01 Jul 2008

Live Wire Segmentation Tool for Osteophyte Detection in Lumbar Spine X-Ray Images

Santosh Seetharaman

R. Joe Stanley
Missouri University of Science and Technology, stanleyj@mst.edu

Soumya De

Sameer Antani

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1704

Recommended Citation

S. Seetharaman et al., "Live Wire Segmentation Tool for Osteophyte Detection in Lumbar Spine X-Ray Images," *Proceedings of the First International Conference on Emerging Trends in Engineering and Technology (2008, Nagpur, Maharashtra, India)*, pp. 715-720, Institute of Electrical and Electronics Engineers (IEEE), Jul 2008.
The definitive version is available at https://doi.org/10.1109/ICETET.2008.40

Live Wire Segmentation Tool for Osteophyte Detection in Lumbar Spine X-Ray Images

Santosh Seetharaman[a], R. Joe Stanley[a], Soumya De[a],
Sameer Antani[b], Rodney Long[b], George Thoma[b]

[a]Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO
[b]Communications Engineering Branch, National Library of Medicine, Bethesda, MD

Abstract

Computer-assisted vertebra segmentation in x-ray images is a challenging problem. Inter-subject variability and the generally poor contrast of digitized radiograph images contribute to the segmentation difficulty. In this paper, a semi-automated live wire approach is investigated for vertebrae segmentation. The live wire approach integrates initially selected user points with dynamic programming to generate a closed vertebra boundary. In order to assess the degree to which vertebra features are conserved using the live wire technique, convex hull-based features to characterize anterior osteophytes in lumbar vertebrae are determined for live wire and manually segmented vertebrae. Anterior osteophyte discrimination was performed over 405 lumbar vertebrae, 204 abnormal vertebrae with anterior osteophytes and 201 normal vertebrae. A leave-one-out standard back propagation neural network was used for vertebrae segmentation. Experimental results show that manual segmentation yielded slightly better discrimination results than the live wire technique.

Keywords: osteoarthritis, osteophyte, segmentation, live wire, neural networks, image processing, lumbar spine, x-ray

1. Introduction

Computer-assisted segmentation of vertebrae in cervical and lumbar spine x-ray images is a challenging problem. Several techniques have been investigated for vertebra segmentation in digitized radiographs, including Active Shape Models (ASM) [1-3], the Hough Transform [4], and edge-based object location and border detection. For ASM and similar types of approaches, the user provides an initial boundary. The limitations of approaches where the user provides an initial approximation of the object boundary include: 1) the user cannot determine the appearance of the final boundary until the algorithm converges, 2) the approach is sensitive to noise in the convergence of the algorithm to the final boundary, 3) user interactions with intermediate boundaries obtained from the algorithm results require the algorithm to be re-executed. Factors making computer-based segmentation difficult include inter-subject variability and the generally poor contrast of digitized radiograph images. Intelligent segmentation, also referred to as live wire, tools provide the capability to allow the user and the segmentation process to collaborate dynamically during object segmentation. The user selects objects of interest based on actions with the mouse over the image data. Live wire provides an interactive process where the user selects points along the object boundary and the computer-based segmentation algorithm determines an intermediate boundary connecting the selected points. The process is repeated until a closed boundary is generated. Editing and training capabilities can be used to modify the semi-automatically determined boundaries as well as the capability to adapt to user and inter-image variability.

The Lister Hill National Center for Biomedical Communications, a research division of the National Library of Medicine (NLM) has built a biomedical information resource consisting of digitized x-ray images and associated textual data from national health surveys. This resource, the Web-based Medical Information Retrieval System, is capable of retrieving images based on image characteristics, either alone or in conjunction with text descriptions associated with the images. There are around 17,000 x-ray images available for researchers interested in osteoarthritis, osteoporosis, degenerative disc disease etc. In this research and as part of the NLM initiative, a semi automated live wire technique is investigated for lumbar vertebrae segmentation.

2. Live wire algorithm description

In this research, the live wire algorithm was extended for application to lumbar vertebra segmentation. A graphical user interface (GUI) was implemented to facilitate the application of the live wire algorithm, with all programs implemented using Matlab 6.1. The description of each step of the live wire algorithm is given in the following steps.

978-0-7695-3267-7 2008

DOI 10.1109/ICETET.2008.40

2.1. Region of interest selection

The first step in the algorithm is for the user to select a region of interest (ROI) containing the vertebra to be segmented. The user is prompted to choose four corner points for the region of interest. The minimum and maximum row and column positions are determined from the user selected points to create a bounding box for the region of interest. Let *I* denote the *Y* row and *X* column gray level region of interest obtained from the original x-ray image. Fig. 1 presents an example region of interest obtained from a lumbar x-ray image.

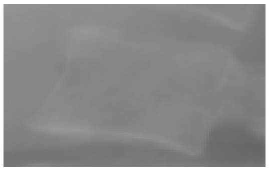


Fig. 1: Example of region of interest extracted from a lumbar x-ray image.

2.2. Image enhancement and initial point selection

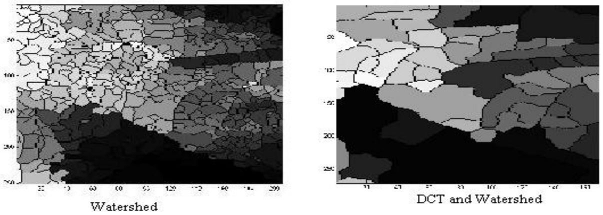
After bounding box determination, the region of interest is zoomed by a magnification factor of two to assist the user in the semi-automated segmentation process. For this live wire implementation, the region of interest gray level image *I* is low-pass filtered using the Discrete Cosine Transform (DCT). Only the low frequency portion of the image is extracted from the DCT transform frequency domain. The lower frequency part of the image lies in the upper left corner of the DCT of the image. So, the DCT coefficients that lie within the rectangular region $1 \leq u \leq U$ and $1 \leq j \leq J$, where $U = Y/2$ and $J = X/2$ are extracted for a lower resolution of the original ROI image which has *Y* rows and *X* columns. The DCT values outside this rectangular region are set to 0 in the DCT matrix. The inverse DCT transform was applied to the resulting low pass filtered DCT coefficient image to generate an enhanced image *E*. After DCT-based enhancement over the region of interest, a message prompts the user to select points along the vertebrae boundary for segmentation. For the lumbar x-ray image set examined in this research, approximately 40 initial points were selected around each vertebra boundary.

2.3. Watershed transform

The watershed algorithm is applied to the filtered DCT image *E* to obtain contours used to determine the vertebra boundary. The watershed algorithm implementation presented in [6] was used. The difference between the ordinary watershed and the watershed after DCT based filtering is shown in Fig. 2 for an original gray level region of interest. Fig. 2(a) gives the output of the watershed algorithm using only the original grayscale image, and Fig. 2(b) presents the output of the watershed algorithm after DCT enhancement. Observing Fig. 2, it can be seen that the watershed regions are much less noisy in the DCT enhanced image (Fig. 2(b)) than in the original image (Fig. 2(a)). The watershed representation for the original gray level image in Fig. 1(a) has numerous small regions, which add complexity to the process for determining the vertebra boundary. Let *W* denote the resulting watershed image with *Y* rows and *X* columns. Then, *Z* denotes the watershed boundaries such that

$$Z(x,y) = \begin{cases} 1, & W(x,y) = 0 \\ 0, & W(x,y) > 0 \end{cases} \quad (1)$$

where, $1 \leq x \leq X$ and $1 \leq y \leq Y$.



(a) **(b)**

Fig. 2: Example of watershed algorithm applied to original gray level and corresponding DCT enhanced images. (a) Watershed algorithm applied to original grayscale image. (b) Watershed algorithm applied to DCT enhanced image.

2.4. Dynamic programming approach to determine vertebra boundary

The watershed transform gives a value of zero to edges and higher values to other regions. The resultant image after watershed transform is considered as a graph. Each path in the graph has to be assigned a cost and the image is converted to a cost matrix. Now, finding the optimal path is just a problem of finding the lowest cost path pixel-wise.

The live wire algorithm uses dynamic programming which provides a cost minimization problem between any two given points in the image. A cost is associated with every pixel. A path is two-dimensional between any two points in the image. The dynamic programming implementation used in this research applies the algorithm developed by Mortensen et al. [5], which uses the following constraints for 4-connected and 8-connected neighbors for each pixel along the path between source pixel *p* and destination pixel *q*. Any pixel in the image is surrounded by 8 neighboring pixels. Each pixel has an 8-connected neighborhood, except for the image boundary cases. In order to traverse a path from the source to the destination in a two-dimensional sense, the path must be able to go through the diagonal pixels also. The 4-connected neighborhood, namely the left, right, top, bottom are the closest to any given pixel. So, a path should go to the 4-connected neighbors instead of the 8-connected neighbors if both have the same cost. In other words, the 4-connected neighbors are preferred to an 8-connected neighbor. But if the cost through the 4-connected neighbor is above a certain limit, then it is better to route the path through the 8-connected neighbors. While routing a path from point *p* to point *q*, it is always necessary to consider the 4-connected and the 8-connected neighborhoods. Since 4 connected neighbors are preferred to 8-connected neighbors, a higher cost is assigned to 8-connected neighbors than 4-connected neighbors. The weighting function *w(p,q)* used for a neighbor q to a pixel *p* is

$$w(p,q) = 1; \; if \;\; L_x(p,q) = 0 \vee L_y(p,q) = 0 \qquad (2).$$

$$w(p,q) = \sqrt{2}; \; if \;\; L_x(p,q) \neq 0 \wedge L_y(p,q) \neq 0 \qquad (3).$$

where, L_x and L_y are the horizontal and vertical components of the link vector *L*, *w(p,q)* is the weighting function assigned for the cost calculation between the pixels p and q, $\wedge$ represents logical AND operation and $\vee$ represents logical OR operation. *L* is the normalized bidirectional link or unit edge vector between pixels *p* and *q* and simply computes the direction of the link between *p* and *q*. Links can be vertical, horizontal or diagonal. Equation (2) refers to the 4-connected neighbors, and Equation (3) refers to the diagonal neighbors and equation. The initial user selected points (see section 2.1) are stored consecutively. For dynamic programming, the initial two user chosen points are processed first. If *p* is the starting pixel, *q* is the ending pixel (where *p* and *q* are consecutive user entered points). The least cost path between the two points is computed using the watershed boundaries as a guide for path determination. The least cost path between *p* and *q* results in a partial boundary segment between and including the initial two points. This procedure is repeated until a closed optimal path is obtained using the user selected input points.

The algorithm to determine the least cost path is presented in detail in [5] and is overviewed here. Let *s* be the start or source point for which the costs to all neighboring pixels are to be found. The local cost function *c(p,q)* is the cost assigned for the path between pixels *p* and *q* which is initialized to 0. Let *A* represent the list of active pixels or pixels which are being considered for the optimal path between *p* and *q*. Let N_4 contain the list of all 4-connected neighbors of pixel *p*, and N_8 contain the list of all the 8-connected neighbors of pixel *p*. Then, *N* consists of all pixels in both N_4 and N_8. Let *K* contains all the pixels positions within a radius of 20 pixels from starting pixel *p*. Let *R(p)* contain the cumulative link cost of all the neighboring pixels to *p*. Let *D* maintains the indices of all the active pixels, and D_{temp} is used for storing indices temporarily. *w* has the associated weights for all of the neighboring pixels to *p*. Let *F* contain the final list of pixels which encompass the lowest cost path. For the algorithm used, $\tau = 2\varepsilon$ was experimentally determined, where ε is the Euclidean distance between *p* and *q*. Using this algorithm, the optimal path was found by searching the graph for the lowest cost path in the image. The algorithm first used the entire image for the graph search. It calculates the shortest path for all the pixels in the image. With larger images, the considerable computations are required, making the algorithm non-suitable for real time usage. Accordingly, the graph search was confined to a radius of 20 pixels from the user specified point, as specified with the parameter *K*. Note that *K* was empirically determined based on the quality of the vertebra boundary paths obtained and the goal of reducing the computation time required for the dynamic programming algorithm.

2.5. Convex hull postprocessing

The final step to obtain the vertebra boundary is to determine the convex hull from the set of closed boundary points *F,* obtained from the dynamic programming-based piecewise paths based on the user selected points. The convex hull of a data set is the smallest convex region that contains the data set of points [5]. The convex hull was applied to *F* for boundary smoothing. Let *S* denote the resulting convex hull output, representing the final set of points for the vertebra boundary. The convex hull algorithm developed by Barber et al. was used in this research [6]. Fig. 3 presents a vertebra image example showing the vertebra boundary before the convex hull (b) and after the convex hull is determined (c).

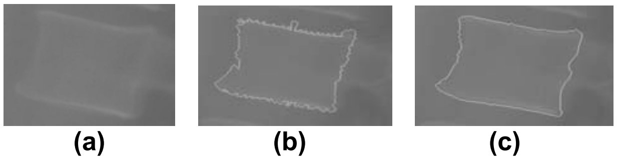


Fig. 3: Example of convex hull for boundary smoothing. (a) Original grayscale image. (b) Live wire segmentation before convex hull. (c) After applying convex hull.

2.6. GUI implementation

The GUI implementation allows the user to select the region of interest for segmenting an individual vertebra in an x-ray image. The GUI then allows the user to select points along the vertebra for determining the vertebra boundary based on the algorithm presented in sections 2.4-2.5. Finally, the GUI provides the user the capability to edit the final boundary by removing portions of the vertebra boundary that are incorrect and selecting additional points in regions where the vertebra boundary is incorrect and re-executing the algorithm in sections 2.4-2.5 to generate the revised vertebra boundary. In order to remove portions of the closed boundary, the user selects the Remove button. The user is prompted to select the upper left and lower right positions of a rectangular region that is removed from the displayed boundary. The boundary is updated by connecting the points along the rectangular box on the sides closest to the centroid of the original closed boundary vertebra to the end points of the open vertebra boundary. In order to add additional points to update the vertebra boundary, the user selects the Edit button. The user is prompted to insert two additional points along the desired boundary. A list of four sequential points is generated based on determining the closest boundary point to each of the two user selected points. The updated boundary is found by re-executing the dynamic programming and convex hull methods for piecewise-segment determination in sections 2.4-2.5. The resulting closed boundary is filled, and any intermediate segments found inside of the vertebra are removed. Finally, a user may save a vertebra boundary by selecting the Save button.

3. Anterior osteophyte feature extraction

In previous research, manually segmented vertebrae were used for the determination of on vertebral distortion along the anterior boundary as an indicator of osteophytes. In order to evaluate the capability of the live wire segmented vertebrae to preserve key vertebrae features such as the presence of anterior osteophytes, the convex hull-based features were computed over a lumbar vertebrae data set segmented manually and using the live wire technique. The convex hull-based features are presented in detail in [5] and are overviewed here. The features are computed for a vertebra segmented from the gray level region of interest I, where $1 \leq x \leq X$ and $1 \leq y \leq Y$. Let V denote a lumbar vertebra within an x-ray image with area A_v such that

$$V = \begin{cases} 1 \text{ if (x,y) is on or inside the vertebra boundary} \\ 0 \text{ otherwise} \end{cases} \quad (4).$$

The convex hull for V is determined using the quick convex hull algorithm [6]. Let B denote the resulting filled convex hull for vertebra V such that

$$B = \begin{cases} 1 \text{ if (x,y) is on or inside the convex hull boundary for } V \\ 0 \text{ otherwise} \end{cases} \quad (5).$$

Let P denote the set of exclusive-or points between V and B such that $P(x,y) = V(x,y) \oplus B(x,y)$. P contains one or more connected regions for each concave vertebral side. Let Q denote the number of unique 8-connected components within P. The four features used for detecting anterior osteophytes include: 1) the ratio of the vertebral area to the convex hull area, 2) the ratio of the exclusive-or area to the convex hull area, 3) the ratio of the exclusive-or area on the vertebra's anterior side to the vertebra area and 4) the ratio of the area of the largest connected region from the exclusive-or regions on the anterior side of the vertebra to the vertebra area. These features are size invariant thus allowing direct comparison of different vertebrae. The first feature, which is the ratio of the vertebral area to the filled convex hull denoted as α, is defined as

$$\alpha = \frac{A_V}{A_C} \quad (6).$$

where A_V is the vertebral area and A_C is the convex hull area. The ratio of the exclusive-or area to the filled convex hull area is denoted as β and is defined as

$$\beta = \frac{\mathrm{A_E}}{\mathrm{A_C}} \quad (7).$$

In order to compute the final two exclusive-or-based features, the centroid location, $(x_{c,}\, y_c)$, is calculated for the vertebra V and the connected regions within P are labeled. Then, the original XY plane is mapped to a new X^1Y^1 plane for V, which is centered at the centroid location with the axes parallel to the original axes. Connected components in E that are on the right hand side of the X^1Y^1 plane and located on the posterior side of the vertebra are removed. Let G denote the number of connected components remaining on the anterior side. Let $H = \{h_1, h_2, ..., h_G\}$ refer to the set of connected components remaining with areas given by A_{h_i} for $1 \leq i \leq G$. The third feature, which is the ratio of the exclusive-or area on the vertebra's anterior side to vertebral area, is denoted as γ and is defined as

$$\gamma = \frac{\sum_{i=1}^{G} A_{h_i}}{A_v} \quad (8)$$

The final convex hull-based feature is based on finding the largest connected component on the anterior side. The connected components in P that are located on the anterior side of the vertebra are selected. Let b denote the number of connected components on the anterior side of the vertebra that are selected and $U = \{u_1, u_2, ..., u_b\}$ refers to the set of connected components remaining with areas given by A_{u_i} for $1 \leq i \leq b$. Then, the ratio of the largest connected region from the exclusive-or regions on the anterior side of the vertebra to the vertebral area is denoted as δ and is defined as

$$\delta = \frac{\max\limits_{i}(A_{u_i})}{A_V},\ 1 \le i \le b \quad (9).$$

These features were extracted for both the manually drawn and the live wire determined borders.

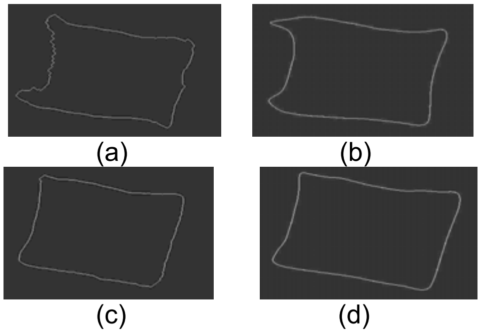
4. Experiments performed

A data set of lumbar spine x-ray images obtained from the NLM was used for comparing anterior osteophyte discrimination based on manually segmented vertebrae and live wire segmented vertebrae. For classification purposes, vertebrae are labeled normal or abnormal, where an abnormal vertebra has anterior osteophytes in one or both of the anterior side corners. A set of 405 images were selected for vertebrae segmentation of which 204 are abnormal and 201 images were normal. An expert radiologist manually labeled the upper and lower anterior and posterior side points, the top and bottom side center points, the anterior side midpoint, and, for a vertebra with osteophytes on the top or bottom anterior corners, the point marking the maximum extent of the osteophyte. For normal vertebrae only the upper and lower anterior and posterior side points, the top and bottom side center points, and the anterior midpoint are labeled. The four convex hull-based features [7] were computed for anterior osteophytes discrimination based on manually segmented vertebrae. In this research, the vertebrae were segmented using the live wire approach overviewed above, and the convex hull-based features were computed for anterior osteophytes discrimination. Approximately 40 points were selected for the live wire segmentation process. After obtaining the closed vertebra boundary from the live wire process, no editing operations were performed. Anterior osteophyte discrimination was performed on the manual and live wire segmented vertebrae to test the capability of the live wire algorithm to preserve key vertebra features.

Using the vertebrae segmented based on manual and live wire approaches, the convex hull-based features for anterior osteophytes discrimination were extracted over the lumbar image data set. Anterior osteophyte discrimination to classify the vertebrae into normal or abnormal was performed using a standard back propagation neural network, leave-one-out approach. The neural network architecture consisted of 4 input nodes and 2 hidden layers. The first hidden layer had 4 nodes and the second hidden layer had 2 nodes with one node for the output. Neural network training and testing was done using a leave-one-out approach for up to 15 epochs, with neural network training terminating when the root-mean-square error (RMSE) was less than 0.1.

5. Experimental results and discussion

Fig. 4 presents two examples of manual and live wire segmented vertebra. Fig. 4(a) and (b) correspond to the same vertebra and Fig. 4(c) and (d) also correspond to the same vertebra.


(a) (b)

(c) (d)

Fig. 4: Examples of live wire and manual segmented vertebrae. Live wire examples are in (a) and (c). Corresponding manual examples are in (b) and (d).

From Fig. 4, it can be observed that the manual vertebra boundary is smoother and less jagged than the corresponding live wire vertebra boundary. The leave-one-out neural network approach classified vertebrae into normal and abnormal based on features computed using the manually drawn and the live wire algorithm drawn borders [8].

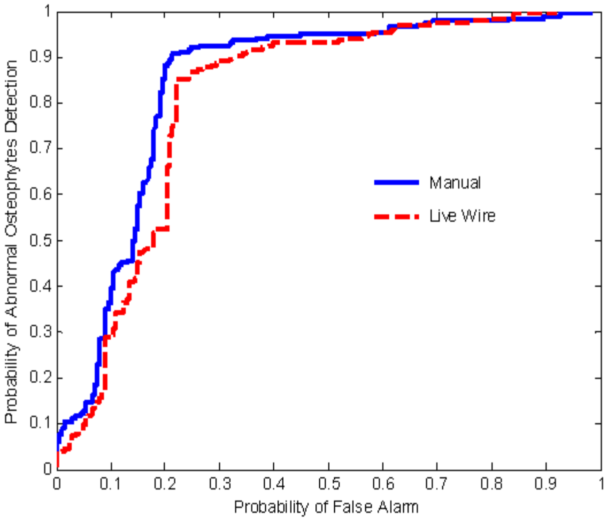


Fig. 5: ROC curve results for anterior osteophyte discrimination for manual and live wire segmented vertebrae.

The osteophytes discrimination results using live wire- and manually segmented vertebrae were compared using receiver operating characteristic (ROC) curves (Fig. 5).

From Fig. 5, the manually drawn borders have a better detection rate and a lower false alarm rate than the live wire method. The neural network results yielded 80% correct anterior osteophytes detection with 22% false alarms for the live wire segmented vertebrae and 18% false alarms for the manually segmented vertebrae. A false alarm refers to classifying a normal vertebra as containing anterior osteophytes. Overall, the experimental results showed that the live wire borders did a reasonable job of preserving anterior osteophytes for the convex hull-based features examined compared to manual borders, with manual borders providing improved anterior osteophytes discrimination. The reason for missing or false alarm may be due to any one or a combination of factors. First, in the live wire algorithm drawn borders, due to the rugged boundary, sometimes the abnormality is missed since there is not much difference between them. Second, the sharper, more obvious, osteophytes are detected more easily than the bigger and broader osteophytes cases. Third, the angle of curvature of the osteophyte(s) from the vertebra is important for anterior osteophytes discrimination based on the convex hull-based features examined. Concavities in the anterior and lower parts of the vertebrae boundary also represent abnormality. An absence of the concavity may be seen as lack of abnormality, and the presence of the concavity may be instrumental in identifying a vertebra as abnormal. Obviously, the greater the number of input points given by the user, the higher is the quality of the segmented vertebra. However, an increase in the number of points selected by the user increases the number of computations and thus takes a lot of valuable time for the user. A reasonably good number of points are required especially around the abnormality and curvature due to the presence of noise for proper detection. Due to the rugged irregular boundary (live wire drawn), sometimes the irregularities are classified as abnormalities. The perception of the vertebra boundary by the user plays an important role in the user-assisted vertebra boundary determination. The boundaries in the watershed image are used to guide the selection of the path between the user selected points. For the image data examined in this research, the watershed boundaries appear to be located in the center of the edges along the vertebra boundary, where strong edges are present. Accordingly, the live wire algorithm is sensitive to the user selected points for boundary determination, where the live wire algorithm yields higher quality segmented vertebrae when the user selects points as close to the actual vertebra boundary as possible.

6. Summary

This research investigated a semi-automated live wire technique for vertebra segmentation. Vertebra image segmentation was done using a combination of DCT enhancement, user selected points, watershed-based boundary determination, dynamic programming path searching, and convex hull post processing. The live wire was implemented using Matlab with an easy to use GUI interface with provisions for editing. The robustness of the live wire algorithm was analyzed relative to manual segmentation for identifying anterior osteophytes. Experimental results showed that the live wire borders did a reasonable job of preserving anterior osteophytes for the convex hull-based features examined compared to manual borders, with manual borders providing improved anterior osteophytes discrimination. Utilizing user interaction with the live wire algorithm should yield improvement in the quality of the segmented vertebrae and lead to better preservation of key vertebrae features.

7. Acknowledgment

This work was supported by NLM under contract number 467-MZ-601342 and the Intramural Research Program of the National Institutes of Health (NIH), NLM, and Lister Hill National Center for Biomedical Communications (LHNCBC).

8. References

1. Zamora, G.; Sari-Sarraf, H.; Mitra, S.; Long, R. Estimation of orientation and position of cervical vertebrae for segmentation with active shape models. Proc. SPIE 2001; 4315:393-403.
2. Smyth, P.P.; Taylor, C.C.; Adams, J.E. Automatic measurement of vertebral shape using active shape models. WACV'96. Proc. 3rd IEEE Workshop on Applications of Computer Vision 1996; 176-180.
3. Behiels, G.; Vandermeulen, D.; Suetens, P. Statistical shape model-based segmentation of digital X-ray images. Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis 2000; 61-68.
4. Tezmol, A.; Sari-Sarraf, H.; Mitra, S.; Long, R. Customized Hough transform for robust segmentation of cervical vertebrae from x-ray images. Proc. 5th IEEE Southwest Symposium on Image Analysis and Interpretation 2002; 224-228.
5. Mortensen, E.N.; Barrett, W.A. Interactive segmentation with intelligent scissors. Graphical Models and Image Processing 60:349-384: 1998.
6. Barber C.B.; Dobkin, D.P.; Huhdanpaa, H.T. The Quickhull algorithm for convex hulls. ACM Transactions on Mathematical Software 22(4):469-483: 1996.
7. Cherukuri, M.; Stanley, R.J.; Long, R.; Antani, S.; Thoma, G. Anterior osteophyte discrimination in lumbar vertebrae using size-invariant features. Computerized Medical Imaging and Graphics 28:99-108: 2004.
8. Zurada, J.M. Introduction to artificial neural systems. New York, N.Y.: West Publishing Co., 1992.