Live level set: A hybrid method of livewire and level set for medical image segmentation

Jianhua Yao^{a)}

Diagnostic Radiology Department, Warren G. Magnuson, Clinical Center, National Institutes of Health, Bethesda, Maryland 20892

David Chen

Diagnostic Radiology Department, Warren G. Magnuson, Clinical Center, National Institutes of Health, Bethesda, Maryland 20892, Medical Informatics, Stanford University, 251 Campus Drive, Suite X-215, Stanford, California 94305-5479

(Received 29 April 2008; revised 18 July 2008; accepted for publication 18 July 2008; published 18 August 2008)

Livewire and level set are popular methods for medical image segmentation. In this article, the authors propose a hybrid method of livewire and level set, termed the live level set (LLS). The LLS replaces the one graph update iteration in the classic livewire with two iterations of graph updates. The first iteration generates an initial contour for a level set computation. The level set distance is then factored back into the cost function in the second iteration of graph update. The authors validated LLS using synthetic images. The results show that the performance of the LLS is superior to both the classic live wire and traditional level set methods in terms of accuracy, reproducibility, smoothness and running time. They also qualitatively evaluated the LLS using real clinical data. © 2008 American Association of Physicists in Medicine. [DOI: [10.1118/1.2968876](https://doi.org/10.1118/1.2968876)]

Key words: livewire, level set, medical image segmentation

I. INTRODUCTION

Interactive/semiautomatic segmentation methods that require user assistance range from complete manual drawing of object boundaries to detection of object regions with minimal user assistance.¹ The goal of research in interactive segmentation should be to increase the accuracy while minimizing user involvement and decreasing the total user's time for segmentation. Some popular approaches in semiautomatic segmentation are active contour models,² active shape models (ASMs),³ graph cut,^{4,5} and the methods we use, livewire⁶⁻⁹ and level set.¹⁰⁻¹² The active contour method uses an initial approximation of an object's boundary and iteratively minimizes a corresponding energy function to refine the contour. ASMs search for boundaries by applying statistical models garnered from training data. All these segmentation methods have their pros and cons. Some researchers tried to incorporate the best characteristics of different strategies. For instance, United Snakes¹³ combined the livewire^{7,8} and active contour² techniques. The traces generated from livewire are used as the initial for further active contour segmentation. In this way, it imposes smoothness on livewire traces and bridges complicated gaps along object boundaries.

Livewire (aka intelligent scissors) was introduced in 1992.^{7,8} It is a user-steered method that relies on finding the shortest path between nodes in a weighted graph. The shortest path represents the suggested contour that minimizes the cost from the user's current mouse position to the start position. It allows for a balance between user interaction and automatic analysis needed for image segmentation. Falcao *et al.*¹ proposed a live lane algorithm to restrain the search domain and speedup the algorithm. Similarly, an enhanced lane algorithm was proposed by Kang *et al.*¹⁴ to restrict the search

domain by following the target boundary. More recently, approaches using live wire have been applied to three-dimensional (3D) segmentation. Souza *et al.*¹⁵ proposed an approach known as iterative user steered 3D image segmentation. In this method livewire results from one slice are propagated to subsequent slices by projecting the midpoints of livewire segments onto the new slices to act as anchor points. Lu *et al.*¹⁶ developed a 3D live-wire method on computed tomography (CT) chest image analysis. Graph cut^{4,5} is another segmentation technique based on combinatorial optimization using graphs. The user roughly marks the objects and background on the image, and the maximum flow is calculated to locate the object boundary. The initial markers can be loosely placed; however, multiple refinements may be needed to achieve accurate segmentation.

Level set algorithms rely on the propagation of an initial boundary under the influence of image forces.^{10,11} The contour is embedded as the zero level set of a high dimensional function known as the level set function. This method has been previously applied in medical image segmentation, such as a system to track the volume growth of uterine leiomyomas over time.¹² Another class of level set algorithms are region-based methods,¹⁷ where various low-level assumptions regarding the intensity, color, texture, or motions are applied. Shape priors extracted from training data can also be used to guide the level set.¹⁸ Due to the intensity assumption and prior information, the region-based level set methods are usually application specific. To address this issue, nonparametric statistics such as mutual information¹⁹ and Markov models²⁰ had also been applied in curve evolution and pixel classification. The nonparametric methods are unsupervised and do not require prior information about the

probability density distribution of the image. Therefore, they can be employed to solve a variety of challenging image segmentation problems. In recent years, several groups^{21–23} extended the level set methods to texture image segmentation. Those methods are unsupervised and rely on nonparametric statistics in image neighborhoods for level set computation in sophisticated images. Level set methods are generally very slow, taking from a few minutes¹⁹ to near an hour²⁰ on a standard image. Since we are investigating interactive segmentation, we require the segmentation to be completed within a few seconds. Hence, we cannot afford a full level set execution on the entire image.

The graph based livewire method has the advantage of quickly finding a global minimum of the energy function; however, the discrete topology of graphs may produce segmentation artifacts due to the noise along the boundary. The level set method is based on a continuous formulation and can produce smooth segmentation; however, it usually requires good initialization and is a time consuming operation. Medical images have many inherent characteristics that make designing a segmentation method, whether it be automatic or semiautomatic, difficult. For instance, the abundant amount of noise in medical images and the lack of strong boundaries can result in inaccurate regions when subjected to either the level set or livewire method. From our experience, neither method alone, the livewire or level set, can accurately segment objects such as uterine fibroids, the liver and knee cartilage due to their indistinct boundaries, noise and image artifacts (see Fig. 12 in Sec. III). For more defined boundaries like bone, these methods can garner better results.

Based on the above observation, we propose a method to incorporate level set distance into the cost function of the livewire to extract an accurate representation of the boundary. In our method we use the active livewire contour as the initial level set. We then take the distance from the zero-level set to factor into the cost function in determining the best path. We name the hybrid method the live level set (LLS). We must emphasize that the proposed method is an interactive segmentation technique with the intention to improve the classic livewire method.

In Sec. II we will describe the classic livewire and traditional level set method, as well as our proposed LLS method. We also demonstrate the advantages of LLS over the original methods. Our experiments and results are presented in Sec. III. Finally, we discuss the limitations and future plans in Sec. IV.

II. METHODS

Our method, which is termed the live level set (LLS), builds on top of the classical livewire approach. The major enhancement of the LLS is a new adaptive graph generation that requires two iterations to incorporate level set distance in the cost function.

FIG. 1. Graph and shortest path. (a) Graph representation of an image. Circles represent nodes (pixels in images), and the lines between nodes represent edges. Each node is connected to its eight neighbors. (b) Example of local costs between nodes. The numbers represent the weight of an edge. (c) Bold lines represent the least cost path from the solid circle (the start point) to all other points in the graph. Numbers represent the total cost to get to that node.

II.A. Classic Livewire

Livewire relies on using dynamic programming in the form of Dijkstra's graph search algorithm to find the optimal cost path between a start node and all other nodes in a weighted graph.

In the original implementation of livewire,^{7,8} the complete image is represented as a graph with each pixel representing a node and edges representing the connection between one pixel and each of its eight neighboring pixels (Fig. 1). The graph can be represented as an array of nodes containing a unique identifier, pointers to its eight neighboring nodes, the cost to arrive at that node from the start node, and a back-pointer which dictates the path with the lowest cost. Figure 1 shows the graph representation of an image and the shortest path between nodes.

The cost function which determines the cost of an edge between two pixels can be computed using various pixel characteristics. Mortenson⁶ used a combination of Laplacian zero-crossing, gradient magnitude, and gradient direction, which is defined as

$$c_m(u, v) = \omega_Z \cdot f_Z(u) + \omega_G \cdot f_G(u) + \omega_D \cdot f_D(u, v), \quad (1)$$

where f_Z is the Laplacian zero-crossing, f_G is the gradient magnitude, f_D is the gradient direction, u and v are neighboring pixels forming an edge in the graph, and each ω is the weight of corresponding feature function.

The flow chart of the classic livewire algorithm is in Fig. 2. The start node in the livewire implementation is the initial mouse click, also termed the seed node. From the seed node we run the dynamic programming algorithm to update the

FIG. 2. Flow chart of the classic livewire algorithm.

graph. While we iterate through this algorithm we immediately look at the cost from a node to neighbor nodes and assign backpointers along the shortest path based on the costs. After the shortest paths from all nodes to the seed node are determined, the user may move the mouse to other pixels and a contour will be drawn based on where the backpointers direct. This series of steps continue as long as the user has more contours to draw.

II.B. Traditional level set method

Level set methods are part of the family of segmentation algorithms that rely on the propagation of an initial boundary under the influence of image forces.¹¹ Level sets are evolving interfaces (contours or surfaces) that can expand, contract, split, or merge. The underlying idea behind the level set method is to embed the moving interfaces as the zero level set of a higher dimensional function $\phi(x, t)$, defined as

$$\phi(x, t) = \pm d, \quad (2)$$

where $\pm d$ is the signed distance to the interface from point x , and t is the time index. Here, x is outside the interface when $\phi(x, t) < 0$, inside the interface when $\phi(x, t) > 0$, and on the interface when $\phi(x, t) = 0$. The evolution of $\phi(x, t)$ can be represented by a partial differential equation

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot x'(t) = 0. \quad (3)$$

Define the scalar speed field F as $F = n \cdot x'(t)$, where $n = \nabla \phi / |\nabla \phi|$ is the normal direction of the level set function, and the above equation becomes:

$$\frac{\partial \phi}{\partial t} + F |\nabla \phi| = 0. \quad (4)$$

The speed function F can be broken down to three parts: propagation, curvature, and advection speed. Then the level set equation can be further expressed as

$$\frac{\partial \phi}{\partial t} = \alpha \tilde{g}(x) \delta |\nabla \phi| + \beta \tilde{g}(x) \kappa(x) |\nabla \phi| + \gamma \nabla \tilde{g}(x) \cdot \nabla \phi, \quad (5)$$

where $\tilde{g}(x)$ defines the propagation term, $\tilde{g}(x) \kappa(x)$ defines the curvature term, $\nabla \tilde{g}(x)$ defines the advection term, α , β , and γ are weighting parameters for each term, and δ is the step size. We used the ITK implementation of the level set algorithm in our system.²⁴

The Laplacian level set defines the speed function based on the Laplacian of images values. The goal is to attract the evolving level set interface to local zero-crossings in the Laplacian image. In our implementation, the image is first convolved with gradient anisotropic diffusion filter.²⁴ The speed image for Laplacian level set is

$$\tilde{g}_L(x) = \nabla^2 \{A[I(x)]\}, \quad (6)$$

where I is the image intensity, $A(\cdot)$ is the anisotropic diffusion, and ∇^2 is the Laplacian operator. $\tilde{g}_L(x)$ is used to replace $\tilde{g}(x)$ in Eq. (5). The Laplacian level set is best known

FIG. 3. Level set distance. The level set distances from outside to inside are -10 , -5 , 0 , 5 , and 10 .

for refining an existing segmentation which is close to the final solution.

One example of the level set distance function $\phi(x, t)$ is illustrated in Fig. 3 after 2000 iterations of Laplacian level set. The zero level set is on the boundary of the circle.

II.C. LLS

Our proposed method termed the live level set (LLS) builds off of the original concept of livewire by introducing two iterations of graph updates with a hybrid cost function dependant on the result of the shortest path produced from the first iteration. We also develop a method to apply an adaptive search space rather than a global search with a purpose to improve the efficiency. The idea of restricting the searching space is borrowed from the live lane and enhanced lane techniques.¹

II.C.1. Adaptive search space

The search space for graph update is adaptively expanded dependant on the shortest path and the current mouse location. In this scheme once the mouse point leaves the current search space, a new search space is expanded that is inclusive of the original search space and a new circular search space centered at the mouse point with radius of a predefined expansion range (default at 20 pixels).

II.C.2. Hybrid cost function

We propose two modifications to the cost function in the LLS. The first one is to use a local maximum gradient rather than a global maximum gradient to normalize the cost, and the second one is to factor in the level set distance. The LLS consists of two different iterations of graph updates using two different cost functions.

The cost function in the first iteration is only dependent on the gradient magnitude. As a higher magnitude denotes a stronger edge, a lower cost is assigned by subtracting it from the maximum gradient magnitude within the search space and normalizing it within the span of gradients. By narrowing the search space, we also limit the image gradients that are relevant in the cost function. Using the maximum gradient within an arbitrary region or within the whole image may at times influence the cost function adversely. If the region of interest has low gradient boundaries, a normalization based

on a high gradient may dilute the cost. This dilution would favor straight lines as opposed to curves, which may lead to inaccurate boundaries. To address this issue, we propose to normalize the cost function based on the maximum gradient within the adaptive search region. This allows us to have more control over preventing external influences from acting on characteristics that are more consistent with the boundary of interest. The cost function of edge (u, v) in the classic livewire method is

$$c_g(u, v) = \max G - \frac{1}{2}[g(u) + g(v)], \quad (7)$$

where $\max G$ is the maximum gradient of either the whole image or a predefined area.

Accordingly, our normalized cost function is

$$c_n(u, v) = \frac{\max G(N) - \frac{1}{2}[g(u) + g(v)]}{\max G(N) - \min G(N)}, \quad (8)$$

where $\max G(N)$ and $\min G(N)$ represent the maximum and minimum gradient magnitude within an adaptive search space N defined in Sec. II C 1.

Using the normalized cost function, the shortest path from the current mouse position to the seed point, which we term the transient path, is computed. The first and last points on the transient path are connected to form a closed contour and used as initial value for level set computation. The initial level set distance is assigned as 0.5 inside the contour and -0.5 outside. The initial values are then passed to a Laplacian level set. A hybrid cost function is then derived based on the level set distance

$$c_h(u, v) = c_n(u, v) + w_l^* \frac{1}{2} \left[\left| \frac{\phi(u, t)}{\max \phi(N)} \right| + \left| \frac{\phi(v, t)}{\max \phi(N)} \right| \right]. \quad (9)$$

Here, the hybrid cost function takes the normalized gradient magnitude cost function $c_n(u, v)$ and adds a weighted average of the normalized level set distances of pixel u and v . $\phi(u, t)$ and $\phi(v, t)$ are level set distances of pixels u and v , respectively, $\max \phi(N)$ is the maximum level set distance in the search space, and w_l represents a weight that can be adjusted to place more or less emphasis on the level set contribution.

Figure 4 illustrates the computation of the hybrid cost function from the normalized gradient cost function and level set distance. The different Euclidean lengths between diagonal and nondiagonal edges are also taken into consideration in the cost function calculation. If the two pixels do not form a diagonal edge, the cost function is compensated by $\sqrt{2}$, i.e., $c(u, v) = c(u, v) / \sqrt{2}$. In Fig. 4 the first iteration using $c_n(u, v)$ failed to converge to the correct boundary, after factoring the level set distance in $c_h(u, v)$, the livewire get the correct contour.

FIG. 4. Hybrid cost function in the LLS. Dark circle indicates the start point, light circle indicates current mouse position. (a) The graph with weighted edges using normalized cost function $c_n(u, v)$. (b) Transient path (dark blue lines). The numbers represent the total cost to get to a certain position. (c) Distance from the zero level set when Laplacian level set is applied to the transient path. (d) Graph generated with the hybrid cost function $c_h(u, v)$, $w_l=1$. (e) Shortest path (dark blue lines) computed from the graph in (d).

II.C.3. Overall LLS methodology

The flow chart of the LLS is shown in Fig. 5. The user's first mouse click on the image is considered the start node. We first generate a search space around the start node and calculate the least cost path to any node within that search space using the normalized gradient cost function. As the user continues to move the mouse, the current shortest path is generated based on the backpointers stored in each node. Eventually, the user will come to a node that is out of the search space. When this occurs, the search space is expanded and the graph is updated using normalized gradient cost function. Then the transient path is retrieved (connect the two end points to form a close contour) and sent to the level

FIG. 5. Flow chart of the LLS algorithm.

set algorithm as the initialization to compute the level set distance. The result is then factored into the hybrid cost function [Eq. (9)] for another graph update to determine the least cost path. The process is repeated until the complete contour is drawn.

II.C.4. Control parameters

There are three user defined parameters in the LLS: search expansion range, level set weight, and level set iteration number. Search expansion range defines the range to expand the search space each time a new node enters the search space. We used 20 pixels as the default value, which is sufficient in most cases since users can usually place the mouse within 20 pixels of the true boundary. Level set weight defines the weight of level set distance component in the hybrid cost function [Eq. (9)]. The level set iteration number defines the number of iterations the Laplacian level set is run to compute the level set distance. A rule of thumb is that the level set iteration number should equal twice the search expansion range. We will evaluate the effect of the control parameters on the performance of the LLS in Sec. III A 1. Based on the evaluation, we chose 0.3 as the default value for the level set weight and 40 as the default value for level set iteration number. Our evaluations indicate that the LLS is not sensitive to control parameter choice. We have successfully applied the default parameters in a large range of images and situations.

II.D. Advantages of LLS

LLS is a hybrid method of two distinct segmentation methods, livewire and level set. It has advantages over both classic livewire (CLW) and traditional level set (TLS) methods, especially for noisy, low contrast images such as medical images.

II.D.1. Advantages over classic livewire method

The hybrid cost function in the LLS has a factor of the level set distance. The curvature term in the speed function of the level set tends to preserve the smoothness of a contour. In this sense, the contour generated by the LLS tends to be smoother than that by the classic livewire. Furthermore, since the level set method is robust to contour discontinuity, the LLS can overcome small gaps and bumps along the boundary. Figure 6 shows an example of the LLS versus the classic livewire segmentation on a synthetic image of a noisy circle with gaps and bumps. It is noted that the LLS result is generally smoother and overcomes the gap and bump along the border.

II.D.2. Advantages over traditional level set method

The traditional Laplacian level set method requires a fairly good initial contour to get a reasonable result. This restriction is alleviated in the LLS since the first graph update using normalized gradient cost function can bring the contour close to the destination and the second graph update further refine the segmentation. Furthermore, the traditional

FIG. 6. Classic livewire vs LLS. Left: result from classic livewire; Right: result from LLS. Dark contours are the initial values, light contours are the segmentations. The LLS results in smoother boundaries and overcomes the gap and bump along the border. Contrast: 50, noise: 20, CNR: 2.5, blurring: 3.

level set tends to oversmooth the contour, which makes it imprecise when segmenting objects with sharp corners. Figure 7 shows two examples of the LLS versus the traditional level set with two different initial contours on a synthetic image of a noisy and blurry square. We see that the traditional level set converges to different segmentations using two different initial contours, and it fails to segment the corners of the square. The level set is a time consuming operation when many iterations are applied to a large image. However in the LLS, first, the level set distance is only computed in a small neighborhood around the transient path; second, only a small number of level set iterations is necessary since the initial contour is already close to the true boundary after the first graph update. Taking into account both factors, the LLS is much more efficient and accurate than the traditional level set.

III. EXPERIMENTS AND RESULTS

We used simulations on artificially constructed images, as well as visual inspection of results on actual clinical images, to quantitatively and qualitatively compare the performance between the CLW, the TLS, and our hybrid LLS method. By using synthetic images we can extract quantitative differences and systematically evaluate the overall performance. Section III A details the systematic performance comparison. Section III B describes the visual comparison using actual clinical images.

FIG. 7. Traditional level set vs LLS. (a) and (b) results from the traditional level set (different initial contours); (c) and (d) results from the LLS. Dark contours are the initial values, light contours are the segmentations. The traditional level set converges to different segmentations (DSC=0.921) using different initial contours and fails to segment the corner. The LLS gets consistent segmentation of the square (DSC=0.984). Contrast: 50, noise: 50, CNR: 1, blurring: 3.

III.A. Quantitative evaluation using synthetic images

We produced two types of synthetic images, one is a circle of 100 pixels in diameter (Fig. 6), and the other is a square with 100 pixels in size (Fig. 7). Gaussian white noise and Gaussian blurring were added to the images. Since medical images are generally noisy and do not have distinct edges, we evaluate our algorithms under different contrast to noise ratio (CNR) and blurring condition in the synthetic images. The Gaussian white noise was added at different magnitudes. The blurring effect was simulated by convolving the image with a Gaussian kernel at different variances. Both noise and blurring artifacts can be turned on or off separately so that different combinations can be applied to generate the synthetic images. Three individuals were selected to manually trace contours around the synthetic circle and square. Each individual drew five initial contours on each object at different time points. These contours were saved and used as the initializations for our simulations.

The evaluation is performed using three statistical metrics: dice similarity coefficient (DSC), mean contour distance (Dist), and smoothness index (SI). We categorize the pixels in four categories according to their presence in the ground-truth (GT) and the computer segmentation (CS). A pixel is treated as a true positive (TP) if it is present in both GT and CS; as a true negative if it is not present in both GT and CS; as a false negative (FN) if it is present in GT but not in CS; and as a false positive (FP) if it is present in CS but not GT. DSC is a statistical measure of spatial overlap between GT and CS, and is defined as,

$$DSC = \frac{2N_{TP}}{2N_{TP} + N_{FP} + N_{FN}}. \quad (10)$$

Here, N_C represents the number of pixels in the category C (TP, FP, or FN). We also compute the mean contour distance to evaluate the overall performance

$$Dist(X, Y) = \text{mean} \left\{ \min_{x \in X} [d(x, y)] \right\}. \quad (11)$$

Here, X is the segmented contour, Y is the ground truth contour, and $d(x, y)$ defines the Euclidean distance between two points x and y . The smoothness index is to characterize the smoothness of the contour. Intuitively a smooth contour should have continuously varying tangent along the contour.²⁵ Therefore, we define the smoothness index by summing up the change of tangent directions along the contour

$$SI = \frac{1}{N} \sum_{i=1..N} T_i \circ T_{i+1}. \quad (12)$$

Here, N is the number of point on the contour, T_i is the tangent direction at point i , and $\circ$ is the dot product between two vectors. SI is value between -1 and 1 . The higher SI is, the smoother the contour is.

For each imaging condition (CNR and blurring), the mean and standard deviation of 30 simulations (two objects, three users, and five initial contours) were computed. The average and standard deviation of the metrics in the 30 simulations

FIG. 8. Evaluation of control parameters. (a) Level set weight; (b) level set iteration number. The big data points are for default parameters.

are used to statistically analyze the performance. We also record the amount of time required to complete the segmentation. Figure 6 is a synthetic circle image with CNR=2.5 and blurring=3, and Fig. 7 is a synthetic square image with CNR=1 and blurring=3.

III.A.1. Evaluation of control parameters

As mentioned in Sec. II C, the level set weight (w_l) in the hybrid cost function and level set iteration number may affect the performance of the LLS. In this experiment, we vary the control parameters and conduct statistical evaluations on the synthetic images using all initial contours. Figure 8(a) plots the mean of Dist and DSC of all evaluations when varying the parameter of level set weight. Figure 8(b) plots the mean of Dist and DSC of all evaluations when varying the parameter of level set iteration number. The big data points on the plots are the default values for parameter selection (0.3 for level set weight and 40 for level set iteration number). The LLS achieves the best performance at level set weight=0.3, but varies only slightly between 0.2 and 0.4 [Fig. 8(a)]. We chose default level set iteration number at 40 since the plot is virtually flat after level set iteration=40 [Fig. 8(b)] and the running time is substantially longer for more level set iterations. In summary, the results indicate that the LLS is robust to parameter selection.

III.A.2. Comparison over image noise

The comparison of three methods over image noise is showed in Fig. 9. In this experiment, we fixed the image blurring at 3 and varied the CNR from 2 to 0.5 (contrast fixed at 50, and noise varied from 25 to 100). The traditional level set gave the worst segmentation accuracy at all noise levels since the initial contours were not close enough to the result. The Classic livewire and LLS had similar performances on low noise images, while the LLS is superior to

FIG. 9. Comparison over image noise. Left: DSC metric; right: Dist metric.

FIG. 10. Comparison over image blurring. Left: DSC metric; right: Dist metric.

the classic livewire on noisy data. The LLS maintains good performance until the CNR is 0.75, the performance drops dramatically when the CNR is 0.5 (noise is twice as strong as the contrast).

III.A.3. Comparison over image blurring

The comparison over image blurring is shown in Fig. 10. In this experiment, we fixed the image CNR at 1 and varied the blurring from 1 to 5. Again, traditional level set demonstrated the worst performance, while the LLS performed slightly better than the classic livewire. It also showed that more image blurring did not adversely affect the performance. All methods achieved the best performance at image blurring=2, which indicated appropriate image smoothing actually helped the segmentation. The results also demonstrate that the LLS is not sensitive to image blurring. It achieves comparable performance at all blurring levels.

III.A.4. Running time comparison

The running time is measured as the duration time from the first point in the prerecorded initial contour sent to the algorithm until the end of the simulation (no lapse between points). The running time comparison is illustrated in Fig. 11. As expected, the traditional level set took the longest time, while the classic livewire is the fastest. Although the LLS took more than three times that of the classic livewire, its mean running time (4.3 s) is still sufficient for interactive segmentation.

III.A.5. Reproducibility comparison

We also compared the reproducibility among three users using the three different methods. Each user conducted five separate segmentation tests on the synthetic images. Table I

FIG. 11. Running time comparison.

TABLE I. Intra-user reproducibility comparison. Numbers in each cell are the standard deviation of DSC and Dist (pixels) metrics of all tests conducted by one user.

	TLS	CLW	LLS
User1	0.049,1.67	0.019,0.76	0.016,0.49
User2	0.022,0.64	0.016,0.75	0.014,0.63
User3	0.032,1.65	0.023,1.05	0.017,0.69

shows the intra-user reproducibility, where each cell represents the standard deviation of DSC and Dist metrics of all tests conducted by one user. The LLS demonstrates best intra-user reproducibility. Table II shows the inter-user reproducibility, where each cell represents the average difference of DSC and Dist metrics of tests conducted between two users. The LLS again demonstrates best inter-user reproducibility.

III.A.6. Smoothness comparison

The LLS factors in the level set distance to regularize the smoothness of the contour. In this experiment we evaluated the smoothness index [Eq. (12)] of the resulting contours from, the TLS, CLW, and LLS. We used both synthetic circle and square images, and the fixed CNR=1 and blurring=3. Table III lists the smoothness index for the three methods. It indicates that the TLS does generate the smoothest contours and the LLS improves the smoothness substantially compared to the CLW.

III.B. Qualitative evaluation using clinical images

We used three types of clinical images to qualitatively evaluate the three segmentation techniques, magnetoresistance (MR) scans of uterine fibroids and knee cartilage and CT scan of livers. The examples we chose are some of the most challenging segmentation tasks in medical image processing. Similar to the simulation experiments, we used the same hand drawn contours as initial contours to the three algorithms. Figure 12 shows the visual comparison of the segmentation methods on clinical data.

Uterine fibroids are the most common pelvic tumors in females. Accurate volume measurement is essential to characterize the growth rate and to develop therapeutics treatment. The appearances of fibroids on MR images are very complex and inhomogeneous (first row in Fig. 12). The boundary is blurry and often has gaps. Noisy textures may at times create internal edges. From the results, the LLS ap-

TABLE II. Inter-user reproducibility comparison. Numbers in each cell are the average difference of DSC and Dist (pixels) metrics of all tests conducted between two users.

	TLS	CLW	LLS
User1 vs user2	0.078,2.84	0.019,0.78	0.016,0.58
User2 vs user3	0.065,2.39	0.022,1.03	0.019,0.79
User1 vs user3	0.046,2.43	0.027,1.36	0.021,0.93

TABLE III. Smoothness comparison using synthetic images.

	TLS	CLW	LLS
Circle image	0.914 ± 0.004	0.852 ± 0.021	0.904 ± 0.007
Square image	0.956 ± 0.006	0.918 ± 0.014	0.953 ± 0.007

pears to capture the most accurate fibroid boundary and its contour is smoother than that of the CLW. The TLS misses the hump in the lower left corner and the CLW oversegments the region in the lower right corner.

Knee cartilage is a thin structure around the knee joint. Determining the volume and thickness of cartilage is important in evaluating the cartilage damage. The segmentation result of the TLS is a failure, where the contour is converged to stronger edges and collapses at times. The contour generated by the CLW is not smooth, while the LLS gives the most visually accurate segmentation.

The size of liver is an important biomarker for a number of diseases. In the CT images, the contrast between the liver and external organs is at times extremely low leading to weakly defined boundaries. The results shown in the third row of Fig. 12 demonstrates that the TLS fails to segment the sharp corners of the liver on the right side and oversegments the liver to include part of the kidney (lower-left corner). The CLW's result is jittery at places and also does not segment the sharp corner correctly. Again, the LLS's segmentation is more favorable than that of both the TLS and CLW.

The smoothness indices of the contours generated by the three methods on the real images are listed in Table IV. It shows that all three methods generate much smoother contours than the initial ones traced by a user. In general, the LLS is smoother than the CLW. In the case of knee cartilage, the smoothness index of the LLS is even better than that of the TLS since the TLS's segmentation is a failure. From visual inspection in these three applications, we see notice-

TABLE IV. Smoothness index on clinical images.

	Initial	TLS	CLW	LLS
Uterine fibroid	0.511	0.824	0.772	0.805
Knee cartilage	0.636	0.817	0.788	0.823
Liver	0.661	0.865	0.811	0.848

able differences between the contours when the region of interest is inhomogeneous and the boundary is less defined. As boundaries become more defined and less inhomogeneous, the differences among the three methods diminish. The segmentation results were shown to two radiologists and they both agreed that the LLS provides the most favorable segmentation results.

IV. DISCUSSIONS

The LLS is a useful tool for medical image segmentation, where the anatomy boundaries are generally less distinct and noise and artifact are often presented. The LLS overcomes these imagery obstacles by marrying livewire and level set techniques. The medical staffs (radiologists and technologists) are good at recognizing the anatomical objects. However, they are not very good at placing the computer mouse at the exact locations, or tracing a smooth contour. Aided by the the LLS tool, it is much easier for the user to define a smooth (see Tables III and IV) and precise (see Figs. 8–10) boundary. It also reduces the variability (see Tables I and II) when the segmentation task is performed by multiple users.

There are some limitations in this investigation. The validation was mainly focused on the accuracy and robustness of the method. That was the reason we used the same initial contours when comparing the LLS with classic livewire and traditional level sets. However, since this is an interactive technique, it allows the user to make adjustments in positioning the mouse. A complete usability evaluation similar to the one proposed in Ref. 5 may be necessary to fully evaluate the advantage of the LLS. Similar to livewire, the LLS is currently a two-dimensional technique. The ideas in graph cut⁴ can be adopted in building a 3D graph. An effort is currently under way to extend the LLS to 3D images.

Our results showed that the LLS is more accurate and less sensitive to initial values than the traditional Laplacian level set method. However, recent advances in level set techniques such as those in Refs. 17–23 have improved the performance of level set tremendously. The main advantage of our method over those sophisticated segmentation methods is the speed. Our method is interactive and can be run in real time.

In conclusion, we proposed a hybrid segmentation method that takes advantage of both the classic livewire and level set methods. The LLS inherits the interactivity of the livewire, which can be executed simultaneously as the user traces the computer mouse over the image. It also inherits the robustness of the level set method, which allows it to overcome noisy and discontinuous boundaries. The advantage is achieved by two iterations of graph updates. In the first iteration, the original livewire graph update brings the contour

FIG. 12. Evaluation on clinical images. First row: T2 weighted MR images of uterine fibroid, second row: T1 weighted MR images of knee cartilage, third row: CT images of liver. (a) clinical images, (b) segmentation results of traditional level set, (c) segmentation results of classic livewire, (d) segmentation results of LLS. Dark contours: hand drawn initial contours, light contours: segmented contours.

close to the real boundary. In the second iteration, the level set distance is computed using the result of the first iteration as initialization. The level set distance is then factored into a hybrid cost function to obtain a robust and smooth segmentation. Both systematic performance comparison using synthetic images and visual inspection using clinical data demonstrated that the LLS is superior to both classic livewire and traditional level set methods.

ACKNOWLEDGMENTS

The authors thank Dr. Ronald Summers for support of the project and critical reviews of the article. This research was supported by the Intramural Research Program of the National Institutes of Health, Warren G. Magnuson Clinical Center.

^{a)}Electronic mail: jyao@cc.nih.gov

- ¹A. X. Falcao *et al.*, "User-steered image segmentation paradigms: Live wire and live lane," *Graph. Models Image Process.* **60**, 233–260 (1998).
- ²M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.* **1**, 321–331 (1988).
- ³T. F. Cootes *et al.*, "Active shape models—Their training and application," *Comput. Vis. Image Underst.* **61**, 38–59 (1995).
- ⁴Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 1124–1137 (2004).
- ⁵Y. Li *et al.*, "Lazy snapping," *ACM Trans. Graphics* **23**, 303–308 (2004).
- ⁶E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," *ACM Computer Graphics (SIGGRAPH)*, **22**, 191–198 (1995).
- ⁷E. N. Mortensen *et al.*, "Adaptive boundary detection using livewire two-dimensional dynamic programming," *IEEE Computers in Cardiology*, Durham, North Carolina, 1992 (unpublished).
- ⁸J. K. Udupa, S. Samarasekera, and W. A. Barrett, "Boundary detection via dynamic programming," *Visualization and Biomedical Computing*, Chapel Hill, North Carolina, 1992 (unpublished).
- ⁹D. Stalling and H.-C. Hege, "Intelligent scissors for medical image segmentation," *Digitale Bildverarbeitung fuer die Medizin* 32–36 (1996).
- ¹⁰V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *IEEE*

International Conference on Computer Vision, Cambridge, Massachusetts, 1995 (unpublished).

- ¹¹J. A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge University Press, Cambridge, 1999).
- ¹²J.-P. Guyon *et al.*, "VETOT, volume estimation and tracking over time: Framework and validation," *MICCAI*, Montreal, Canada, 2003 (unpublished).
- ¹³J. Liang, T. McInerney, and D. Terzopoulos, "United Snakes," *Med. Image Anal.* **10**, 213–233 (2006).
- ¹⁴H. W. Kang and S. Y. Shin, "Enhanced lane: Interactive image segmentation by incremental path map construction," *J. Rubber Res.* **64**, 282–303 (2002).
- ¹⁵A. Souza *et al.*, *Iterative Live Wire and Live Snake: New User-Steered 3D Image Segmentation Paradigms*, (SPIE Medical Imaging, San Diego, CA, 2006).
- ¹⁶K. Lu and W. E. Higgins, *Improved 3D Live-Wire Method with Application to 3D CT Chest Image Analysis* (SPIE Medical Imaging, San Diego, CA, 2006).
- ¹⁷D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *Int. J. Comput. Vis.* **72**, 195–215 (2007).
- ¹⁸A. Tsai, A. J. Yezzi, and A. S. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Trans. Med. Imaging* **22**, 137–154 (2001).
- ¹⁹J. Kim *et al.*, "A nonparametric statistical method for image segmentation using information theory and curve evolution," *IEEE Trans. Image Process.* **14**, 1486–1502 (2005).
- ²⁰S. P. Awate *et al.*, "Adaptive Markov modeling for mutual-information-based, unsupervised MRI brain-tissue classification," *Med. Image Anal.* **10**, 726–739 (2006).
- ²¹M. Rousson, T. Brox, and R. Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," *CVPR*, 2003 (unpublished).
- ²²G. Doretto *et al.*, "Dynamic texture segmentation," *International Conference on Computer Vision*, 2003 (unpublished).
- ²³S. P. Awate, T. Tasdizen, and R. T. Whitaker, "Unsupervised texture segmentation with nonparametric neighborhood statistics," *ECCV*, 2006 (unpublished).
- ²⁴L. Ibanez and W. Schroeder, *ITK Software Guide Kitware, Inc.*, 2003.
- ²⁵A. G. Constantine and P. Hall, "Characterizing surface smoothness via estimation of effective fractal dimension," *J. R. Stat. Soc.* **56**, 97–113 (1994).