

SEGMENTATION OF ULTRASOUND LIVER IMAGES: AN AUTOMATIC APPROACH*

*Nualsawat Hiransakolwong*¹

*Kien A. Hua*¹

*Khanh Vu*²

*Piotr S. Windyga*¹

¹School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2362, USA.

²Computer Science Department
Oklahoma State University
Tulsa, OK 74106, USA.

E-mail: {nual, kienhua, pwindyga} @cs.ucf.edu

E-mail: vu@cs.okstate.edu

ABSTRACT

Segmentation of ultrasound liver images presents a unique challenge because these images contain strong speckle noise and attenuated artifacts. Most ultrasound image segmentation techniques focus on region growing or active contours. These are semi-automatic segmenting systems, in which seed points or initial contours have to be manually identified. In this paper, we propose a fully automatic segmentation system for ultrasound liver images. We apply the Peak-and-valley method to pixels scanned along the Hilbert curve, and propose a “windows adaptive threshold” procedure to further reduce noise from the images. After Otsu’s segmentation algorithm is applied to the images, a core area algorithm is employed to detect liver objects with the help of a feature knowledge base. We compared our method with other techniques and the manual segmentation method. The results indicate the accuracy of our system and our automatically segmented images contain less noise than the other methods.

KEYWORDS

Ultrasound images, Fully automatic segmentation, Windows adaptive threshold, Core area

1. INTRODUCTION

Ultrasound image segmentation is an important problem in medical image analysis and visualization. Because these images contain strong speckle noise and attenuation artifacts [3], it is difficult to automatically segment these images to detect interested objects in the correct position and orientation. Most image segmentation methods focus on *region growing* or *active contours*. For instance, to segment homogenous regions, the region growing method [3] first requires users to identify a seed point, using geographic priority and a multi-feature vector space of the seed point as criteria. The interference of speckle noise makes it unreliable to classify image pixels. The active contour

methods (e.g., [4]) are designed to find edges of a region whose color or other features are significantly different from those of the surrounding region. However, speckle noise makes clear edges difficult to detect. Furthermore, most active contour-based approaches are developed from the *snake* algorithm, which requires the user to identify an initial contour. Thus, both methods are only semi-automatic systems and suffer from speckle noise, which are present in ultrasound images.

We propose a full automatic segmentation system for ultrasound liver images. This solution can be divided into three steps. First, we filter noise by modifying the “peak-and-valley” to scan pixels along the Hilbert curve. Then we use the “Cubic Spline Interpolation” between local peaks and valleys to smooth the image. Second, we propose a “windows adaptive threshold” procedure, to further remove noise and to improve Otsu’s algorithm for obtaining the right segmentation threshold. Finally, we label distinct, disconnected objects and use the “core area” to detect the liver object with the help of a feature knowledge base. Our method experimented with a set of ultrasound liver images. We compared the quality of images segmented by our method with those by other techniques and the manual segmentation method. The results indicate the superior performance of our technique.

We review related techniques in Section 2. We describe our approach in Section 3. Our experimental study is discussed in Section 4. Finally we present concluding remarks in Section 5.

2. REVIEW BACKGROUND KNOWLEDGE

In this section, we review techniques that are adapted to support our system.

*This research is partially supported by a grant from Florida Hospital.

2.1 Peak-and-valley

Peak-and-valley is a non-linear filter method. It reduces impulsive noise while modifying the gray levels of the image as little as possible, resulting in maximum preservation of the original information [1]. The main idea of the peak-and-valley method is to substitute the intensity values of local peak pixels with the local maximum value between edges, and to fill the intensity values of local valley pixels with the local minimum value between edges (see Equation 1-3). From [1], the experimental results showed the peak-and-valley is better than the median filter.

The 1-D for k pixels peak-and-valley algorithm is as following:

$$P'(i+j) = \min(P(i-1), P(i+k)) \\ \text{if } P(i+j) < P(i-1) \text{ and } P(i+j) < P(i+k) \quad (1)$$

$$P'(i+j) = \max(P(i-1), P(i+k)) \\ \text{if } P(i+j) > P(i-1) \text{ and } P(i+j) > P(i+k) \quad (2)$$

$$P'(i+j) = P(i+j) \text{ else.} \quad (3) \\ \forall j = 0, 1, 2, \dots, k-1$$

Where $P(i)$ is the original intensity value at pixel i . $P'(i)$ is the new intensity value at pixel i . From equation (1), it represents a valley of k pixels. Equation (2) represents the case of the peak of k pixels and equation (3) represents neither a peak nor a valley.

From equations 1-3, the peak-and-valley and the median filter are identical only if k is equal to 1. Otherwise they are different.

2.2 Hilbert curve

The Hilbert curve is a space-filling curve that visits every point in a square grid with a size of 2×2 , 4×4 or any other power of 2. David Hilbert first described it in 1892, from [8][9].

2.3 Cubic Spline Interpolation Method

Since we use peak-and-valley, a local peak/valley becomes a flat peak/valley. To smooth peaks/valleys, we apply Cubic Spline Interpolation to the image between all local peaks and valleys. This makes a step line become a smooth curve. The Cubic Spline Interpolation method estimates the second derivatives at the points of reference (peaks/valleys) and uses these derivatives in the interpolation. Its algorithm and source code are in [6].

2.4 Otsu's Algorithm

The goal of thresholding is to convert a grayscale image into a binary image, separating an object's pixels from the background pixels. We select Otsu's methods for our dataset. Otsu's method is formulated as a discriminant analysis. Statistics are calculated for the two classes of intensity values (foreground and background) that are separated by an intensity

threshold. The criterion function is $\sigma_{B_i}^2 / \sigma_T^2$ for every intensity, $i = 0, \dots, I-1$, where $\sigma_{B_i}^2$ is the between-class variance and σ_T^2 is the total variance and $I = 256$, the maximum of the intensity gray level. The intensity that maximizes this function is the optimal threshold [5].

2.5 Relaxation

Relaxation is an iterative process that uses of the local thresholds. This initially classifies the segmentation, and this provides an estimate of the probability of being black or white at each pixel. For each pixel, we modify the segmentation and the estimate probability of being black or white based on the surrounding eight pixels. Redo until no further changes are seen in successive steps (see in [10]).

2.6 Labeling distinct disconnected objects

We use a labeling algorithm from [7] to label all the distinct disconnected objects.

3. OUR APPROACH

In this section, we first describe how we modify or use the techniques from Section 2, and propose new techniques to support our approach. Finally, we describe the entire segmentation algorithm.

3.1 Modified Peak-and-Valley Method

In the original peak-and-valley algorithm, pixels are scanned vertically and then horizontally. The procedure is repeated until no further changes are detected in successive iterations. However, because horizontal scanning of pixels destroys their vertical relationship (and vice versa), the algorithm requires as many as 12 iterations (six vertical, horizontal scans each) for ultrasound images, according to our experiments. It is well known that the Hilbert curve preserves the local relationship better than the serial line scan. We propose to apply the original algorithm to pixels scanned along the Hilbert curve. The modified algorithm needs to perform on average only one scan time per image.

3.2 The Windows Adaptive Thresholds

We introduce windows adaptive threshold procedure to further remove noise, as follows:

1. The image of size $N \times M$ is partitioned into $n \times m$ subimages of size $N/n \times M/m$. Since we scan pixels along the Hilbert curve, an image is grouped as 2×2 , 4×4 , ..., $2^k \times 2^k$ subimages, for k being a natural number. Our experiments indicate that it is optimal when $n = m = 2$.

2. We apply Otsu's threshold algorithm to each window to get local thresholds. For each pixel that has a higher intensity than the local threshold of its enclosing window, we substitute it with the local threshold value.

3.3 Core Area

Due to the interference of noise, objects in ultrasound images cannot be reliably identified using traditional features, such as area, shape and texture. We use another feature, the core area [2], for this purpose. The core area of an object is the largest enclosed square whose sides align with the image frame. Figure 1 shows the core area (white square) of a liver object. An efficient algorithm to detect the core area is presented in [2]. Based on the matching of the core areas, we can identify objects of interest whose features (including the core area) are stored in the knowledge base.

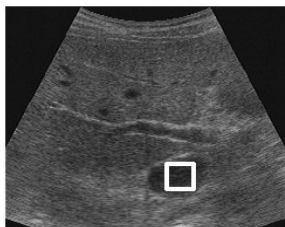


Figure 1: The white square is the core area of a liver object

3.4 The Segmentation Algorithm

Our automatic segmenting algorithm for the liver ultrasound images is as follows:

1. Use equations 1, 2 and 3 (setting $k = 4$) applied the whole image by scanning pixels along the Hilbert curve.
2. Find all local peaks and local valleys, apply the "Cubic Spline Interpolation," and substitute intensity values of pixels between them with their interpolation values.
3. Apply the "windows adaptive threshold."
4. Apply Otsu's algorithm to find the global threshold to classify pixels. Objects are set with white color, and background with black color.
5. Label distinct disconnected objects. We compute the features of these objects, including the *min/max* values of *column/row* coordinates, areas, orientations and centroids. We call the window that covers each object the "object window."
6. Using the features of the objects of interest from the knowledge database, we obtain the *range* within which the features of "candidate objects" must be. If no candidate is found satisfying the ranges, we need to add/reduce the threshold in step 4. Otherwise, go to step 7.
7. Detect the core area of each candidate's "object window."
8. Compare the core areas with the core areas of the interest objects in the database and pick the

candidate objects that have the value within the range. Finally, apply the "Morphological Filtering" with dilation operation for smoothing the structure of the object.

Our procedure can be applied when the knowledge base of the liver object is not available. In this case, we detect an object that has the largest core area. We then build the knowledge base by saving the core area of the liver objects in various views. All of these steps are done automatically. Figure 2 shows segmented images in various stages of our algorithm.

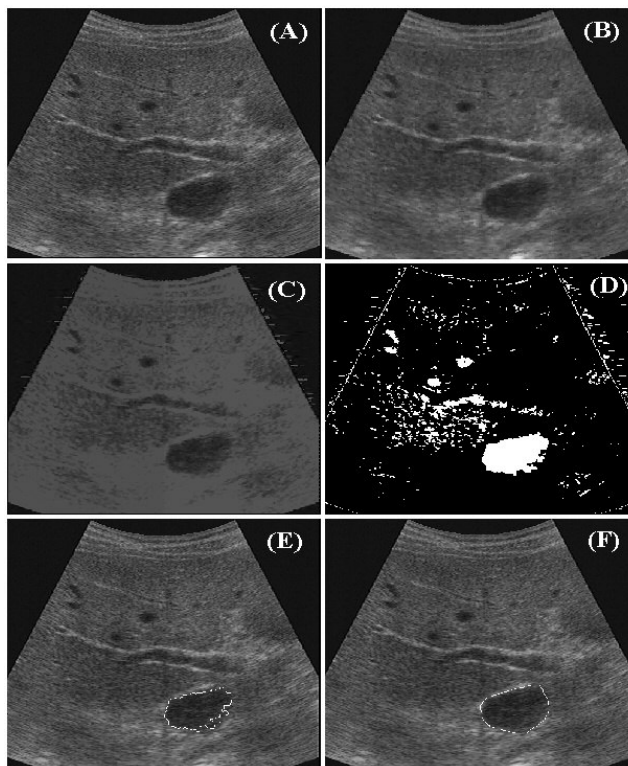


Figure 2: An example shows our approach, (A) an original image, (B) an image after applied peak-and-valley, (C) an image is applied with the windows adaptive threshold, but before we binarize it, (D) a binary image, (E) the detected object contour and (F) object contour by manual.

4. EXPERIMENTAL STUDY

We perform our experiments using a set of 14 representative ultrasound liver images obtained from Florida Hospital. We first observe the effectiveness of our algorithm in finding the right global threshold (Step 4). Although it is a very time consuming process initially, but in the experiments our algorithm never looped twice, between 4 and 6. The right threshold was computed correctly in the first iteration, thanks to the "windows adaptive threshold" (Step 3).

We performed two experimental studies:

- Experiment 1: compares our algorithm, from Step 1 to Step 4, with the relaxation algorithm.
- Experiment 2: compares our algorithm with the original peak-and-valley algorithm.

4.1 Performance under Experiment 1

Figure 3 shows an ultrasound liver image segmented by the compared techniques. Visually inspecting the results indicated by our method, Figure 3 (B), can separate each object more clearly than by the relaxation approach, Figure 3 (C). Our segmented objects also contain less noise. In addition, our approach is much faster; it took less than a second, compared to an average of 58 seconds by the relaxation method.

4.2 Performance under Experiment 2

These experiments indicate the relative accuracy of different approaches in detecting liver objects and can be expressed in terms relating to the distances of orientation and centroid features. We define (x,y) as a centroid coordinate of an object. The object's features include orientation (O), row centroid (x) and column centroid (y). From this point on in Table 1 we will be using "Hpv" to represent our approach, "Orpv" to represent the original peak and valley approach and "relax" to represent relaxation approach. We used values calculated from manual segmentation to represent the relative absolute truth. Within Table 1 this relative absolute truth is assigned the numerical value of ground zero. We selected the differences of the orientation and centroid of detected object to those by the manual segmentation as the interested metric in this experiment. Smaller differences indicate better performance. Therefore the Hpv approach is clearly the closest to ground zero or relative absolute truth! Our approach is also executed faster than the original peak-and-valley, as observed in Section 3.1. The relaxation approach gives the largest numerical values because the relaxation approach cannot separate objects clearly, in Section 4.1.

	Hpv	Orpv	relax
O distance	0.08-8.66	0.49-15.34	2.12-174.68
Average	4.12	5	81.68
Std	2.8	3.94	74.88
x distance	0-9	1-9	1-68
Average	3.71	3.71	27.79
Std	2.6	2.61	22.36
y distance	1-12	1-11	2-107
Average	3	4.43	47.29
Std	2.77	3.34	33.75
Time	<1 sec.	4 seconds	58 seconds

Table 1: summarize the experimental results

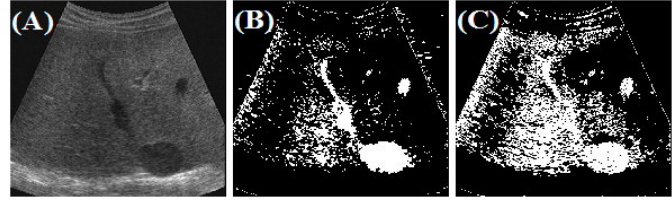


Figure 3: (A) an original image, (B) a binary image using our windows adaptive threshold and (C) a binary image using relaxation threshold with 100 iterations.

5. CONCLUSIONS

We propose an image segmentation system for ultrasound liver images. We modified existing techniques and combined them with our proposed procedures to enable accurate detection of liver objects. Compared with previous approaches, our technique offers many advantages including better accuracy, greater noise reduction, and faster speed. Moreover, our system is fully automatic, thus suitable to integrate into other automate systems.

6. ACKNOWLEDGEMENTS

We appreciate Bernard Fudim, Mercia Mason and Lorie Munizzi for their help in proof reading this paper.

7. REFERENCES

- [1] Piotr S. Windyga, "Fast Impulsive Noise Removal," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 173-179, 2001.
- [2] Khanh Vu, Kien A. Hua and Duc A. Tran, "An Efficient Core-Area Detection Algorithm for Fast Noise-Free Image Query Processing," *In Proc. of The 16th ACM-SIGAPP Annual Symposium on Applied Computing*, pp. 258-263, Mar. 2001.
- [3] Xiaohui Hao, Charles Bruce, Cristina Pislaru and James F. Greenleaf, "A Novel Region Growing Method for Segmenting Ultrasound Images," *IEEE Ultrasonics Symposium*, vol. 2, pp. 1717-1720, 2000.
- [4] Jiankang Wang and Xiaobo Li, "A System for Segmenting Ultrasound Images," *Pattern Recognition proceedings 14th international conference*, vol. 1, pp. 456-461, 1998.
- [5] N. Otsu, "A Threshold Selection Method from Gray Level Histogram," *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-8, pp. 62-66, 1979.
- [6] Namir C. Shamma, *C/C++ Mathematical Algorithms for Scientists & Engineers*, McGraw-Hill, Inc., pp. 65-74, 1995.
- [7] Scott E. Umbaugh, *Computer Vision and Image Processing a practical approach using CVIPtools*, Prentice Hall PTR, 1998.
- [8] <http://www.compuphase.com/hilbert.htm>
- [9] <http://mathworld.wolfram.com/HilbertCurve.html>
- [10] K.R. Castleman, *Digital Image Processing*, Englewood Cliffs, NJ, Prentice Hall, 1995.