A ROBUST FULLY-AUTOMATIC SCHEME FOR GENERAL IMAGE SEGMENTATION

Yuan Been Chen,
Dept. of Electronic Engineering, Chienkuo Technology University, Changhua City, Taiwan, R.O.C

ABSTRACT

This work proposes a robust fully-automatic segmentation scheme based on the modified edge-following technique. The entire scheme consists of four stages. In the first stage, global threshold is computed. Followed by the second stage in which positions and directions of the initial points are determined. Local threshold is derived based on the histogram of gradients from the third stage. Finally, in the fourth stage, Searching procedure is started from each initial point to obtain closed-loop contours. The whole process is fully automatic, so that the disadvantages of semi-automatic schemes of manually selecting the initial contours and points, and the watershed schemes sensitive to the selection of the threshold value can be dramatically improved at the same time. Owing to the tremendous reduction of human errors and operating time, the proposed scheme is more robust and applicable on various image and video applications than the conventional segmentation schemes.
I. INTRODUCTION

Image segmentation is the primary research subject of image processing applications. The key purpose of image segmentation is to look for specific objects or regions in an image frame for recognition or compression. Particularly, applications of medical image analysis, video compression, pattern recognition, etc. are explored by using image segmentation schemes. In general, segmentation schemes could be categorized into two principal types: semi-automatic segmentation [1,2] and fully automatic segmentation [3-4].

(A) Semi-automatic segmentation: The problem of this scheme is the requirement of subjective human interference, which makes the repeatability of segmentation on the same image not comparable to that done with fully automatic segmentation and operating time becomes longer. However the merit of semi-automatic segmentation is the fact that the same segmentation method could be applied on any image, without restrictions on its type. This scheme would require selecting different initial points for different images, and setting different thresholds, before deriving the desired contour of an object. The snake (active contour) scheme proposed by Kass et al. is based on the initial contour to obtain the correct contour by minimizing local energy function [5-8]. The disadvantage of the snake scheme is that selecting the initial contour may take a long time. Additionally, it is more difficult to obtain initial contour in an image frame with more complicated contents. Therefore, Falcao et al. developed the LWOF (Live Wire on the Fly) scheme[9] to select the initial point close to the contour. Based on this
initial point, the LWOF scheme starts to search for the contour. If it goes the incorrect way during a search, an extra point needs to be selected to correct the erroneous searching direction. Until the closed-loop contour is found, the searching procedure is not stopped. The LWOF scheme can be applied to any object form. However, an object with many sharp corners needs many user-selected points, thereby increasing operation time.

(B) Fully-automatic segmentation: The fully-automatic segmentation may be categorized into two main types for the different applications.

(i) Applied to a specific type of image: The segmented objects exhibit some similarity. Accordingly, some criteria may be predetermined to simplify the segmentation procedure. In the case of medical images, fully automatic segmentation is applied on body parts such as leg bones [4], brain [10, 11], fingers [12], or ribs [13] to get their contours. With such a method, information on characteristics such as the area, lengths, and angles of an organ could be obtained to provide assistance to a doctor’s diagnosis. The contents of images usually could be known in advance, so that the pre-processing could be done to help the segmentation procedure. Fully automatic segmentation is known to be exceedingly fast and completely free of human interference. In practice, these attributes would provide fast, correct and objective data to assist with diagnosis, which is what medical image processing is all about. The disadvantage is that such a method is only applicable on limited types of images that belong to a specified category. For example, segmentation applied on ribs would not be applicable to the brain.
(ii) Applied to the general type of images: In the video compression applications [14, 15], the object-based compression scheme can be used to segment objects from the background, and then compresses the objects and background individually for achieving a high compression ratio. Hence, the fully automatic segmentation becomes an important step in its compression process. Based on the active contours scheme, the Geodesic active contours [20,21] and Level sets[21,22] were proposed to detect and track multiple moving objects in image sequences. Recently the extended gradient vector flow (E-GVF) field model [23] has been proposed for multi-object segmentation. The downstream process is automatic and requires no human interaction, making the active contour algorithm more suitable for practical applications. But these scheme usually focus on separate objects, it is not suitable for the images which include the objects side by side or objects inside the object. In the Watershed scheme, 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 [16] and water immersion [17, 18]. 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. Recently the improved watershed transform [23] has been proposed, which enables the use of different prior information-based difference functions for different objects, instead of the usual gradient calculation. The advantages of the watershed approach are that it can segment multiple objects simultaneously, and ensure that these objects are 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 with 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 searching direction [19]. Without considering all neighboring points, the conventional edge-following technique could easily go 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.

II. PROPOSED SCHEME

The entire scheme consists of four stages. Fig. 1 illustrates the flowchart of the entire system. In step 1, an image frame is partitioned into $B \times B$ blocks, and then the global threshold is computed as shown in Fig. 1(a).

In step 2, the positions and directions of the initial points are determined as shown in Fig. 1(b). In step 3, the entire image frame is partitioned into many $b \times b$-pixel blocks to calculate the local threshold in each block as shown in Fig. 1(c). Finally, in step 4, the searching procedure is started from each initial edge point to obtain closed-loop contours in the fourth stage as shown in Fig. 1(d). In the following description, the computing results are rounded-off to an integer.
Step 1. Determining the Global Threshold \( T_g \)

Initially, an image frame is partitioned into \( B \times B \) blocks. \( B \) is equal to 3 in the following description. Raising the value of \( B \) can obtain more initial points, but the processing time will be increased. The maximum of the difference between the right and left neighboring points in direction of \( d \) are defined as \( C_{m,n}(x,y) \) in Eq.(1).

\[
C_{m,n}(x,y) = \max(|I(r_{m,n}^d(x,y)) - I(l_{m,n}^d(x,y))|) \quad \text{for } 0 \leq d \leq 3
\]

(1)

\[
T_{g,m,n} = \max(C_{m,n}(x,y))
\]

(2)

\[
T_g = \min(T_{g,m,n})
\]

(3)

Where \( r_{m,n}^d(x,y) \) and \( l_{m,n}^d(x,y) \) are the right and left neighboring points of \((x,y)\) in direction of \( d \), respectively. \( d \) is the 8-neighboring directions, from 0 to 7, as shown in Fig.2. Since \( r_{m,n}^d(x,y) \) and \( l_{m,n}^d(x,y) \) are symmetric in direction of \( d \) and \((d+4)\), only four directions need to be calculated. \((m,n)\) are the coordinates of each block in an image frame, both \( m \) and \( n \) range from 0 to \( B-1 \). \((x,y)\) are the coordinates in each block, where \( x \) ranges from 0 to \((M/B)-1\) and \( y \) ranges from 0 to \((N/B)-1\). \( M \) and \( N \) represent the width and the height of the image, respectively. Function \( \max \) and \( \min \) represent the maximum and minimum of values, respectively. \( I(r) \) is the gray-level value of point \( r \). The relationship of points \((x,y)\), \( r_{m,n}^d(x,y) \) and \( l_{m,n}^d(x,y) \) are shown below.

\[
r_{m,n}^d(x,y) = \left[ x + \cos \left( d \times \frac{\pi}{4} \right), \ y - \sin \left( d \times \frac{\pi}{4} \right) \right]
\]

(4)

\[
l_{m,n}^d(x,y) = \left[ x + \cos \left( (d+4) \mod 8 \times \frac{\pi}{4} \right), \ y - \sin \left( (d+4) \mod 8 \times \frac{\pi}{4} \right) \right]
\]

(5)

In each block, rank all values of the \( C_{m,n}(x,y) \) to select the largest one and define it as \( T_{g,m,n} \), and then select...
the smallest one of the $T_{g_{m,n}}$ and define it as $T_g$. Therefore, at least one initial point in each block could be discerned. For example, the $T_{g_{m,n}}$ in each block of “alumgms” image is shown in Fig. 3, the $T_g$ being 80.

**Step 2. Finding the Initial Points**

After the $T_g$ is computed, the initial points of the image can be found in this stage. The maximum difference of each pixel can be defined as follows.

$$\max (|r_{d}(x,y) - l_{d}(x,y)|) \quad |_{0 \leq d \leq 3} (6)$$

Where $x$ ranges from 0 to $M-1$ and $y$ ranges from 0 to $N-1$. The initial points are determined in the following steps:

(a) Let $h$ be equal to 0.

(b) If $C(x,y)$ is greater than $T_g$, the initial point $(x,y)$ and the direction are stored in $W_{h,0}$ and $d_{h,0}$, respectively. Increase $h$ by one. On the other hand, if $C(x,y)$ is less than $T_g$, then the point $(x,y)$ is not the initial point.

(c) Repeat step (b) for all points.

After performing the above four steps, all of the initial points and initial directions are stored in $W_{h,0}$ and $d_{h,0}$, respectively. And the number of initial points is equal to $H$. Where the first subscript $h$ represents the $h^{th}$ contour and the second subscript 0 represents the initial point.

**Step 3. Determining the Local Threshold $T_l$**
Local threshold \( T_l \) is the criterion for the searching procedure in the next stage. Initially, an image frame is partitioned into many \( b \times b \)-pixel blocks. If the image has cluttered background, small \( b \) could be selected to preserve the properties of the small objects. Otherwise, if the image has smooth background, big \( b \) could be selected to reduce the processing time. The average of gray-level absolute differences of gray-level differences of each pixel in four directions is defined as follows,

\[
D_{m,n}(x,y) = \frac{1}{4} \sum_{d=0}^{3} \left( | I(r_{m,n}^d(x,y)) - I(l_{m,n}^d(x,y)) | \right)
\]

(7)

Where \((m,n)\) are the coordinates of each block in an image frame, \( m \) ranges from 0 to \((M/b)-1\) and \( n \) ranges from 0 to \((N/b)-1\). \((x,y)\) are the coordinates in each block, both \( x \) and \( y \) range from 0 to \( b-1 \). \( d \) is the 8-neighboring directions, from 0 to 7, as shown in Fig.2. Since \( l_{m,n}^d(x,y) \) and \( l_{m,n}^{d+4}(x,y) \) are symmetric in direction of \( d \) and \((d+4)\), only four directions need to be calculated. In each block, rank all values of the \( D_{m,n}(x,y) \) to select the largest one and define it as \( T_{lm,n} \).

\[
T_{lm,n} = \text{Max}(D_{m,n}(x,y))
\]

(8)

The histogram of \( T_{lm,n} \) is generated as follows [25].

(a). Assign zero values to all elements of the histogram \( H(k) \), where \( k \) ranges from 0 to 255.

(b). For all \( m \) and \( n \) of the image, increment \( H(T_{lm,n}) \) by 1.

(9)

The local threshold \( T_l \) can be defined as

\[
\sum_{k=0}^{T_l} H(k) = \frac{M \times N}{b \times \frac{b}{2}}.
\]

(10)

Considering the special case that an image contains only black-white interlaced strips, each strip has the
width of one-pixel. In this case, the image has the maximum contour points which are half of the total points. 

$T_l$ is selected to include half of the total points in the left side of the histogram. It can prevent from missing the possible contour points. For example, in a 500×500-pixel image, $b$ is set to be 1, the maximum contour points should be $\frac{500 \times 500}{2} = 125000$. In other words, the front half-part of the $H(k)$ contains the non-contour points. Taking the “alumgrns” image as an example, the histogram of $Tl_{m,n}$ is depicted in Fig. 3(c). In next stage, the searching procedure initially searches only in three directions. When the differences between the right and left neighboring points in three directions are smaller than $T_l$, the seven directions are re-searched. For example, when the differences between the right and left neighboring points in directions 3, 2 and 1 are smaller than $T_l$, the seven directions in 5, 4, 3, 2, 1, 0 and 7 defined in Fig. 2 are re-searched.

**Step 4. Modified Edge-Following Scheme**

In the former three stages, initial points, global Threshold $T_g$ and local threshold $T_l$ are computed. In this stage, the searching procedure is started from each initial point until the closed-loop contour is found. Fig.4 illustrates the flow chart of the modified edge-following scheme. Each initial point and its neighboring points are shown in Fig. 5. The shadowed areas represent the 12 positions required in determining the next contour point. The starting position and direction of the $h^{th}$ initial point are represented by $W_{h,k}$ and $d_{h,k}$, where $k$ represents the $k^{th}$ searched contour point. Point positions of the
object contour are represented by \( W_{h,k} = (x_{h,k}, y_{h,k}) \).

The modified edge-following scheme is described as follows.

1. Let \( h \) be zero, where \( h \) indicates the \( h^{th} \) initial point. In this step, the searching procedure begins from the first initial point.

2. Let \( s \) be zero, \( s \) represents the searching directions of initial points. While \( s \) is equal to 0 and 1, the initial searches in initial point \( h \) are \( d_{h,0} \) and \([(d_{h,0} + 4) \mod 8] \), respectively. The two kinds of values of \( s \) indicate that there are two searching directions of each contour point.

3. Let \( k \) be zero, where \( k \) indicates the \( k^{th} \) contour point. In this step, the searching procedure begins from the first contour point of the \( h^{th} \) initial point.

4. If there is not a large change in the direction, the \( d_{h,k+1} \) of the next point would be three possible ones: \([(8 + d_{h,k} -1) \mod 8], d_{h,k}, \) and \([(8 + d_{h,k} +1) \mod 8] \). For example, when \( d_{h,k} \) is equal to 1, the next contour point \( W_{h,k+1} \) could appear at the predicted contour point \( P^0_{h,k+1}, P^f_{h,k+1} \) or \( P^2_{h,k+1} \) as shown in Fig. 6.

5. For the left-sided point \( L^{d_{h,k+1}}_{h,k+1} \) and right-sided point \( R^{d_{h,k+1}}_{h,k+1} \) 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 \( P^{d_{h,k+1}}_{h,k+1}, L^{d_{h,k+1}}_{h,k+1} \) and \( R^{d_{h,k+1}}_{h,k+1} \) is shown in Fig. 7. The line formed by the \( W_{h,k} \) and \( P^{d_{h,k+1}}_{h,k+1} \) points is perpendicular with the line between \( L^{d_{h,k+1}}_{h,k+1} \) and \( R^{d_{h,k+1}}_{h,k+1} \). The \( L^{d_{h,k+1}}_{h,k+1} \) and \( R^{d_{h,k+1}}_{h,k+1} \) of the predicted \( (k+1)^{th} \) contour point could be interpreted by the position of the \( k^{th} \) contour point and \( d_{h,k+1} \), as follows:
\[
R_{h,k,i+1}^d = x_{h,k} + \left( 2 \cos \left( (d_{i,k+1} - 1) \times \frac{\pi}{4} \right) \right), \quad y_{h,k} - \left( 2 \sin \left( (d_{i,k+1} - 1) \times \frac{\pi}{4} \right) \right) \quad (12)
\]

\[
L_{h,k,i+1}^d = x_{h,k} + \left( 2 \cos \left( (d_{i,k+1} + 1) \times \frac{\pi}{4} \right) \right), \quad y_{h,k} - \left( 2 \sin \left( (d_{i,k+1} + 1) \times \frac{\pi}{4} \right) \right) \quad (13)
\]

To prevent noise interference, the left and right neighboring points and their averaged pixel values are considered into the equation. If the difference is too large, the wrong contour point may be found. The gray-level average values of \( R_h \) and \( L_h \) are shown below, respectively.

\[
\overline{R}_h = \frac{1}{k} \sum_{p=1}^{k} I(R_{h,k-p}) \quad (14)
\]

\[
\overline{L}_h = \frac{1}{k} \sum_{p=1}^{k} I(L_{h,k-p}) \quad (15)
\]

\[
E_{h,k+1}(u) = \left| I(L_{h,k+1}^{d_{i,k+1}+u}) - I(R_{h,k+1}^{d_{i,k+1}+u}) \right| - \left| I(L_{h,k+1}^{d_{i,k+1}+v}) - \overline{L}_h \right| - \left| I(R_{h,k+1}^{d_{i,k+1}+v}) - \overline{R}_h \right|, \text{ for } 0 \leq u \leq q-1 \quad (16)
\]

Where \( u \) ranges from 0 to \( q-1 \), \( q \) is the number of directions required which is initially set to be 3, and then \( v = (q + 1)/2 \) is equal to 2. Equation (16) is used to determine the \((k+1)\)th contour point. The first term represents the gray-level difference of \( R_{h,k+1}^{d_{i,k+1}} \) and \( L_{h,k+1}^{d_{i,k+1}} \). Second and third items could prevent the equation from finding the wrong contours due to the noise interference.

5. Sort all \( E_{h,k+1}(u) \) to select the largest value and define it as \( \text{Max}\{ E_{h,k+1}(u) \}, \text{ for } 0 \leq u \leq q-1 \}. \) If the \( \text{Max}\{ E_{h,k+1}(u) \}, \text{ for } 0 \leq u \leq q-1 \} \) value is greater than the local threshold value \( T_l \), the correct direction is found, go to step 8.

6. If \( \text{Max}\{ E_{h,k+1}(u) \}, \text{ for } 0 \leq u \leq q-1 \} \) is smaller than \( T_l \), it is possible that the previously searched direction has deviated from the correct path. Let \( q \) be 7 to compute Eq. (16) to find the seven neighboring points, repeating step 4 to get \( \text{Max}\{ E_{h,k+1}(u) \}, \text{ for } 0 \leq u \leq q-1 \}. \)
7. If the \( \text{Max}_u \{ E_{h,k+1}(u), \text{for } 0 \leq u \leq q-1 \} \) value is greater than the threshold value \( T_l \), the correct direction is found, else stop the searching procedure and go to step 10.

8. From \( \text{Max}_u \{ E_{h,k+1}(u), \text{for } 0 \leq u \leq q-1 \} \), the correct direction \( d_{h,k+1} \) and position \( E_{h,k+1}(u) \) of the \((k+1)^{th}\) contour point are computed as follows.

\[
d_{h,k+1} = ((d_{h,k}+v-u) \mod 8)
\]

\[
W_{h,k+1} = \left( x_{h,k} + \cos \left( d_{h,k+1} \times \frac{\pi}{4} \right) , \quad y_{h,k} = \sin \left( d_{h,k+1} \times \frac{\pi}{4} \right) \right)
\]

9. 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 or \( \text{Max}_u \{ E_{h,k+1}(u), \text{for } 0 \leq u \leq q-1 \} \) is smaller than \( T_l \), the searching procedure is completed. If neither condition is true, then \( k \) is replaced by \( k+1 \), go back to step 3 to find the next contour point.

10. If \( s \) being 1, the searching procedure in two directions of the \( h^{th} \) initial point are completed, go to step 11.

11. Else let \( d_{h,0} \) equal to \( [(d_{h,0}+4) \mod 8] \) and \( s \) being 1, repeat steps from 1 to 7 to search for the contour points in the opposite direction of \( d_{h,0} \).

11. If \( h \) is equal to \( H-1 \), the search for all contours is completed. Otherwise, \( h \) is replaced by \( h+1 \), and goes back to step 2 to begin the searching procedure from the next initial point.

12. Remove the contours without closed-loop forms, except the incomplete object appears at four boundaries of the image.

In this stage, we start the searching procedure from each initial point in two opposite directions of
and \( [(d_{h,0} + 4) \mod 8] \) to obtain closed-loop contours. The advantage of the conventional edge-following method is its simplicity, since it only has to compute the gradients of the eight points surrounding a contour point to obtain the next contour point. However, the limitation of the conventional edge-following method is that it is easily influenced by noise, causing it to fall into the wrong edge. This wrong edge can form a wrong route to result in an invalid segmented area. 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 in three directions, only when all values of the \( E_{h+k,l}(u) \) are less than \( T \) then the seven directions are searched. Hence, the searching time can be minimized.

III. Experimental Results

A. Parameters selection and definition

In the following experiment, the LWOF, E-GVF snake, watershed and proposed methods are adopted and compared in processing time and segmentation accuracy. In LWOF [9], the user can adequately select some points close to an object to obtain a segmentation result that is closest to that observed. In this work, contrast level=40, contrast range=20. In snake [5], the elasticity force acts to keep the curve from stretching. Larger values make the contour harder to stretch along its length. The rigidity force acts to keep the curve from bending too much, as, for example, when turning a corner. Larger values make the curve harder to bend. The viscosity controls how quickly and how far the curve can be
deformed between iterations. Larger values make the curve harder and slower to deform. Larger external forces cause a stronger force toward the image edges. In this work, elasticity=1/6, rigidity=1/6, viscosity=4/6, external force =2, snake iteration=100. The segmentation function adopted by the watershed method in our simulations is gradient [13]. Additionally, the merging operation is based on the region mean where the threshold indicates the criterion of region merging. The number of segmented regions decreases while the threshold value increases. In this work, weight for mean gray level=0.5, iterations for smooth =5, iterations for merge=30.

The signal-to noise ratio (SNR) is defined as

\[
\text{SNR} = 10 \log_{10} \left( \frac{x(i, j)^2}{\sum_{i,j=1}^{M,N} [x(i, j) - \hat{x}(i, j)]^2} \right) \text{ dB.}
\]

Where \(x(i,j)\) and \(\hat{x}(x, j)\) represent the original image and noisy image, respectively.

Error rate is defined as

\[
\text{Error rate} = \frac{\text{number of error pixels}}{\text{Total number of object pixels}}.
\]

B. Results for the “square” image

Fig.8 shows the 360×360 “square” image with noises added by Gaussian distribution, at the signal-to noise ratio (SNR) of 9.14dB. Fig. 8(a) and 8(b) are the original and the noisy images, respectively. Fig. 8(c) is the result obtained by the conventional edge-following scheme. The noises have great effect upon the result. Without considering all neighboring points, the conventional edge-following technique could easily go to the wrong
direction. Fig. 8(d) illustrates the result by the LWOF segmentation. There are not many points selected since the angles of the turns are not very large, but the contour is also not smooth due to the noises. Fig. 8(e) shows the result obtained by the snake scheme. Some dark areas could be lost in the sharp corners. Fig. 8(f) shows the result obtained by the watershed scheme. Most of the target areas could be segmented, but the noises still cause the visible error. Fig. 8(g) depicts the segmented result obtained by using the proposed scheme. The proposed scheme can improve the drawback of the conventional edge-following scheme and obtain the correct area. Table 1 compares error rates of the four segmentation schemes at SNRs of 18.87dB, 12.77dB, and 9.14dB. It is clear that with the proposed scheme, the segmented area has the lowest error in each of the four SNR scenarios.

C. Results for the “alumgrns” image

Fig. 9 depicts the 404×271 “alumgrns” image. There are many closed-form regions in this image. Snake and LWOF schemes are not suitable for segmenting the objects in this image. Since one object requires one initial selection, the initial points must be selected many times in this image. Figures 9(a), 9(b) and 9(c) are the results by the watershed scheme with the threshold values being 20, 30 and 40, respectively. The most segmented objects can be found with the threshold value being 20, but some of them are incorrect. Some of the regions, especially the small ones, could be ignored with the threshold value being 40. The segmented objects decrease while the threshold value increases. Fig. 9 (d) is the result by the proposed scheme with global threshold $T_g$ being 80 and local threshold $T_l$ being 10. We can see that the proposed scheme rather than the watershed scheme could obtain the small regions.
D. Results for the “peppers” image

Figure 10(a) shows the “peppers” image, which includes many overlapping objects at different shapes. Snake and LWOF schemes are not suitable for segmenting the objects in this image; the initial points also must be selected many times in this image. Figures 10(b), (c) and (d) are the results obtained by the watershed scheme with the threshold values being 20, 30 and 40, respectively. In Figs. 10 (b) and (c), it is obvious that the two images are over-segmented. Moreover, 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. Owing to the raising of the threshold, Fig. 10 (d) is under-segmented; some peppers are mixed with neighboring one. Figure 10(e) is the segmented result obtained by 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.

E. Results for the “mobile” image

Fig. 11 depicts the 364×244 “mobile” image, which includes many different animals on the left side, a calendar on the right side, and a train in the lower side. Figs. 11(b), (c) and (d) are the result obtained by the watershed scheme with threshold values being 20, 30 and 40, respectively. Fig. 11(b) has the lowest threshold value, the over-segmented objects are obtained in the marked area. The result in Fig. 11(c) is very similar to the proposed scheme, but some of the tiny parts may still be missed as shown in the marked area. The threshold value is raised in Fig. 11(d) and could cause some distortion, such as the numerals on the calendar disappearing. Fig.11
(e) is the result obtained by the proposed scheme with $T_g$ being 181 and $T_l$ being 27. Comparing this with the watershed scheme, the proposed scheme could compute the applicable threshold value, hence the result is more accurate. The marked areas are the tiny regions which could not be obtained in the watershed scheme. Table 2 lists the computational time of the watershed and the proposed schemes. Although the processing speed of the proposed scheme is not the fastest, the computational time between the proposed scheme and the watershed scheme are very close. The results obtained by the watershed methods take slightly lower processing time than the proposed method when the threshold selection time is not counted in the watershed method. The above experiments were all implemented based on C-codes running on an Intel T2250@1.73GHz CPU.

**F. Results for the “table tennis” image**

Fig 12(a) shows the original 360×240 “table tennis” image. The contrast between the wall, clothes, and the table is very low, so that the watershed scheme would obtain impure segmentation results. Figs.12 (b), (c) and (d) are the results by using the watershed scheme while threshold value being 20, 30 and 40, respectively. When threshold value being 20, the segmented contours are more but disorderly and confused, objects could be over-segmented to the undesired regions. For example, the wall between poster and table can be effected by the shadow, and the clothes can not be correctly segmented because the light changing and the fold. The border of the table is discontinuous because the gray-level between the line and wall is too close. While the threshold values are raised to be 30 and 40, the over-segmented effect could be improved, but some of the
correct contours could be missed. In Fig. 12 (d), it’s obviously that the table almost disappears, and some edge of the clothes is lost. Fig. 12 (e) is the result by using the proposed scheme while $T_g$ being 169 and $T_l$ being 21. Comparing the proposed scheme with watershed scheme, the contour of table is segmented without the broken section, and the clothes could be more completely segmented.

IV. CONCLUSION

In this research, an image frame is first partitioned into $B \times B$ blocks to find the global threshold and the initial points. The original image is then partitioned into many small blocks, and local threshold values are determined in each block. Based on initial points, the modified edge-following technique is performed to locate the closed-loop contours. By also computing and analyzing the characteristics of the left and right neighboring points of the next estimated contour point, the ability to overcome noise interference would be greater. In the noisy image, the proposed scheme not only can improve the drawback of the conventional edge-following scheme but also achieve better results than the watershed and the snake schemes. Owing to the computation process of $T_g$ and $T_l$, the total computational time of the proposed scheme has slightly increased over that of the watershed scheme. But the processing time would be less than the watershed scheme only if the edge-points searching procedure is considered. The proposed segmentation scheme that does not need human input 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 sensitivity to the selection of the threshold value. Therefore, the fully automatic segmentation scheme proposed herein can segment multiple objects for various image and video applications.

V. REFERENCE


[16]. A. Moga, B. Cramariuc, M. Gabbouj, “An efficient watershed segmentation algorithm suitable for


Partition an image frame into $B \times B$ blocks and calculate the global threshold $T_g$.

Find the initial points and the initial directions.

Partition an image frame into many blocks, each block is $b \times b$. Calculate the local threshold $T_l$.

Search the contour by using the modified edge-following scheme.

Fig. 1 The flowchart of the proposed scheme applied on the “rice” image.
Fig. 2 The $T_{g_{mn}}$ in each block of alumgrains image. (a) original image. (b) The values of $T_{g_{mn}}$.

Fig. 3 An image frame partitioned into small blocks with a block size of $b \times b$ pixels.
Derive \([(8+d_{h,k}-1) \mod 8] \cdot d_{h,k} \cdot [(8+d_{h,k}+1) \mod 8]\)

Compute \(E_{h,k+1}(u)|u=0\sim2\) for three directions

Max(\(E_{h,k+1}(u)=T_l\))

Compute the seven directions other than the opposite direction of \(d_{h,k}\)

\(E_{h,k+1}(u)|u=0\sim6\)

Find actual \(d_{h,k}\), compute \(W_{h,k+1}\) with \(W_{h,k}\) & \(d_{h,k}\)

\(k=k+1\)

\(s=1, d_{h,0}=(d_{h,0}+4) \mod 8\)

\(h=h+1\)

\(h>H-1\)

Remove the contours without closed-loop

End

Fig. 4 The flow chart of modified edge-following scheme
Fig. 5 The locations of the point \((x,y)\) and their neighboring points.

\[
\begin{array}{cccccc}
(x-2,y-2) & (x-1,y-2) & (x,y-2) & (x+1,y-2) & (x+2,y-2) \\
(x-2,y-1) & (x-1,y-1) & (x,y-1) & (x+1,y-1) & (x+2,y-1) \\
(x-2,y) & (x-1,y) & (x,y) & (x+1,y) & (x+2,y) \\
(x-2,y+1) & (x-1,y+1) & (x,y+1) & (x+1,y+1) & (x+2,y+1) \\
(x-2,y+2) & (x-1,y+2) & (x,y+2) & (x+1,y+2) & (x+2,y+2)
\end{array}
\]

Fig. 6 The predicted points of \(P^0_{h,k+1}\), \(P^1_{h,k+1}\) and \(P^2_{h,k+1}\) when \(d_{h,k}\) equaling to 1.

Fig. 7 Two examples of the predicted \(P^d_{h,k+1}\) with its neighboring points of \(L^d_{h,k+1}\) and \(R^d_{h,k+1}\) (a) \(d_{h,k+1}\) being 2. (b) \(d_{h,k+1}\) being 1.
Fig. 8 The segmented results of the square image with noises added by Gaussian distribution at SNR of 9.14dB.

(a) The noisy image. (b) The result obtained by using the conventional edge-following scheme. (c) The result obtained by using the LWOF scheme. (d) The result obtained by using the snake scheme. (e) The result obtained by using the watershed scheme. (f) The result obtained by using the proposed scheme.
Fig. 9 The segmented results of alumgrms image. (a) The result obtained by the watershed scheme with the threshold value of 20. (b) The result obtained by the watershed scheme with the threshold value of 30. (c) The result obtained by the watershed scheme with the threshold value of 40. (d) The result obtained by the proposed scheme.

Fig. 10 The segmented results of bacteria image. (a) The original image. (b) The result obtained by the watershed scheme with the threshold value of 20. (c) The result obtained by the watershed scheme with the threshold value of 40. (d) The result obtained by the proposed scheme.
Fig. 11 The segmented results of ‘mobile’ image. (a) The original image. (b) The result obtained by the watershed scheme with the threshold value of 20. (c) The result obtained by the watershed scheme with the threshold value of 30. (d) The result obtained by the watershed scheme with the threshold value of 40. (e) The result obtained by the proposed scheme.
Fig. 12 The segmented results of ‘chessboard’ image. (a) The original image. (b) The result obtained by the watershed scheme with a threshold value of 20. (c) The result obtained by the watershed scheme with a threshold value of 30. (d) The result obtained by the watershed scheme with a threshold value of 40. (e) The result obtained by the proposed scheme.

Table 1 The error rates and CPU Time of the four segmentation schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Error rates</th>
<th>SNR=18.87dB</th>
<th>SNR=12.77dB</th>
<th>SNR=9.14dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>0.34</td>
<td>1.28%</td>
<td>1.43%</td>
<td>1.48%</td>
</tr>
<tr>
<td>LWOF</td>
<td>×</td>
<td>1.72%</td>
<td>2.10%</td>
<td>2.27%</td>
</tr>
<tr>
<td>Snake</td>
<td>1.51</td>
<td>1.97%</td>
<td>2.29%</td>
<td>2.54%</td>
</tr>
<tr>
<td>watershed</td>
<td>0.31</td>
<td>1.89%</td>
<td>2.23%</td>
<td>2.50%</td>
</tr>
</tbody>
</table>

Table 2 The error rates of the watershed and the proposed schemes.

<table>
<thead>
<tr>
<th></th>
<th>Proposed scheme</th>
<th>Watershed with a threshold value of 20</th>
<th>Watershed with a threshold value of 30</th>
<th>Watershed with a threshold value of 40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmented objects</td>
<td>32</td>
<td>16</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>Error rates</td>
<td>15.79%</td>
<td>57.89%</td>
<td>71.05%</td>
<td>86.84%</td>
</tr>
</tbody>
</table>