

# Progressive Livewire for Automatic Contour Extraction

Steven Marx and Chris Brown  
University of Rochester Computer Science Department

August 25, 2000

## 1 Problem

Segmentation via contour finding is an important first step in many image processing applications. Reliable contours are usually found by a combination of human and computer image analysis, with the human interactively guiding the computer and editing the results. In the field of medical image sequence analysis, we would like to minimize human interaction by taking advantage of coherence between neighboring images in a sequence. We aim to “track” structures such as brain, skin, or vessels in sequences of MRI and CT scans and use the resulting contours to produce a triangulated surface.

Our goals in creating progressive livewire were (1) to produce a complete contour at pixel-level accuracy, (2) to require as little user interaction as possible, both in the first image and in consecutive images, and (3) to produce a fast and adaptable tool that can be used in many applications with few manual adjustments.

## 2 Prior Work

Medical image analysis (e.g. [14]) has explored both contour-based (edge-finding) and region-growing approaches to extracting structures. We aim to find surfaces such as the brain, skin, ventricles, vessels, and tumors by a contour-based segmentation technique.

Snakes (active contours) [11; 9; 15; 16; 3] and their three-dimensional counterparts, deformable surfaces [4; 5] are popular in contour-finding, and have been applied to medical images (e.g. [10]). A snake is defined by setpoints (snaxels), which move to minimize the snake’s total energy. A snake’s energy is often defined by two energy components, internal energy and external energy. Internal energy is usually designed to simulate physical properties, such as propensities for minimal size and constant curvature. External energy comes from the image (often a smoothed brightness gradient) or user interaction. Snaxels usually must be separated by several pixels for curvature (internal energy) computations to be accurate, so if a complete contour is needed, interpolation (often with splines) defines the contour between snaxels.

Livewire [2; 13] is an interactive, pixel-level contour-finding algorithm. It finds a minimal-cost path between a start and a goal point in a directed graph. An edge in the graph represents the border between two neighboring pixels in the original image. Each of these boundary elements (bels) is assigned a cost. The cost function can be arbitrary, but is often the sum of weighted terms

expressing good contour properties. User interaction is required to choose an appropriate starting point and successive “control points” along the correct contour.

Progressive livewire combines the strengths of both techniques. We train livewire and use it to segment a single image in a sequence of medical images. Points of high curvature from this complete contour are used to initialize a special type of snake. The snake is copied to the next image in the sequence, where it actively finds an optimal configuration. The new snaxel locations are used to construct a new livewire contour to regain pixel-level accuracy. Livewire is then automatically retrained, and the process repeats on subsequent images until the entire sequence has been segmented.

### 3 Method

The progressive livewire process is as follows:

1. Determine relative spatial resolutions for contour phenomena, lengths of search loci, and cost functions to detect the contour.
2. If necessary, preprocess for noise.
3. Interactively train livewire and segment an initial image.
4. Find points of high curvature and snaxel search loci, and construct a snake.
5. Minimize the snake on the next image and construct a new livewire contour.

Steps three through five are repeated on subsequent images until the entire sequence has been segmented.

**Determining Variables and Functions:** Certain variables customize progressive livewire to a particular application. The spatial resolution for curvature computation can be determined by experimentation and inspection, as can the lengths of search loci. Many cost functions are possible, but the training process eases the job of setting their parameters (Section 5).

**Preprocessing:** In applications where images are likely to be noisy, as in medical image analysis, a preprocessing step to remove noise is prudent. For the experiments discussed in the paper, we applied a  $3 \times 3$  median filter to all images before segmentation. Correcting intensity non-uniformities simplifies boundary-finding algorithms as well.

**Livewire:** We implemented livewire as in [2], experimenting with many of the cost functions enumerated there. For our application, the cost function is the sum of two cost components, equally weighted. Each component  $C_i$  has the form

$$C_i(b) = -N[\mu_i, \sigma_i](f_i(b)), i = 1, 2,$$

where  $b$  is the boundary element and  $f_i(b)$  is the scalar feature. The contribution of a cost component is the value of an (inverse) Gaussian probability density function with mean and standard deviation equal to those of a scalar boundary feature, computed during training (of an interactively drawn boundary or a previously-found acceptable boundary segment). For speed, the inverse Gaussians are implemented by a look-up table of the 256 possible grayscale values in our 8-bit

images. The two scalar boundary features we used are simply the pixel intensities on either side of the oriented boundary element.

The shortest path algorithm we used is Dijkstra’s Algorithm, implemented with a priority queue [7]. Dijkstra’s Algorithm gives the shortest path from a start point to every other point in the image, so the user can interactively see what contour will be produced by choosing a given endpoint. The user needs simply click an initial point on the contour and then interactively choose a few more points along the boundary, “retraining” on the current contour as necessary. During this process, control points can be deleted and changed in case of user error (i.e. missed mouse clicks).

**High Curvature:** A snake is constructed from the original livewire contour by choosing points of high curvature from the contour. Curvature is defined by a discretized first-derivative computation:

$$C(s) = (V(s - r/2) - 2V(s) + V(s + r/2))^2$$

where  $s$  is the position of a point along the contour and  $V(s)$  is the location vector of point  $s$ . Because of snake inaccuracy with close snaxels, points of high curvature are constrained to be no closer than  $r/2$  pixels apart. The resolution should be set based on the overall curviness of the object to be segmented, where objects with small details require lower values of  $r$ . For the work discussed in the paper,  $r \approx 10$  is satisfactory. An  $r$  chosen too high will result in too few control points to construct an accurate contour, and a snake with an  $r$  chosen too low is slower and more susceptible to local image noise.

**Snake:** The snaxel search loci are straight line segments centered on the maximum curvature points, with direction the average of neighboring snake segment normals [12]. The energy of the snake is the sum of optimal livewire contours connecting neighboring snaxels, such that minimizing the overall snake energy also minimizes the overall cost of the resulting livewire. Snake minimization is performed over one iteration of dynamic programming [1; 6]. In computing the livewire paths between snaxels, a threshold limits the search space for purposes of speed. This threshold is raised only if necessary to span the gap between neighboring snaxels.

When snake minimization is complete, a new livewire is constructed from the snaxels and trained on this new contour, and the process is progressively applied to each image in the sequence.

## 4 Results

Typical results are shown in Fig. 1.

## 5 Discussion

Progressive livewire finds minimal-cost paths, so the choice of cost function is a central issue. There are many easily implemented cost functions and a huge number of ways to combine them. Automatic training makes it possible to test the potential of any cost function. We experimented with nearly all the cost functions described in [2], called  $f1 - f8$ .

Features  $f3$  through  $f6$  are gradient magnitude approximations. Each of these weighted with an inverse Gaussian function and properly trained yielded acceptable results, but the drawback to using these cost components is that contours with low intensities can have the same gradient magnitude as contours with high intensities. Feature  $f8$  is a measure of how close a livewire contour is to the

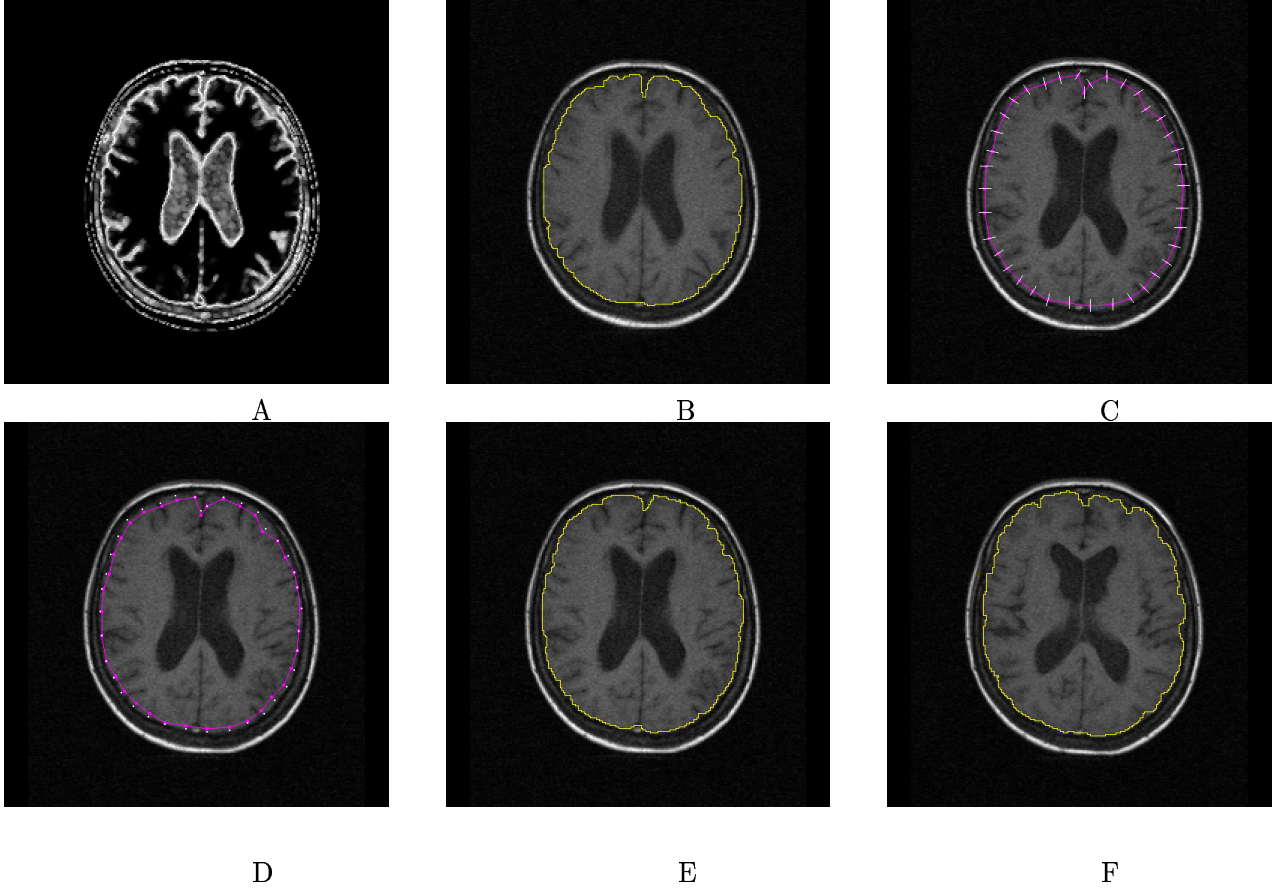


Figure 1: A: Livewire cost function, slice 1. B: Livewire contour, slice 1. C: Maximum curvature points, initial snake (purple) and search loci (white), slice 2. D: Snake minimized on slice 2. E, F: progressive livewire contours, slices 2 and 3. Images are from a T1-weighted axial MRI sequence, courtesy of the University of Rochester Medical Center. Contour discontinuities are a conversion artifact.

contour extracted from the previous image. This is a powerful feature when segmenting sequences of images, but because snake minimization between images constrained consecutive livewires to vary little in location,  $f8$  is of little use in a progressive livewire.

Features  $f1$  and  $f2$  are the simplest features and also the most useful in our experiments.  $f1$  is defined simply as the intensity on the inside of the boundary, and  $f2$  is the intensity on the outside of the boundary. By weighting each with its own inverse Gaussian and combining the two components, we were able to train our cost function to differentiate between contours that otherwise exhibited similar characteristics. In a sense, these features allow livewire to follow properties on both sides of the contour simultaneously and directly, rather than indirectly according to a gradient measure. Large-scale intensity nonuniformities should be corrected to keep them from biasing the results, but our images did not need this step.

Even with a well-chosen cost function and extensive training, progressive livewire, like most image segmentation algorithms, does not yield perfect results every time. Livewire is a minimal-cost searching algorithm, so it inherently favors short paths. When contour characteristics are highly variable across the object boundary, the standard deviation rises, having the effect of generally raising the cost of all contours, leaving less difference between good contours and bad contours.

This leaves livewire less likely to choose the correct contour and more likely to cut off convexities and skip over concavities in the image.

Another problem that arises in using progressive livewire is when an incorrect contour is locally superior to the correct contour. One place this can happen is in head MRIs where bone marrow can sometimes look exactly like brain matter. When, for some reason, the actual brain surface differs locally from the rest of the surface, neighboring bone marrow may attract the path. Using additional snake internal energy functions may help reduce this problem.

## 6 Future Work

Progressive livewire is a very general algorithm that can be augmented in any number of ways. More sophisticated cost components and automated cost function construction, a priori knowledge, localization of training, and incorporation of more information can all easily improve on the basic algorithm.

Deriving good cost functions automatically from sample input and desired output ensembles is a challenging mathematical problem. However, it is also hard to search for good choices, weights, and combinations of cost functions. Genetic programming techniques are a natural choice for automating this search.

A priori knowledge is always an aid to a low-level image processing algorithm, and such knowledge is often available. In the case of head MRIs, we know in advance that there will be skin, skull, and bone marrow outside the brain. Using this knowledge, a higher level process could remove potentially distracting contours from the image. One possible approach is to examine an intensity histogram over a search path starting from outside the head and ending inside the brain. Ideally, the skull and bone marrow would be recognizable from the histogram and easily removed from the image. Another approach is to track two contours simultaneously, one for bone marrow and one for the brain surface, not allowing either to cross into the other.

A problem discussed in Section 5 is the case of a contour of varying local gradient. In medical images, this can occur in abdominal CTs where internal organs are pressed against each other so that one portion of a contour may vary drastically from another portion. Progressive livewire as described here would be likely to fail in this situation, as no cost function would yield a consistent characterization of the contour. A possibility is to “sector” the contour [8], training different portions of the contour differently. Each segment of the contour would then be able to search according to its own local criteria in subsequent images. Different cost functions as well as different training values could be used in this way.

Another way to improve the performance of progressive livewire would be to add more image information. By using several registered images of different modalities, the cost function could be a composite of features from different types of images. This multi-spectral information could aid in differentiating between similar-looking contours. In the application of medical imagery, MRI scans are often produced at different weights, each incorporating different information. Combining all this multi-spectral information would be a natural extension of progressive livewire.

## References

- [1] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

- 12(9):855–866, September 1990.
- [2] A. Falcao, J. 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.
  - [3] A. Blake and M. Isard. *Dynamic Contours*. Springer Verlag, 1998.
  - [4] J. Coughlan, A. Yuille, C. English, and D. Snow. Efficient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, June 2000.
  - [5] C. Davatzikos and R. Bryan. Using a deformable surface model to obtain a shape representation of the cortex. *IEEE Transactions on Medical Imaging*, 15(6):785–795, December 1996.
  - [6] J. Denzler and H. Niemann. Evaluating the performance of active contour models for real-time object tracking. In *Second Asian Conference on Computer Vision*, pages II/341–II/345, Singapore, 1995.
  - [7] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik* 1, pages 269–271, 1959.
  - [8] S. Fenster and J. Kender. Sectorized snakes: Evaluating learned-energy segmentations. In *Proceedings of the International Conference on Computer Vision*, 1998.
  - [9] S. Gunn and M. Nixon. A dual active contour for improved snake performance. Technical report, University of Southampton Department of Electronics and Computer Science, 1995.
  - [10] O. Jesorsky, J. Denzler, E. Nöth, and T. Wittenberg. Analysis of voice-onset using active rays and hidden markov models. In *Advances in Quantitative Laryngoscopy*, pages 51–58, Erlangen, 1997.
  - [11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, 1988.
  - [12] C. Maurer, G. Aboutanos, B. Dawant, R. Maciunas, and J. Fitzpatrick. Registration of 3-d images using weighted geometrical features. *IEEE Transactions on Medical Imaging*, 15(6):836–839, December 1996.
  - [13] E. Mortensen and W. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384, 1998.
  - [14] S. Sonka and J. M. Fitzpatrick. *Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis*. SPIE Press, 2000.
  - [15] D. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision Graphics and Image Processing: Image Understanding*, 55(1):14–26, January 1992.
  - [16] C. Xu and J. Prince. Gradient vector flow: A new external energy source for snakes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–71, 1997.