Robust Fully-Automatic Segmentation Based on Modified Edge-Following Technique

* **Y. B. CHEN, *OSCAL T.-C. CHEN

*Signal and Media Laboratories, Dept. of Electrical Engineering, National Chung Cheng University, Chia-Yi, Taiwan, R.O.C

**Dept. of Electrical Engineering, Chien Kuo Institute of Technology, Changhua City, Taiwan, R.O.C

ABSTRACT

This work proposes a robust fully-automatic segmentation scheme based on the modified edge-following technique. First, an image frame is partitioned into many B×B-pixel blocks. In each block, the pixel point having the maximum gray-level difference from its neighboring point can be selected as the initial point. Additionally, the threshold can be determined automatically according to the histogram of gray-level difference values of initial points with their neighboring points. By using this threshold value, the edge points are determined from the initial points. Based on edge points, the modified edge-following technique with multiple direction detections is performed to find the closed-loop contours. The whole process is fully automatic, so that the disadvantages of both snake and Live Wire On the Fly (LWOF) schemes on manually selecting the initial contours and points, and the watershed on manually setting the threshold value can be dramatically improved at the same time. Owing to the rapid and fully-automatic segmented results, the proposed scheme is more robust and applicable on various image and video applications than the conventional segmentation schemes.

1.INTRODUCTION

Image segmentation is the fundamental research topic of image processing applications. It can be classified into two categories: semi-automatic segmentation [1,2] and fully automatic segmentation [3,4]. Particularly, applications of medical image analysis, video compression, pattern recognition and so on are explored by using image segmentation schemes. The main purpose of image segmentation is to search for specific objects or regions in an image frame for recognition or compression. In compression applications, the MPEG-4 adopts the object-based compression scheme to segment objects from the background, and then compresses the objects and background individually in order to achieve a high compression ratio. Hence, the fully automatic segmentation becomes an important step in its compression process.

There are many image segmentation approaches, which are popularly used in various applications, summarized in the following.

(1) Active contour approach: Since users have to select the initial contour, this approach is applied in the semi-automatic segmentation. The snake scheme proposed by Kass *et al*. bases on the initial contour to search for the correct contour by minimizing local energy function [5-6]. The disadvantage of the snake scheme is to choose the initial contour that may take a long time. Additionally, it is more difficult to obtain initial contour in an image frame with more complicated contents. Hence, Falcao *et al*. developed the LWOF scheme [7] to select the initial point close to the contour. Based on this initial point, the LWOF scheme starts to search for the contour. When it goes to the wrong way during a searching path, an additional point need be selected to correct the wrong searching direction. Until the closed-loop contour is found, the searching operation is not stopped. The LWOF scheme can be applied to any object shape. In the complicated background, the adequate points are selected by users can yield the correct contour. However, the object with many sharp corners needs many user-selected points, and thereby increasing operation time.

(2) Watershed approach: The image is viewed as the topographic surface. The gray-level value of each pixel represents its height. The catchment basins denote the segmented regions of an image. There are two watershed schemes of rain falling [8] and water immersion [9,10]. The rain falling scheme is very straightforward but takes a lot of computation complexity. The water immersion scheme has complicated operation steps but can rapidly segment objects from the background. The advantages of the watershed approach are to segment multiple objects simultaneously, and to ensure these objects in closed-loop forms. However, the threshold for classifying objects and background is very sensitive to the content of an image. In other words, when the threshold value has a little change, the watershed approach may come out much different segmented results.

In this work, the robust fully-automatic segmentation scheme is developed based on the modified edge-following technique. The conventional edge-following technique only analyzes the current point and next point in the edge direction [11]. Without considering all neighboring points, the conventional edge-following technique is easily trapped to the wrong direction. The modified edge-following technique considers more neighboring points to determine the next contour point. Hence, it increases the probability of finding correct contour points. In order to segment multiple objects, an image frame is partitioned into many B×B-pixel blocks. From each block, the point that has the maximum gray-level difference with its neighboring points is selected as an initial point. These initial points may come from different object contours. When the block size is adequate, small objects and objects with small contrast differences can have their initial points to be selected. The threshold value is determined according to the histogram of gray-level difference values of initial points with their neighboring points. By using this threshold value, the edge points are determined from the initial points. Based on edge points, the modified edge-following technique is performed to find the closed-loop contours. The proposed segmentation scheme that does not need users to select initial points and threshold values can greatly improve the drawbacks of the snake scheme on user-selected initial contour, the LWOF scheme on user-selected initial points, and the watershed schemes sensitive to the selection of the threshold value. Therefore, the segmentation scheme proposed herein can fully automatically segment multiple objects for various image and video applications.

2. THE PROPOSED SEGMENTATION SCHEME

The proposed segmentation scheme can be performed in two parts. The first part is to search for the initial points. The second part is to find the contours based on the initial points.

2.1 The first part:

Initially, an image frame is partitioned into many B×B-pixel blocks, as shown in Fig. 1, with M×N blocks.

The summation of gray-level differences of each pixel in four directions is defined as follows,

$$D_{mn}(x,y)=\sum_{d=0}^{3} | D_{mn}^{d}(x,y) | \quad (1)$$

and

$$D_{mn}^{d}(x,y) = G_{mn}^{d}(r) - G_{mn}^{d+4}(l) \quad (2)$$

where r and l are the neighboring points of (x,y) in directions of d and $(d+4)$, respectively. Here, (x,y) is the coordinate of each pixel in a block where x and y range from 0 to $B-1$. $G(r)$ is the gray-level value of point r. Each point (x,y) and its neighboring points are shown in Fig. 2. The relationship of points (x,y), r and l are shown below.

$$r=\left(x+round\left(cos\left(d\times\frac{\pi}{4}\right)\right),\ y-round\left(sin\left(d\times\frac{\pi}{4}\right)\right)\right); \quad (3)$$

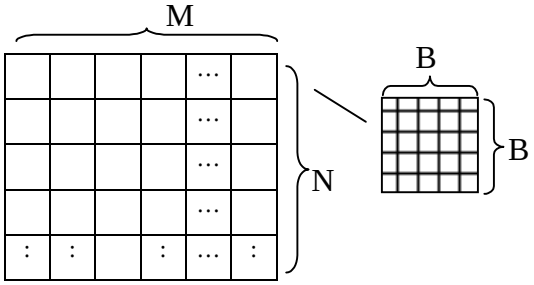


Fig. 1 An image frame partitioned into M×N blocks with a block size of B×B pixels.

$$l=\left(x+round\left(cos\left((d+4)\times\frac{\pi}{4}\right)\right),\ y-round\left(sin\left((d+4)\times\frac{\pi}{4}\right)\right)\right) \quad ; \quad (4)$$

where d is the direction. There are total eight directions, from J^0 to J^7, as shown in Fig.3. (m,n) is the coordinate of each block in an image frame where m ranges from 0 to $M-1$ and n ranges from 0 to $N-1$. Because of the symmetric property, only four directions are needed to be calculated in Eq.(1).

(x-2,y-2)	(x-1,y-2)	(x,y-2)	(x+1,y-2)	(x+2,y-2)
(x-2,y-1)	(x-1,y-1)	(x,y-1)	(x+1,y-1)	(x+2,y-1)
(x-2,y)	(x-1,y)	(x,y)	(x+1,y)	(x+2,y)
(x-2,y+1)	(x-1,y+1)	(x,y+1)	(x+1,y+1)	(x+2,y+1)
(x-2,y+2)	(x-1,y+2)	(x,y+2)	(x+1,y+2)	(x+2,y+2)

Fig. 2 The locations of the point (x,y) and their neighboring points.

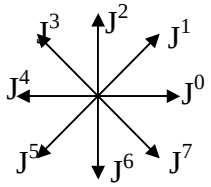


Fig. 3 The 8 directions of J^0, J^1, … , and J^7.

In each block, rank all values of the $D_{mn}(x,y)$ for x and y from 0 to $B-1$ to select the largest one and define it as T_{mn}. The point (x_{mn},y_{mn}) of $Max(D_{mn}(x,y))$ is defined as an initial point in the block (m,n). The histogram of the values of T_{mn} for m from 0 to $M-1$ and n from 0 to $N-1$ is generated. Around M×N/4 initial points are selected to be edge points. According to this histogram, the global threshold can be found and defined as T_g. Figures 4(a) and 4(b) are the original alumgrns image and the alumgrns image with the initial points in all blocks, respectively. It is very clear that not every point is the edge point. Figure 4(c) is the image with edge points obtained by using the threshold value T_g. These edge points are more close to contours.

The edge points are decided in the following steps:

(a) Let h equal to 1.

(b) If D_{mn} is greater than T_g, rank $D_{mn}^{d}(x,y)$ in four directions to find the maximum value and its corresponding direction d. Because the edge direction is perpendicular with the direction d from the maximum value of $D_{mn}^{d}(x,y)$, the initial direction d_0 of the block (m,n) is defined as $(d+2)$. The point (x,y) and the direction d_0 are stored in W_{0,d_0}^{h}. Increase h by one.

On the other hand, if D_{mn} is less than T_g, then the block (m,n) does not contain the edge point.

(c) Repeat step (b) until all blocks have been processed。

After performing the above three steps, all of the edge points and initial directions are stored in W_{k,d_k}^{h}, where k represents the k^{th} contour point.

In the first part, an image frame is partitioned into many $B\times B$-pixel blocks where B equals to 8 in this work. The main goal of this part is to find the edge points and the initial directions. From each block, the point that has the maximum value of D_{mn} is selected as an initial point. Rank all values of D_{mn} in each block to select the maximum value and define it as T_{mn}. Since a quarter of initial points are selected as edge points, the global threshold T_g can be selected from the histogram of the values of T_{mn} from all blocks. By using this threshold value, the edge points are determined from the initial points, and the initial directions d_0 are also found

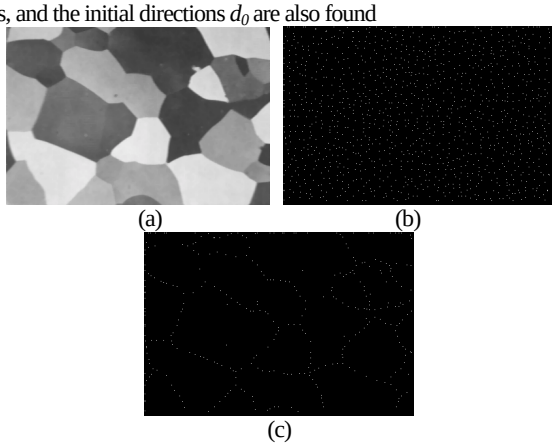


Fig. 4 The analysis results of the alumgrns image (a) The original image. (b) The image with initial points. (c) The image with edge points.

2.2 The second part

1.Let h be one, where h indicates the h^{th} edge point.

2. Each edge point (x_i^h,y_i^h) and its neighboring points are shown in Fig. 2. The shadowed areas represent the 12 positions required in determining the next contour point. The starting position P_h and direction d_0 of the h^{th} edge point are represented by W_{k,d_k}^{h}, where k represents the k^{th} searched contour point, and here k is equal to 0. Point positions of the object contour are represented by $\boldsymbol{W^k=(x^k,y^k)}$.

3. If there is not a large change in the direction, the d_{k+1} of the next point would be three possible ones: $[(8+d_k-1) \bmod 8]$, d_k, and $[(8+d_k+1) \bmod 8]$. For example, when d_k is equal to 3, the next contour point $W_{k+1,d_{k+1}}^{h}$ could appear in J^2, J^3 or J^4 direction and at the $W_{k+1,2}^{h}$, $W_{k+1,3}^{h}$ or $W_{k+1,4}^{h}$ point as shown in Fig. 5. For the left-sided point $L_{k+1,d_{k+1}}^{h}$ and right-sided point $R_{k+1,d_{k+1}}^{h}$ of the predicted contour point, computing the gray-level difference between these two positions could assist in determining the contour point and effectively prevent noise interference. The relationship between $W_{k+1,d_{k+1}}^{h}$, $L_{k+1,d_{k+1}}^{h}$ and $R_{k+1,d_{k+1}}^{h}$ is shown in Fig. 6. The line formed by the W_{k,d_k}^{h} and $W_{k+1,d_{k+1}}^{h}$ points is perpendicular with the line between $L_{k+1,d_{k+1}}^{h}$ and $R_{k+1,d_{k+1}}^{h}$. The $L_{k+1,d_{k+1}}^{h}$ and $R_{k+1,d_{k+1}}^{h}$ of the predicted $(k+1)^{th}$ contour point could be interpreted by the position of the k^{th} contour point and d_{k+1}, as follows:

$$R_{k+1,d_{k+1}}^{h}=\left(x_k^h+round\left(2cos\left((d_{k+1}-1)\times\frac{\pi}{4}\right)\right),\ y_k^h-round\left(2sin\left((d_{k+1}-1)\times\frac{\pi}{4}\right)\right)\right) \quad (5)$$

$$L_{k+1,d_{k+1}}^{h}=\left(x_k^h+round\left(2cos\left((d_{k+1}+1)\times\frac{\pi}{4}\right)\right),\ y_k^h-round\left(2sin\left((d_{k+1}+1)\times\frac{\pi}{4}\right)\right)\right) \quad (6)$$

The gray-level average values of $\boldsymbol{R^h}$ and $\boldsymbol{L^h}$, as shown below, respectively.

$$\overline{L^h}=\frac{1}{k}\sum_{p=0}^{k}I(L_{k-p,d_{k-p}}^{h}) \quad (7)$$

$$\overline{R^h} = \frac{1}{k}\sum_{p=0}^{k} I(R^h_{k-p,d_{k-p}}) \quad (8)$$

$$\Delta^h_{k+1}(i) = \left|I(L^h_{k+1,d_k+j-i}) - I(R^h_{k+1,d_k+j-i})\right| - \left|I(L^h_{k+1,d_k+j-i}) - \overline{L^h}\right| - \left|I(R^h_{k+1,d_k+j-i}) - \overline{R^h}\right| \quad i=1,2\ldots q,\ j=(q+1)/2 \quad (9)$$

Of which, q is the number of directions required, q is initially set to be 3, and then n is equal to 2. Equation (9) is used to determine the $(k+1)^{th}$ contour point. To prevent noise interference, we consider the left and right neighboring points and their averaged pixel values into the equation. If the difference is too great, the wrong contour point may be found. The first item represents the gray-level difference of $R^h_{k+1,d_{k+1}}$ and $L^h_{k+1,d_{k+1}}$. Second and third items could prevent the equation from finding the wrong contours due to the noise interference.

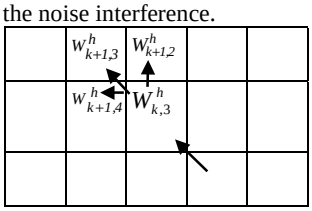


Fig. 5 The predicted points of $W^h_{k+1,2}$, $W^h_{k+1,3}$ and $W^h_{k+1,4}$ when d_k being 3.

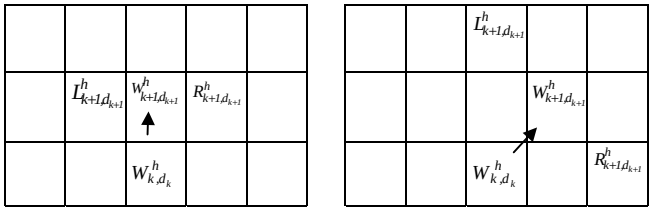


Fig. 6 Two examples of the predicted $W^h_{k+1,d_{k+1}}$ with its neighboring points of $L^h_{k+1,d_{k+1}}$ and $R^h_{k+1,d_{k+1}}$. (a) d_k being 2. (b) d_k being 1.

4. Sort all $\Delta^h_{k+1}(i)$ to select the largest value and define it as $Max(\Delta^h_{k+1}(i))$. If the $Max(\Delta^h_{k+1}(i))$ value is greater than the threshold value T_l, the correct direction is found, go to step 6. Because the global threshold T_g is the summation of gray-level differences at four directions, the local threshold T_l for the gray-level difference at one direction can be $T_g/4$.
5. If $Max(\Delta^h_{k+1}(i))$ is smaller than T_l, it is possible that previously searching direction is deviated from the correct path. Let q be 7 to compute Eq. (8) to find the seven neighboring points, other than the one W^h_{k,d_k}, repeating step 4 to get $Max(\Delta^h_{k+1}(i))$.
6. From $Max(\Delta^h_{k+1}(i))$, the correct direction d_{k+1} and position $W^h_{k+1,d_{k+1}}$ of the $(k+1)^{th}$ contour point are computed as follows.

$$d_{k+1} = d_k + j - i \quad (10)$$

$$W^h_{k+1,d_{k+1}} = \left(x^h_k + round\left(cos\left(d_{k+1}\times\frac{\pi}{4}\right)\right),\ y^h_k - round\left(sin\left(d_{k+1}\times\frac{\pi}{4}\right)\right)\right) \quad (11)$$

7. When the $(k+1)^{th}$ contour point is in the same position as any of the previous searched contour points or has gone beyond the four boundaries of the image, a closed region is completed. If neither condition is established, then k is replaced by k+1, go back to step 3 to find the next point.
8. If h is equal to M×N/4, search for all contours is completed. Otherwise, h is replaced by h+1, and go back to step 2 to find the next contour.
9. Let d_0 equal to (d_0+4), repeat steps from 1 to 8 to search for the contour points in the opposite direction of d_0.
10. Remove the regions without closed-loop forms.

In the second part, we start the searching procedure from each edge point in two opposite directions of d_0 and (d_0+4) to obtain closed-loop contours. The proposed scheme considers more neighboring points to determine the next contour point. Hence, it increases the probability of finding correct contour points. The searching procedure mostly occurs at three directions, only when all values of the $\Delta^h_{k+1}(i)$ are less than T_l then the seven directions are searched. Hence, the searching time can be minimized. The local threshold T_l is derived from the global threshold T_g. Additionally, the whole process of the proposed segmentation scheme in finding initial points, selecting edge points and searching the contour points are fully automatically.

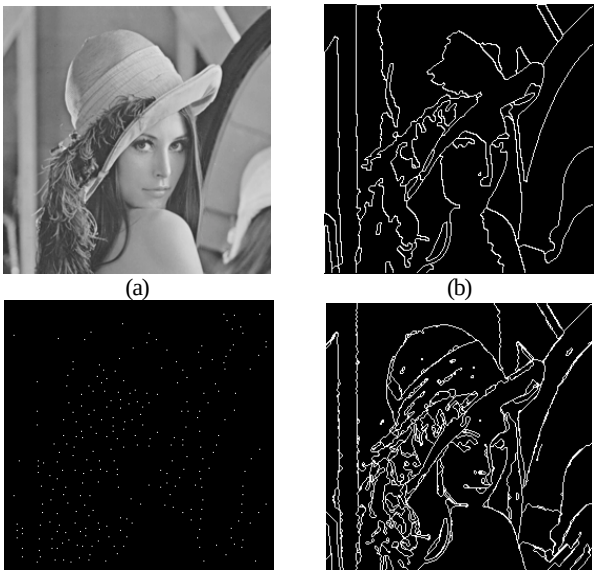
3. COMPUTER ANALYSES

The images of 256×256-pixel Lena and 256×256-pixel Peppers are utilized for segmentation process in this work. Figure 7 shows the segmented results of Lena image. The original Lena image is shown in Fig. 7(a). The result by using the watershed scheme at the threshold value being 30 is depicted in Fig. 7(b). The contrast between the hat and the background in the left-up side is very low, so that the watershed scheme cannot partition these two areas. Figure 7(c) shows the image with edge points by using the proposed scheme. The edge points are located evenly in the contours of hat, hair, face, shoulder, and the object behind Lena. The region close to hair has more edge points because of larger gray-level differences. Figure 7(d) is the result obtained by using the proposed scheme. The up-left side of the hat and the background can be clearly classified into two portions. The region of the hair is segmented into many little closed-loop areas.

The Peppers image shown in Fig. 8 has many overlapping objects at different shapes. Figure 8(a) is the results by using the snake scheme, which requires two initial contours to be selected for the two independent objects before converging to the actual contours. The outer and inner contours for the selected objects are the initial and segmented contours, respectively. This is the most time-consuming scheme, and the result may be far from the correct contour. Figure 8(b) shows the results from the LWOF scheme that also requires the initial points to be selected separately, four points for the left pepper and nine points for the right pepper. Figure 8(c) is the result by using the watershed scheme while the threshold value being 35. The two most important peppers cannot be correctly segmented. The left-down region of the left pepper is mixed with the other peppers, and the right pepper is divided into three major parts. Figure 8 (d) is the result by using the watershed scheme while threshold being 33. Comparing the results between Figs. 8(c) and 8(d), the threshold value has only a little change by 2, but the results are quite different. In Fig. 8(d), there exit more segmented regions. Figure 8(e) shows the image with edge points by using the proposed scheme. The edge points are located evenly in the contours of two large peppers. Figure 8(f) is the segmented result obtained by using the proposed scheme. The two large peppers can be appropriately segmented out, except that the footstalks are divided into several small regions. The other peppers that have small sizes can almost be correctly segmented too.

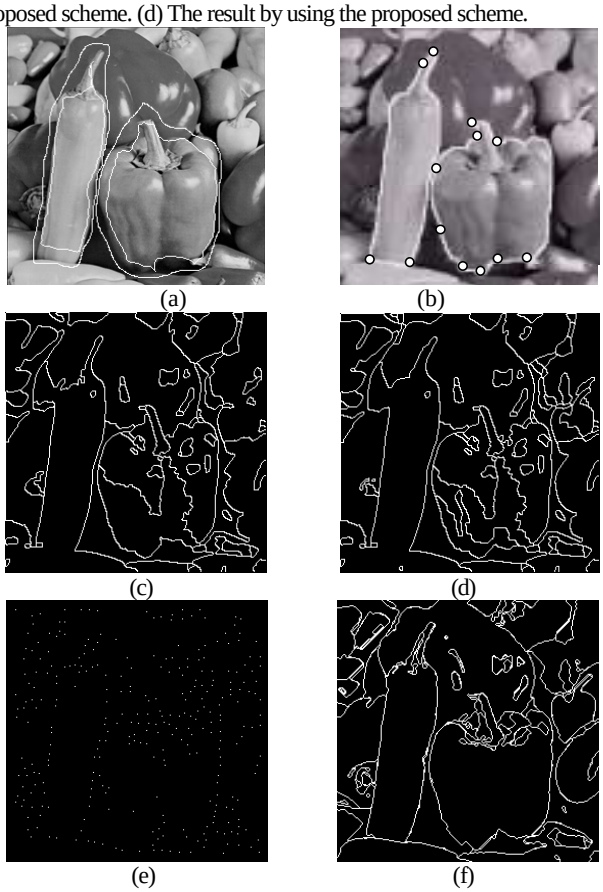
Table 1 lists the number of manually selected points required to segment out two large peppers and the ways of setting threshold values in the proposed, snake, LWOF and watershed schemes. The snake and LWOF schemes need to manually select initial contours and points, respectively. In these two schemes, not only the processing time increases but also the errors are caused by a user to select initial contours or points. The proposed and watershed

schemes don't need to choose the initial contours or points such that the processing time is very low and the segmented results are easily repeatable. In the threshold setting, only the watershed scheme requires the threshold value provided by a user. Additionally, its segmented result is very sensitive to the threshold value. Therefore, The segmentation scheme proposed herein is superior to the conventional ones.



(a) (b) (c) (d)

Fig. 7 The segmented results of Lena image. (a) The original image. (b) The result by using the watershed scheme. (c) The initial points by using the proposed scheme. (d) The result by using the proposed scheme.



(a) (b) (c) (d) (e) (f)

Fig. 8 The segmented results of Peppers image. (a) The result by using the snake scheme. (b) The result by using the LWOF scheme. (c) The result by the watershed scheme with a threshold value of 33. (d) The result by the watershed scheme with a threshold value of 35. (e) The image with edge points by using the proposed scheme. (f) The result by using the proposed scheme.

Table 1. The numbers of manual selected points and the ways of setting threshold values in Peppers image.

Schemes / Operation Ways	Proposed	Watershed	LWOF	Snake
Manual selected points	0	0	13	701
Threshold setting	Auto	Manual	Auto	Auto

4. CONCLUSION

In this work, an image frame is divided into many small blocks. In each block, the point having the maximum gray-level difference with its neighboring points is selected as an initial point. A quarter of initial points are viewed as the edge points. In addition, the global and local threshold values are determined according to the histogram of gray-level difference values of initial points with their neighboring points. Based on edge points, the modified edge-following technique is performed to find the closed-loop contours. The whole process in the proposed scheme is fully automatic. Hence, human errors and operating time are greatly reduced. The proposed segmentation scheme can greatly improve the drawbacks of the snake and LWOF schemes on user-selected initial contours and points, respectively, and of the watershed schemes sensitive to selection of the threshold value. Therefore, the segmentation scheme proposed herein can fully automatically segment multiple objects for various image and video applications in a robust manner.

5. REFERENCES

[1]. Y. B. Chen, Oscal T.-C. Chen, H. T. Chang and J. T. Chien, "An automatic medical-assistance diagnosis system applicable on x-ray images," *Proc. of IEEE MWCAS 2001*, vol 2, pp. 910-914 , 2001.

[2]. Y. B. Chen and Oscal T.-C. Chen, "Semi-automatic image segmentation using dynamic direction prediction," *Proc. of IEEE ICASSP 2002*, vol 4, pp. 3369-3372 , 2002.

[3]. M. Atkins and B. Mackiewich, "Fully automatic segmentation of the brain in MRI," *IEEE Trans. on Med. Imag.*, vol. 17, no. 1, pp. 98-107, 1998.

[4]. G. B. Aboutanos, J. Nikanne, N. Watkins and B. M. Dawant, "Model creation and deformation for the automatic segmentation of the brain in MR images," *IEEE Trans. on Biomedical Engineering*, vol. 46, no. 11, pp. 1346-1356, 1999.

[5]. W. N. Lie and C. H. Chuang, " Fast and accurate snake model for object contour detection," *Electronics Letters*, vol. 37, no. 10, pp. 624 –626, 2001.

[6]. C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 359-369, 1998.

[7]. A. X. Falcao, J. K. Udupa and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: live wire on the fly," *IEEE Trans. on Med. Imag.*, vol. 19, no. 1, pp. 55-62, 2000.

[8]. A. Moga, B. Cramariuc, M. Gabbouj, "An efficient watershed segmentation algorithm suitable for parallel implementation," *Proc. of IEEE International Conference on Image Processing*, vol. 2 , pp. 101 –104, 1995.

[9]. C. J. Kuo, S. F. Odeh, and M. C. Huang, "Image segmentation with improved watershed algorithm and its FPGA Implementation," *Proc. of IEEE ISCAS 2001*, vol 2, pp. 753-756, 2001.

[10]. L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13(6), pp. 583 –597, June 1991.

[11]. I. Pitas, *Digital Image Processing Schemes and Application*, New York: John Wiley & Sons, Inc., 2000.