary, by applying mouse-controlled spring forces to the snake. Because the user still has the opportunity to correct the errors on the traces as the snake is deforming, the number of seed points needed to generate the initial livewire object boundary can be further reduced.

The combination of snakes and livewire relies on an efficient constraint mechanism. A constraint on a snake may be either soft or hard. Hard constraints generally compel the snake to pass through certain positions or take certain shapes [7,1], while soft constraints merely encourage a snake to do so. Two kinds of soft constraints, springs and volcanos, were described in the original finite difference snakes paper [8] and they are incorporated into our finite element formulation. A simple yet efficient way to impose hard constraints on snakes for the integration of snakes and livewire is to properly update the stiffness matrix and encode constraint values in the system force vector. This approach maintains the symmetry of the stiffness matrix for economical skyline storage and efficient computation via a single factorization [11]. Since they are encoded in the system force vector, the constraint values may be updated and other hard (or soft)



**Fig. 1.** Performance of united snakes demonstrated using a synthetic image with strong Gaussian white noise (variance 0.25). (a) A livewire is sensitive to noise (the required seed points are shown). (b) The united snake is robust to noise and accurately conforms to the boundary. (c) The united snake segmentation is close to the ideal boundary. The superior performance is a consequence of the imposed hard constraints (indicated by asterisks in (b)), without which the snake would slip away from high curvature points.

constraints may be added as the snake is deforming. Therefore, the user may dynamically adjust the constraints to refine the object boundary during snake deformation.

## 2.6 Summary

To summarize, the united snakes software unites several snake variants with livewire to provide a general purpose tool for interactive medical image segmentation which amplifies the efficiency, flexibility, and reproducibility of the component techniques. United snakes offer more control for relatively less user interaction. As it quickly locks onto the image features of interest with reasonable tolerance to errors in the livewire, the snake fully exploits the user guidance and expert prior knowledge reflected captured by the initial livewire trace.

As an initial demonstration that united snakes improves upon the robustness and accuracy of its component techniques, Fig. 1 shows a synthetic image of a known curve degraded with strong Gaussian white noise (variance 0.25). Given its image-based nature, the livewire is sensitive to noise as shown in Fig. 1(a). A snake initialized with the livewire gives a better result (Fig. 1(b)). Fig. 1(c) shows that the united snakes result is very close to the boundary in the ideal image, despite the strong noise. This performance is a consequence of the imposed hard constraints, without which the snake would slip away from high curvature points.

## 3 Applications

In this section, we demonstrate the potential of the united snakes technique in a series of experiments with medical images.

### 3.1 Segmenting Neuronal Dendrites in EM Images

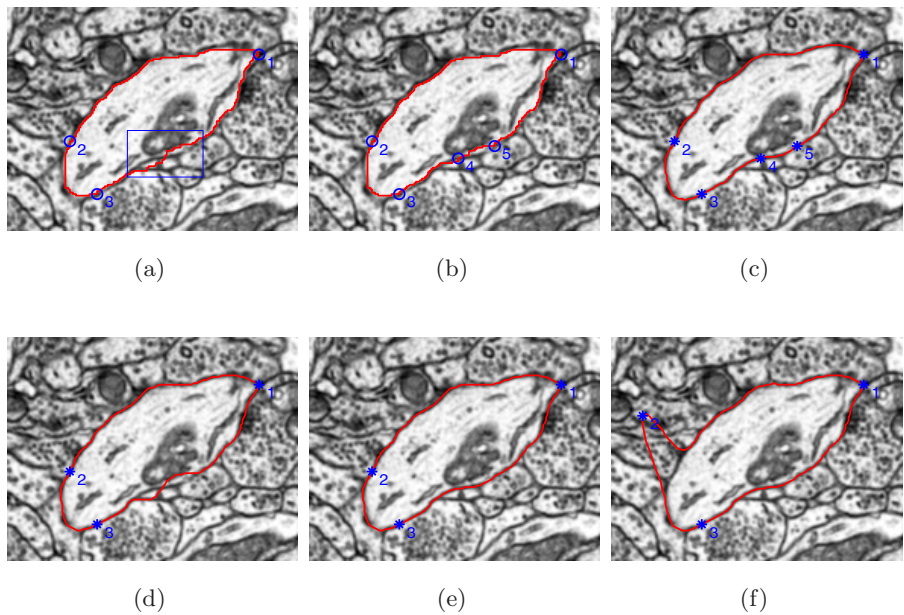
A neuronal dendrite is the receiving unit of a nerve cell. The area of contact between the dendrites of different cells is called a synapse and is located on the dendritic spines. In humans, changes in dendritic spines are seen with aging and with diseases that affect the nervous system, such as dementia, brain tumors and epilepsy [4]. Detailed anatomical models of dendritic spines and their synapses will provide new insights into their function, thus providing better opportunities to understand the underlying causes and effects of these diseases. To build such models, the dendrite must be segmented from the surrounding tissue in positive electron micrography (see [4] for a detailed description of how snakes are used in reconstruction of 3D nerve cell models from serial microscopy). Here, we are interested in localizing nerve cell membranes, which appear dark in positive micrography.

In the United Snakes system, the user begins an image segmentation task using a livewire. An initial seed point is placed near the boundary of the object of interest. As the cursor, or free point, is moved around, the livewire, or trace, is interactively displayed from the seed point to the free point. If the displayed trace is acceptable, the free point is collected as an additional seed point.

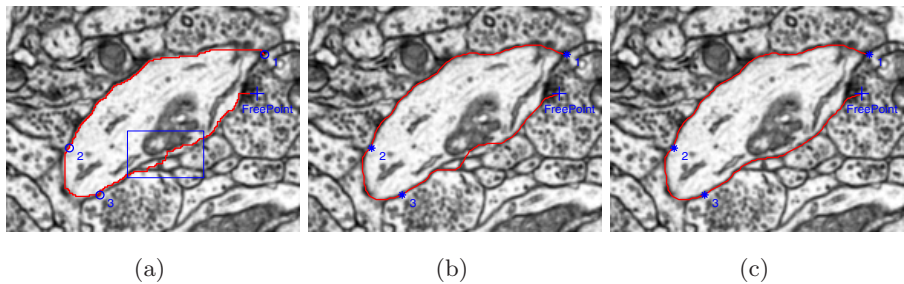
We can capture an approximate cell boundary in Fig. 2(a) with just three seeds. The livewire tends to stick to the object boundary using the seed points as a guide. The trace between the two adjacent seeds is frozen. The user has no further control over these traces other than backtracking. In order to generate a more accurate result in the area indicated by a rectangle, more seed points may be placed as in Fig. 2(b). Although the livewire boundary is somewhat jagged and exhibits some small errors, it is in general as accurate as manual tracing, but more efficient and reproducible.

Next, we instantiate a snake using the livewire-generated boundary to initialize the snake and using the seed points to constrain it. The user may select a shape function for the snake which is suitable for the object boundary. In our cell segmentation example, if the livewire result with five seed points is used to construct a finite difference snake, it is able to tolerate the livewire errors and very quickly and accurately lock onto the cell boundary without any need for further user interaction (Fig. 2(c)). Using the livewire result with three seed points, the snake becomes “stuck” in the problematic area (Fig. 2(d)) due to the livewire-generated boundary errors. However, this situation can be easily remedied using the mouse spring (Fig. 2(e)). Furthermore, as the snake is deforming, the hard constraints may be adjusted to refine the snake boundary. In Fig. 2(f) for example, constraint point 2 is moved to illustrate this snake boundary adjustment capability. By contrast, it is not nearly as easy to adjust a seed point in the livewire algorithm.

We refer to this form of livewire-snake integration as *static* integration—once the livewire result is used to initialize a snake, the segmentation process continues using only the constrained, user-controlled snake. The user may also set the united snake system to a more *dynamic* integration “mode”—once the livewire trace between the last seed point and the free point is formed, a corresponding



**Fig. 2.** Using united snakes in static mode to segment neuronal EM images. (a) Approximate livewire boundary using just 3 seeds (blue rectangle indicates a problem area). (b) Additional seed points can improve livewire’s accuracy. (c) Initialized from the livewire in (b), the snake tolerates livewire errors and locks on cell boundary without further user interaction. (d) Initialized from the livewire in (a), the snake “sticks” in the problem area, but it is easily adjusted (e) using the mouse. (f) Snake adjustment capability illustrated by moving constraint point 2.



**Fig. 3.** Using united snakes in dynamic mode to segment neuronal EM images. (a) Livewire boundary showing 3 seeds and free point (blue rectangle indicates a problem area). (b) Open snakes dynamically generated from the livewire trace and constrained by seed and free points. (c) Third snake corrected in the problem area using the mouse.



open snake with constraints at the seed point and the free point is constructed and automatically set in motion for deformation. When the free point is collected as a seed point, this open snake is merged with the snake constructed from the previous livewire traces (if they exist). All seed points are automatically applied as constraints. Fig. 3 illustrates this process where “+” indicates the current free point. The livewire and snake results are shown separately in 3(a) and (b) respectively. Since the snake is automatically set in motion, the user may use the mouse spring to correct it in any problematic areas along the snake (Fig. 3(c)).

### 3.2 Dynamic Chest Image Analysis

The aim of the dynamic chest image analysis project is to show focal and general abnormalities of lung ventilation and perfusion based on a sequence of digital chest fluoroscopy frames collected over a short time period (typically about 4 seconds) [10,12,9]. The project uses only plain X-ray fluoroscopy (with and without breathing) for the ventilation and perfusion studies. Consequently, the radiation dose to patients is low and, unlike a nuclear medicine scan, no preparation is required before the examination and radioactive isotopes are unnecessary. The information gleaned from these images is helpful in several aspects of cardiothoracic radiology. Diseases directly related to the parameters being measured include pulmonary embolism, pulmonary emphysema, cardiac failure, congenital heart disease and other diseases (tumors, obstructive lesions or infections) which may change pulmonary ventilation and/or perfusion. An essential first step for ventilation and perfusion analysis is the delineation of the lungs and the heart from each frame in a chest image sequence. The united snakes system is used to perform the delineation. Typically most of the user interactions to initialize and edit the snake are applied to the first image of the sequence only. The resulting snake is then propagated and deformed through the remaining frames of the image sequence.

As the chest image (Fig. 4(a)) shows, some segments of the lung boundary have strong edges and some segments are rather weak. The elliptical viewport generates very strong edges, consequently, part of the livewire between seed point 4 and seed point 1 snaps to the strong viewpoint edges rather than the desired lung boundary (Fig. 4(b)). On-the-fly training is not effective since the lung boundary does not exhibit consistent edge properties. Thus, the livewire alone requires much backtracking and many seed points (Fig. 4(c)) to generate an accurate lung boundary.

We construct a Hermite snake from the livewire traces in Fig. 4(b) and the first seed is used as a hard constraint on the snake since the edge information there is very weak. The snake can easily be pulled out of the strong edge and locked onto the lung boundary (as shown in Fig. 4(d)) without the use of on-the-fly training. Thanks to this hard constraint, it can firmly stick to the lung apex and follow the lung motion throughout the entire chest image sequence with minimal user interaction, by propagating the result from one frame to next frame. The first image contains the largest lung boundary in the sequence while



the smallest lung boundary is shown in Fig. 4(e). Fig. 4(f) shows an additional example.

In the case of the heart, Fig. 5(a) illustrates that part of the livewire trace from seed point 1 to seed point 2 is a straight line where the cardiac boundary is missing. Furthermore, at the bottom of the image, the livewire technique does not generate an acceptable cardiac boundary (i.e. from seed point 3 to seed point 1), so we have manually drawn a rough curve between the points. A least squares approximation to the initial livewire curve with a cubic B-spline with 5 knots (shown in Fig. 5(b)) can be used as an initialization to a B-snake. A hard constraint may be further imposed on control polygon node 3 to effectively bridge the gap along the heart boundary. The result is shown in Fig. 5(c) after only a few iterations. The B-snake then is used to track the heart motion through the image sequence.

### 3.3 Quantifying Growth Plates in MR images

The aim of the growth plate project is to determine the right time for surgery for patients with abnormal growth of the legs. To this end, the four tiny (essentially horizontal) lines in the image (Fig. 6(a)) must be detected to quantify the growth plate.

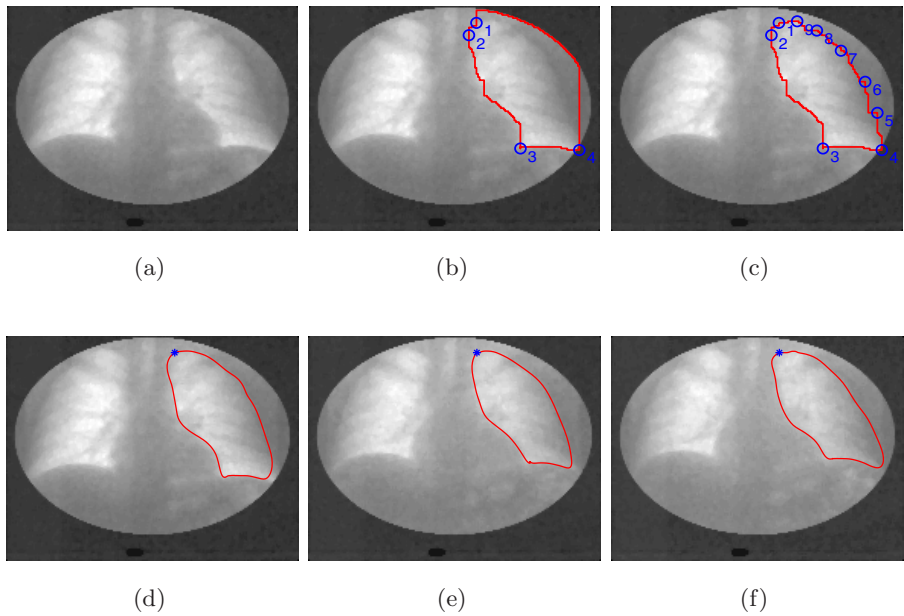
In this scenario, it is difficult for the user to manually trace an initial contour for a snake because of the small size of the lines and the small distance between each pair of lines. However, livewires can be used to quickly generate an acceptable snake initialization with just two or three seed points as shown in Fig. 6(b). In the final results shown in Fig. 6(c), two hard boundary conditions are applied on each of four finite difference snakes.

### 3.4 Other Segmentation Examples

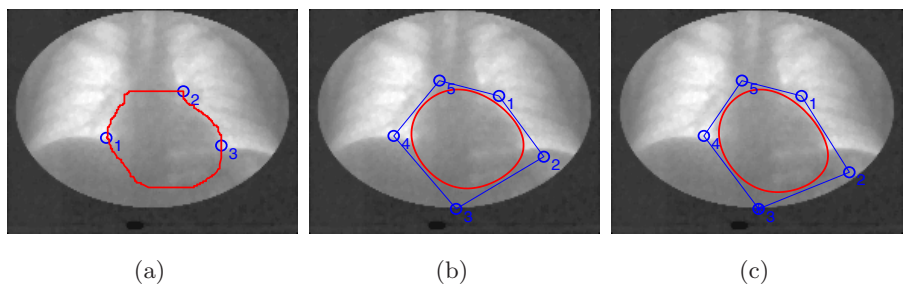
We have applied united snakes to two other medical images, an angiogram (Fig. 7) and a vertebra image (Fig. 8), to which Mortensen and Barrett applied their livewire algorithm in [17]. With only a few seeds, united snakes generate the boundaries shown in Figs. 7(c) and 8(c), which are comparable to the ideal boundaries used as references in [17].

## 4 Conclusion

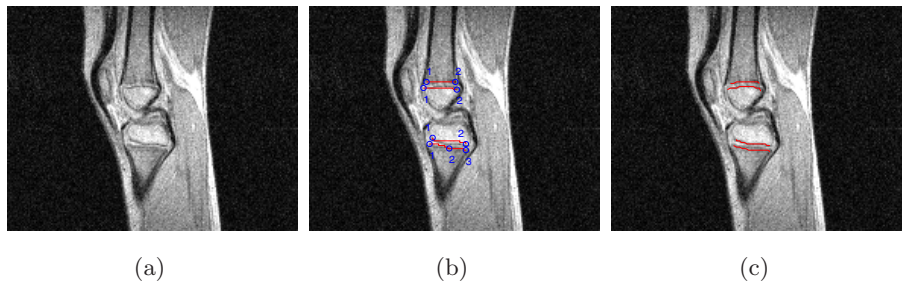
We have developed a software package called “United Snakes” which unites several snake variants with livewire to offer a general purpose tool for interactive medical image segmentation and tracking. The union of these techniques provides more flexible control than the individual techniques while reducing user interaction. We have demonstrated the generality, accuracy and robustness of united snakes in applications to the segmentation of neuronal dendrites in EM images, to dynamic chest image analysis, and to the quantification of growth plates, among other segmentation examples. It appears that united snakes are in several ways superior to livewire or snakes alone.



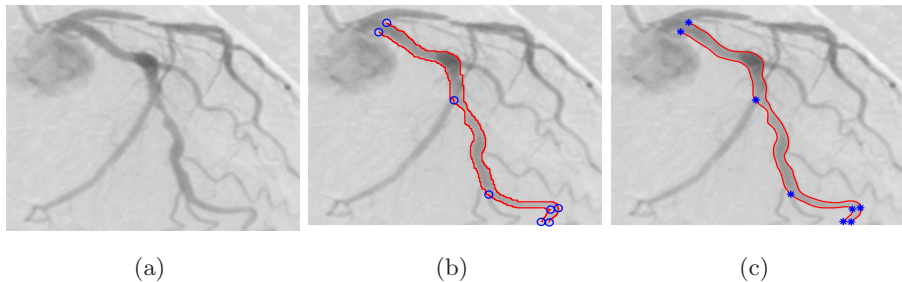
**Fig. 4.** Lung delineation in X-ray fluoroscopy image sequence. (a) First image in sequence. (b) Livewire generated boundary. (c) Livewire trace requires backtracking and additional seed points to correct boundary delineation. (d) Hermite snake constructed from livewire trace in (b) and constrained by seed point 1. (e-f) Segmentation of other images in the sequence.



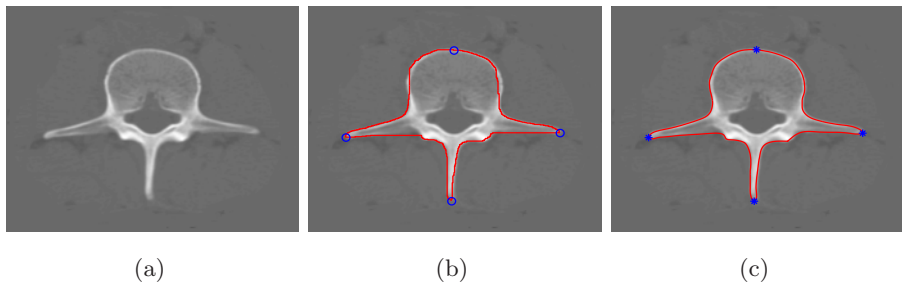
**Fig. 5.** Heart delineation in X-ray fluoroscopy image sequence. (a) Initial livewire generated boundary. (b) Initial B-spline snake (red) and control polygon (blue) constructed from livewire trace. Control point 3 is used as a hard constraint. (c) Resulting segmentation.



**Fig. 6.** Quantifying growth plates in MR images. (a) An MR growth plate image. (b) The livewire results. (c) The united snakes results.



**Fig. 7.** Segmenting a vessel in an angiogram. (a) The image used in [17]. (b) Livewire segmentation. (c) United snake generates boundaries comparable to *ideal* boundaries in [17].



**Fig. 8.** Segmenting the outer boundary of a vertebra. (a) The image used in [17]. (b) Livewire segmentation. (c) United snake boundary is comparable to the *ideal* boundary in [17].

## Acknowledgments

JL gratefully acknowledges the valuable comments and suggestions of Prof. Timo Järvi as well as the support of the Turku Centre for Computer Science and the Instrumentarium Foundation. The chest image was collected by Dr. Raimo Virkki and provided by Dr. Aaro Kiuru. The growth plate image was provided by Prof. Martti Kormano and Dr. Matti Sauna-aho. The cell image was obtained from Dr. Kristen Harris of the Harvard Medical School. The angiogram and spine images were provided courtesy Eric Mortensen of Brigham Young University.

## References

1. A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990. 119
2. W. Barrett and E. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331–341, 1997. 116, 118
3. A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998. 116, 117
4. I. Carlbom, D. Terzopoulos, and K. Harris. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. *IEEE Trans. on Medical Imaging*, 13(2):351–362, 1994. 121
5. L. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993. 116, 117, 118
6. A. X. Falão, J. K. Udupa, S. Samarasekera, and S. Sharma. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60:233–260, 1998. 116, 118
7. P. Fua and C. Brechbühler. Imposing hard constraints on deformable models through optimization in orthogonal subspaces. *Computer Vision and Image Understanding*, 65:148–162, 1997. 119
8. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 116, 117, 118, 119
9. J. Liang, A. Haapanen, T. Järvi, A. Kiuru, M. Kormano, E. Svedström, and R. Virkki. Dynamic chest image analysis: Model-based pulmonary perfusion analysis with pyramid images. In E. A. Hoffman, editor, *Medical Imaging 1998: Physiology and Function from Multidimensional Images*, pages 63–72, San Diego, CA, 1998. 123
10. J. Liang, T. Järvi, A. Kiuru, M. Kormano, E. Svedström, and R. Virkki. Dynamic chest image analysis: Model-based ventilation study with pyramid images. In E. A. Hoffman, editor, *Medical Imaging 1997: Physiology and Function from Multidimensional Images*, pages 81–92, Newport Beach, CA, 1997. 123
11. J. Liang, T. McInerney, and D. Terzopoulos. United snakes. In *Proc. Seventh International Conf. on Computer Vision (ICCV’99)*, Kerkyra (Corfu), Greece, September 1999. IEEE Computer Society Press. 116, 117, 119
12. J. Liang, R. Virkki, T. Järvi, A. Kiuru, M. Kormano, and E. Svedström. Dynamic chest image analysis: Evaluation of model-based ventilation study with pyramid images. In R. Zurawski and Z.-Q. Liu, editors, *IEEE First International Conference on Intelligent Processing Systems*, pages 989–993, Beijing, China, 1997. 123

13. T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, 1995. [117](#)
14. T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, 1996. [116](#)
15. S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: Implementation and application to stereo. In *Proceedings DARPA*, pages 720–726, 1990. [116](#), [117](#)
16. E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceedings of Computer Graphics (SIGGRAPH'95)*, pages 191–198, Los Angeles, CA, August 1995. [116](#), [118](#)
17. E. N. Mortensen and W. A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384, 1998. [116](#), [118](#), [119](#), [124](#), [126](#)
18. A. Singh, D. Goldgof, and D. Terzopoulos, editors. *Deformable Models in Medical Image Analysis*. IEEE Computer Society Press, 1998. [116](#)
19. L. Staib and J. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, 1992. [116](#), [117](#)
20. D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, 1994. [117](#)
21. C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998. [118](#)