Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 24 (2006) 1090–1102
image AND vision COMPUTING
www.elsevier.com/locate/imavis

A new edge detection approach based on image context analysis

Yuan-Hui Yu [a], Chin-Chen Chang [b,*]

[a] Department of Multimedia and Game Science, Southern Taiwan University of Technology, Tainan 710, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 40724, Taiwan, ROC

Received 12 January 2005; received in revised form 7 March 2006; accepted 31 March 2006

Abstract

A new adaptive edge detection approach based on image context analysis is presented in this paper. The proposed approach uses the information from predictive error values produced by the GAP predictor to detect edges. The experimental results indicate that both the visual evaluations and objective performance evaluations of the detected image in the proposed approach are superior to the edge detection of Sobel, Canny and the scheme presented by Tsai et al. [P. Tsai, C.C. Chang, Y.C. Hu, An adaptive two-stage edge detection scheme for digital color images, Real-Time Imaging 8 (4) (2002) 329–343]. To meet the needs of users, the flexibility in the threshold selection in the proposed approach is the same as that of the edge detection scheme [P. Tsai, C.C. Chang, Y.C. Hu, An adaptive two-stage edge detection scheme for digital color images, Real-Time Imaging 8 (4) (2002) 329–343]. The proposed approach, which is far more accurate than the detection scheme in [P. Tsai, C.C. Chang, Y.C. Hu, An adaptive two-stage edge detection scheme for digital color images, Real-Time Imaging 8 (4) (2002) 329–343], can precisely locate object contours in the image, especially for complex scenes. This feature, which the edge detection scheme [P. Tsai, C.C. Chang, Y.C. Hu, An adaptive two-stage edge detection scheme for digital color images, Real-Time Imaging 8 (4) (2002) 329–343] lacks, is of extreme importance to some applications such as data hiding, watermarking, morph, and pattern recognition. In addition, the approach can be integrated into a prediction-based lossless image compression scheme to provide both the lossless compression codes and edge maps of objects, which facilitate the image transmission and objects recognition for medical diagnoses and other applications.

Keywords: Edge detection; Predictive error value; GAP; Image context analysis

1. Introduction

Because of its significant importance in many research areas, edge detection has received much attention during the past two decades. Since, the edge is a prominent feature of an image, it is a vital foundation for image processing, computer vision, image understanding system, and pattern recognition. The detection results benefit applications such as image enhancement, recognition, morphing, restoration, registration, compression, retrieval, watermarking, hiding, etc. [10,16].

Edge detection is the process of locating edges in an image by detecting the prominent variations in intensity for a gray level image or in chrominance for a color image. Generally, an edge detection method can be divided into three stages. In the first stage, a noise reduction process is performed. In order to gain better performance of edge detection, image noise should be reduced as much as possible. This noise reduction is usually achieved by performing a low-pass filter because the additive noise is normally a high-frequency signal. However, the edges can possibly be removed at the same time because they are also high-frequency signals. Hence, a parameter is commonly used to make the best trade-off between noise reduction and edges information preservation. In the second stage, a high-pass filter such as a differential operator is usually employed to find the edges. In the last stage, an edge localization process is performed to identify the genuine edges, which are distinguished from those similar responses caused by noise. Thresholding techniques may be used to accomplish this process [14].

A variety of approaches for edge detection have been proposed for different purposes in different applications. Among the earliest works of edge detection are Sobel, Prewitt, Roberts, and Laplacian edge detectors, all of which use convolution masks to approximate the first or second derivative of an image [14]. In [1], an optimal filter for edge detection referred to Canny edge detector, was proposed. Three performance criteria defined for the optimality edge detection were good detection, good localization, and unique answer to a true edge.

* Corresponding author. Tel.: +886 4 23259100; fax: +886 4 23277425.
E-mail addresses: yhyu@mail.stut.edu.tw (Y.-H. Yu), ccc@cs.ccu.edu.tw (C.-C. Chang).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.03.006

In [4], a discrete singular (DSC) convolution algorithm for edge detection was proposed. Some DSC (low-pass) filters in the context of distribution theory have been introduced. In [2], a content-based image retrieval system based on a block-based edge detection algorithm was proposed, using moment-preserving techniques. In this algorithm, the edge detection technique was applied to detect a visually important edge for image retrieval. In [15], an edge detection algorithm based on least square support vector machine with Gaussian radial basis function kernel was proposed. And it uses both the gradients and the zero crossings to locate the edge positions.

In [8], an unsupervised learning algorithm based on the wavelet analysis was proposed to detect image edges. And a wavelet domain vector hidden Markov tree model was accordingly established to capture the statistical relationships between wavelet coefficients at different scales and at different subbands. With this model, each wavelet coefficient is viewed as an observation of its hidden state, which indicates if the wavelet coefficient belongs to an edge. In [12], an edge detection method based on a cellular automata model was proposed. A uniform cellular automation rule using a von Neumann neighborhood was employed to carry out edge detection on binary and grayscale images. A synchronization-based image edge detection was introduced in [11]. In this scheme, image edge detection is viewed as an action of a partial differential equation with suitable restrictions. The result is satisfactory for texture images.

In [6], a lossless image compression method based on least-square-based adaptive prediction scheme was proposed. According to the analysis in this paper, it is edge-directed property of the least square optimization that contributes to the high compression performance. In [10], an adaptive two-stage edge detection scheme for digital color images was presented. In the first stage of this scheme, a color image is quantized to a grayscale image using moment-preserving thresholding technique. Then, an edge detection technique based on block truncation coding was employed to detect edges in the second stage.

Among these edge detectors, Sobel edge detector can be regarded as the historic 'standard', while Canny edge detector as a modern 'standard'. They are still employed quite often to compare with those proposed methods in publications today [3].

Although several traditional approaches for edge detection have been devised, existing methods seldom provide satisfactory edge detection results for any real or synthetic images [3]. We shall propose a simple and effective approach based on image context analysis. The proposed approach provides a more satisfactory edge detection results than Sobel, Canny edge detectors and the scheme in [10]. The proposed approach can be integrated into a prediction-based lossless image compression scheme to provide both the lossless compression codes and edge map of objects, which facilitate the image transmission and objects recognition for medical diagnoses or radiologists' studies.

The image context analysis in the proposed approach is based on gradient-adjusted prediction (GAP), which is adopted in context-based adaptive lossless image coding (CALIC). GAP predictor uses a context, which is a combination of the intensity values of already processed neighboring pixels defined by a template, to produce the predictive values. The context in the casual template of GAP is used to analyze whether the current pixel is an edge point or not. The proposed approach is much more accurate in analyzing the differences between the proposed edge detector and the others due to the use of the casual template.

The remainder of this paper is organized as follows. In Section 2, GAP and three related works are briefly described. In Section 3, the proposed approach is presented. The experimental results are given in Section 4. Finally, conclusions are drawn in Section 5.

2. Related works

In this section, we shall introduce the basic concept of GAP and briefly review the Sobel edge detector, Canny edge detector, and the edge detector presented in [10].

2.1. GAP predictor

The GAP is a nonlinear predictor adopted by the context-based, adaptive, lossless image codec (CALIC). It can make itself adapt to the intensity gradients near the pixel to be predicted [13]. Fig. 1 shows the casual template employed in GAP predictor. The template involves two previous scan lines of coded pixels. The neighboring pixels of the current pixel x used in the prediction process are shaded.

Here, the notations i, j, k, l, m, n, o, and x represent not only the pixel values, but their locations as well. GAP uses g_h and g_v to compute the gradient of intensity near the current pixel in the horizontal direction and in the vertical direction, respectively. The g_h and g_v are calculated as

$$g_h = |i-m| + |j-k| + |j-l| \tag{1}$$

and

$$g_v = |i-k| + |j-n| + |l-o|. \tag{2}$$

The detection of the magnitude and orientation for an edge across the casual template is based on the difference of g_h and g_v. Finally, the gradient-adjusted prediction procedure that produces the predictive values is shown below.

if $(g_v - g_h > 80) \bar{x} = i$ /*sharp horizontal edge*/

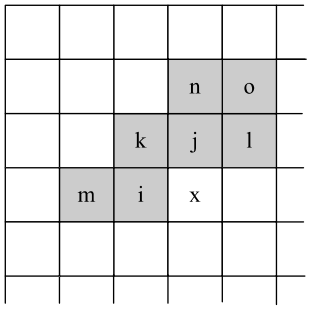

Fig. 1. Casual template used by GAP.

```
else if (g_v − g_h < −80) x̄ = j                    /*sharp vertical edge*/
  else{
    x̄ = (i + j)/2 + (l − k)/4                      /*smooth area*/
    if (g_v − g_h > 32) x̄ = (x̄ + i)/2              /*horizontal edge*/
    else if (g_v − g_h > 8) x̄ = (3x̄ + i)/4         /*weak horizontal edge*/
    else if (g_v − g_h < −32) x̄ = (x̄ + j)/2        /*vertical edge*/
    else if (g_v − g_h < −8) (3x̄ + j)/4            /*weak vertical edge*/
  }
```

To decode an image from the compression code, a reverse prediction is performed which is just the reverse process of the above-mentioned encoding process. The decoded pixel values are recovered from after adding the predictive values obtained from the reverse prediction process and the opposite prediction error values from the compression codes.

2.2. Sobel edge detector

In Sobel edge detector, the task of edge detection is fulfilled via performing a 2D spatial gradient convolution operation on an image. It uses the following two convolution kernels (masks) k_x and k_y, as shown in Fig. 2, to estimate the gradient g_x and g_y in horizontal and vertical directions, respectively.

Here, g_x and g_y are computed as

$$g_x = -1w_1 + 1w_3 - 2w_4 + 2w_6 - 1w_7 + 1w_9, \tag{3}$$

$$g_y = 1w_1 + 2w_2 + 1w_3 - 1w_7 - 2w_8 - 1w_9, \tag{4}$$

where w_i, $i = 1,2,\ldots,9$ are intensity levels of each pixel in the convolution window shown in Fig. 3. The gradient magnitude g is computed as

$$g = \sqrt{g_x^2 + g_y^2}. \tag{5}$$

–1	0	1
–2	0	2
–1	0	1

(a) convolution kernel k_x

–1	–2	–1
0	0	0
1	2	1

(b) convolution kernel k_y

Fig. 2. The masks of the Sobel edge detector (a) convolution kernel k_x (b) convolution kernel k_y.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fig. 3. A 3×3 convolution window.

To increase the speed of computation, alternatively, g can be approximately computed as

$$g = |g_x + g_y|. \tag{6}$$

Finally, the gradient magnitude is thresholded.

Sobel edge detector is a simple and effective approach to find edges in an image. However, it is sensitive to noise in the image. Moreover, the detected edges are thick, which may not be suitable for some applications where the detection of the outmost contour of an object is required.

2.3. Canny edge detector

The Canny edge detector was devised to be an optimal edge detector, which satisfies all of the three performance criteria. The first criterion is to minimize the situations of detecting false edges and missing actual edges. The second criterion is to minimize the distance between the detected edges and the actual edges. The third criterion is to minimize multiple responses to an actual edge, i.e. to ensure there is only one response for an actual edge point.

Guiding with these criteria, the Canny edge detector was designed in a multi-stage process. The first step of this detector is to filter out noise in the image by using a Gaussian smoothing filter. The second step is to locate the edge strength in the smoothed image by computing the image gradient, which helps to indicate where the edge is. A two-dimensional first derivative operator, such as Sobel, may be employed to highlight the maximum of the first derivative where those edges are located.

The last step is to thin down the edges by tracking along the edge in the edge direction and set any pixel that is not at the maximum to be 0, which is called non-maximum suppression. The pixel set to be 0 is not regarded as an edge. The whole tracking process is controlled by the use of two thresholds: *low* and *high*. If the magnitude is above the *high* threshold, it is considered as an edge. If the magnitude is below the *low* threshold, it is considered as a non-edge. If the magnitude is between the two thresholds, it is set to be 0 except that there is a path from an edge pixel to a pixel whose magnitude is above the *low* threshold [14].

2.4. Tsai et al.'s edge detector

In 2004, Tsai et al. proposed an adaptive two-stage edge detection scheme for digital color images based on the absolute moment block truncation coding (AMBTC) technique. In the first stage, a moment-preserving thresholding technique presented in [9] was exploited to reduce the dimension of a color image, i.e. the red, green, and blue color components of the input image are reduced to a one-dimensional grayscale image. In the second stage, an edge detection procedure based on AMBTC was employed to detect edges.

Originally, AMBTC is a lossy image compression technique [5]. In the encoding procedure of AMBTC, the input image is first partitioned into non-overlapping blocks of size $n \times n$ (for instance 4×4 pixels). Then pixel values of each block are used

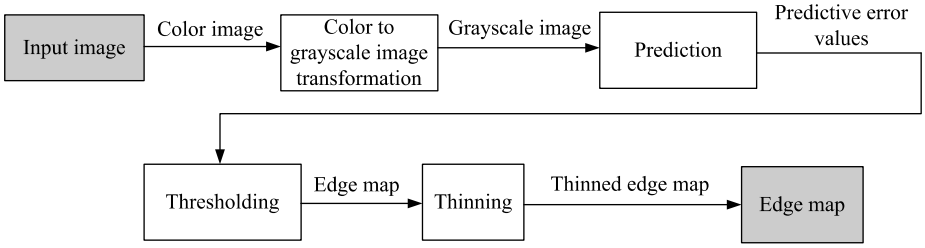

Fig. 4. The flowchart of the proposed approach.

2	6	8	9
5	7	8	200
8	9	215	201
7	210	218	210

(a) Original image

11	12	16	16
12	15	17	199
16	18	216	204
16	218	215	208

(b) Predictive pixel values

–9	–6	–8	–7
–7	–8	–9	1
–8	–9	–1	–3
–9	2	3	2

(c) Predictive error values

0	0	0	0
0	0	0	255
0	0	255	255
0	255	255	255

(d) Edge map

Fig. 5. An example of the prediction and thresholding process.

to determine two quantization levels, *low* and *high*, and to produce a bit plane B, which saves the quantization values for each pixel in each block. The triple (*low*, *high*, B) representing an encoded block is used to reconstruct the approximate pixel values (*low* or *high*) in the decoding procedure.

The bit plane B is generated according to the mean value $\bar{m}$ of an image block. A bit value of 1 is set in B if a pixel whose value is greater than or equal to $\bar{m}$. Contrarily, a value of 0 is set in B if a pixel whose value is less than $\bar{m}$. Let BS denote the block size. The two quantization levels, *low* and *high*, are computed as follows

$$low = \bar{m} - \frac{\bar{\sigma}}{2} \times \frac{BS}{BS - g}, \tag{7}$$

$$high = \bar{m} + \frac{\sigma}{2} \times \frac{BS}{g}. \tag{8}$$

Here, $\bar{\sigma}$ is defined as

$$\bar{\sigma} = \frac{1}{BS} \sum_{i=1}^{BS} |x_i - \bar{m}|, \tag{9}$$

where g and x_i denote the number of pixels whose values are greater than or equal to $\bar{m}$, and the ith pixel value of each image block, respectively.

In this scheme, the information of B is used to detect edge boundaries. To detect edges, a horizontal and a vertical scan of all bit planes for the input image are separately performed. The places where the variations of values are from 0 to 1 or from 1 to 0 are marked as edge points (labeled 0) during the stage of scan. Otherwise, no marking is made (labeled 1). Finally, the edge map corresponding to the input image is determined from the result of merging the horizontal and the vertical directional edge blocks. The merging operation is defined as the logical AND of the horizontal and vertical corresponding edge blocks.

3. The proposed approach

In this section, we shall present the proposed edge detection scheme based on the predictive coding technique. The flowchart of the proposed approach is shown in Fig. 4.

The proposed approach consists of four main procedures, color to grayscale image transformation, prediction, thresholding, and thinning. The task of edge detection is executed pixel

0	0	0	0
0	0	0	255
0	0	255	255
0	255	255	255

(a) Original edge map

255	255	255	255
255	255	0	255
255	0	255	255
0	255	255	255

(b) Edge map after horizontal scan

255	255	0	255
255	0	0	255
0	0	255	255
255	255	255	255

(c) Edge map after vertical scan

255	255	0	255
255	0	0	255
0	0	255	255
0	255	255	255

(d) Edge map after combination

Fig. 6. An example of the thinning process.

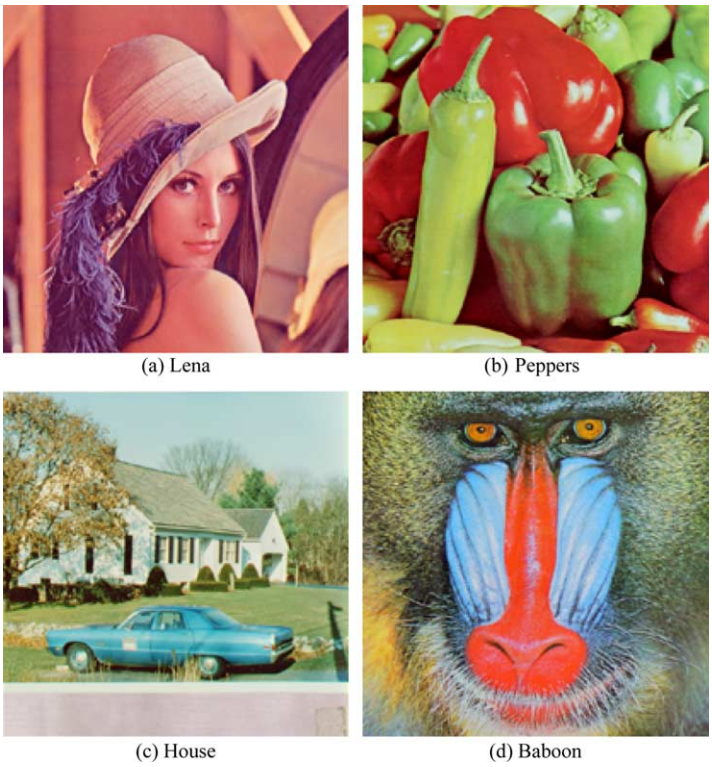

Fig. 7. The four original color images.

Fig. 8. Edge maps obtained by using the proposed approach with various TH_2 values (a) (TH_2 = 10%, FOM = 0.3924) (b) (TH_2 = 30%, FOM = 0.8509) (c) (TH_2 = 40%, FOM = 0.8276) (d) (TH_2 = 50%, FOM = 0.7822) (e) (TH_2 = 60%, FOM = 0.7438) (f) (TH_2 = 70%, FOM = 0.6998) (g) (TH_2 = 80%, FOM = 0.6547) (h) (TH_2 = 90%, FOM = 0.6100) (i) (TH_2 = 100%,FOM = 0.5638).

by pixel. First, the input color image is transformed into grayscale image. Then the image is fed into the GAP predictor to produce predictive error values. The prediction is progressed in a predefined scan order. The casual template used by GAP predictor in image compression is proved to be effective in reducing the prediction errors, especially on the smooth area of an image [13]. The template is effective to indicate the positions of edge points in edge areas when the predictive error values are relatively larger than those in smooth areas because GAP predictor uses gradient of intensity to detect the magnitude and orientation for an edge across the casual template.

A fixed threshold TH_1 is used in the thresholding procedure in the proposed approach to classify the pixels in the image into two groups. In other words, if a pixel whose corresponding absolute value of the predictive error value is greater than or equal to TH_1, it is classified as an edge point (marked as 0). Otherwise, it is classified as a non-edge point (marked as 255). As a result, an edge map is obtained.

To illustrate the prediction and thresholding process, an example is shown in Fig. 5. Assume that an original color image has been transformed into a grayscale image. The pixel values of this grayscale image are shown in Fig. 5(a). The predictive pixel values and predictive error values produced by GAP of this image are shown in Fig. 5(b)–(c), respectively. Assume TH_1 is set to 5. After thresholding of the predictive pixel error values, the edge map is obtained, as shown in Fig. 5(d).

To thin the edges, a thinning procedure is involved in the proposed approach. The intensity variations of the original edge map are first scanned in horizontal and vertical directions. Then a combination operation of the resultant edge maps is performed to produce the final edge map. The thinning approach is similar to that of the scheme in [10]. However, there are differences in the mark places and the combination operation.

Fig. 6 shows an example of the thinning process. The original edge map is shown in Fig. 6(a). The order of the horizontal scan is from left to right and from top to bottom. When scanning mark the place, where an intensity variation is from 0 to 255 or 255 to 0, to be 0. Otherwise, mark the place 255. We will take the first row as an example. The intensity values in the first two columns are 0 and 0, which indicate no edge point in the vertical direction. Thus, the intensity value of the first column is marked as 255. Likewise, the intensity values of the second and third columns are all 255's. The last

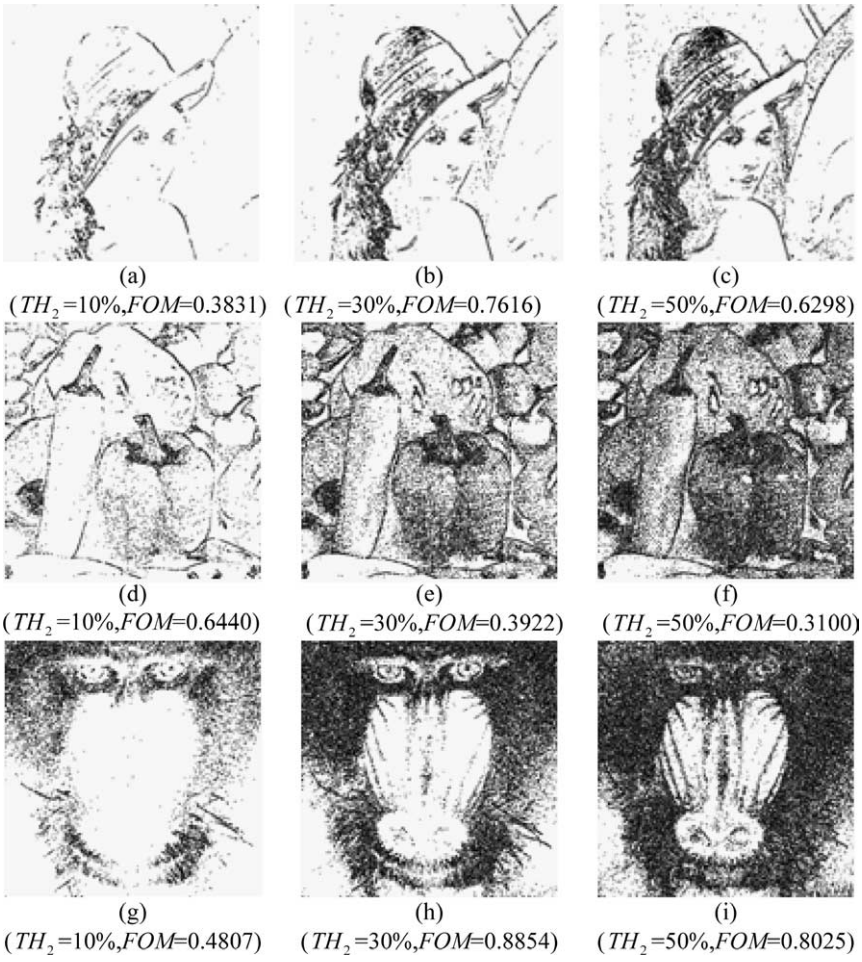

Fig. 9. Obtained edge maps from different type of images by using the proposed approach with TH_2=10%, 30, and 50% (a) (TH_2=10%, FOM=0.3831) (b) (TH_2=30%,FOM=0.7616) (c) (TH_2=50%,FOM=0.6298) (d) (TH_2=10%,FOM=0.6440) (e) (TH_2=30%,FOM=0.3922) (f) (TH_2=50%,FOM=0.3100) (g) (TH_2=10%,FOM=0.4807) (h) (TH_2=30%,FOM=0.8854) (i) (TH_2=50%,FOM=0.8025).

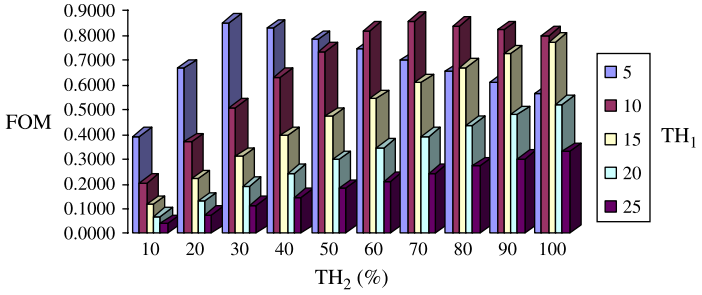

Fig. 10. Experiments on the influence of *FOM* values when parameters TH_1 and TH_2 are varied for the House image.

column is always marked as 255. Fig. 6(b) shows the result of the horizontal scan of edge points.

The order of the vertical scan is from top to bottom and from left to right. The scanning method is similar to the horizontal scan. Let us also take the first column as an example. The intensity values in the first two rows in Fig. 6(b) are 255 and 255, which indicate no edge point in the horizontal direction. Thus, the intensity value of the first row is marked to be 255. Likewise, the intensity value of second row is marked as 255. The intensity values in the third and fourth rows are 255 and 0, which indicate an edge point in the horizontal direction. Thus, the intensity value of the third row is marked as 0. The last row is always marked as 255. Fig. 6(c) shows the results of the vertical scan of edge points.

To obtain the final edge map, the edge maps after the horizontal and vertical scanning of edge points are combined by using the OR logical operation. For example, because the intensity values in the first row and column in Fig. 6(b) and (c) are all 255's, the result of the OR operation is 255. Fig. 6(d) shows the final edge map resulting from the combination of Fig. 6(b) and (c).

The proposed approach is an adaptive thresholding edge detection method. The incorporation of the adaptive threshold selection in the edge detection process is used to avoid the unsatisfactory visual results usually caused by a fixed threshold edge detection scheme [10]. The number of edge points generated in thresholding procedure can suit the requirements of different applications. Practically, users accept the visual results of an edge detector according to their subjective evaluation [3]. The percentage of edge points presented in the resultant edge map of the thresholding procedure can be adjusted on users' judgment.

The example demonstrated in Fig. 5 shows 100% edge points were presented in Fig. 5(d). The proposed edge detection approach picks up the numbers of the predictive error values

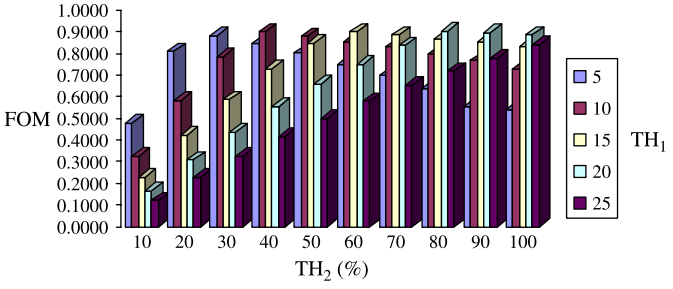

Fig. 11. Experiments on the influence of *FOM* values when parameters TH_1 and TH_2 are varied for the Baboon image.

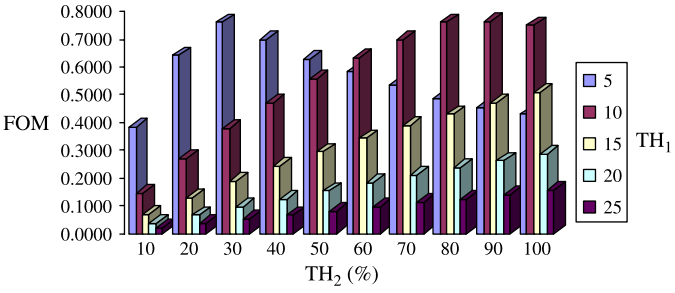

Fig. 12. Experiments on the influence of *FOM* values when parameters TH_1 and TH_2 are varied for the Lena image.

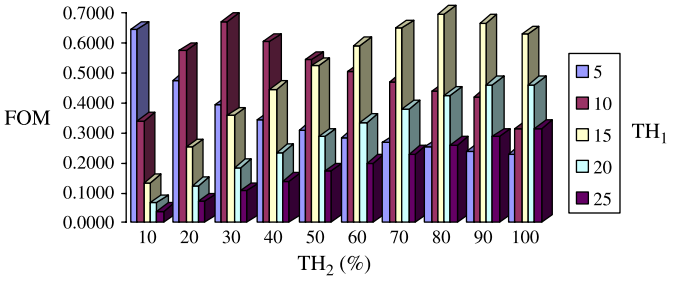

Fig. 13. Experiments on the influence of *FOM* values when parameters TH_1 and TH_2 are varied for the Peppers image.

Table 1
Obtained *FOM* values from the test images by the proposed approach with thresholds $TH_1 = 5$, and $TH_2 = 30\%$

Images	*FOM*
Boat	0.8391
Goldhill	0.6604
Zelda	0.6077
Man	0.7609
Crowd	0.7804
Couple	0.7226
Barbara	0.6353
Airplane	0.8069
Toys	0.8388
Sailboat	0.7025

Fig. 14. Four edge maps obtained from the test images by using the proposed approach (a) Boat ($FOM=0.8391$) (b) Airplane ($FOM=0.8069$) (c) Couple ($FOM=0.7226$) (d) Zelda ($FOM=0.6077$).

whose absolute values are greater than or equal to the threshold TH_1 (i.e. edge points) according to the order from the maximum value to the value TH_1 with a user's defined percentage of edge points TH_2. Generally, when TH_1 is fixed, using lower percentage of edge points can avoid noise effect but may lose some meaningful edges. While using higher percentage of edge points can produce more fine details of objects than that with lower percentage but may produce too

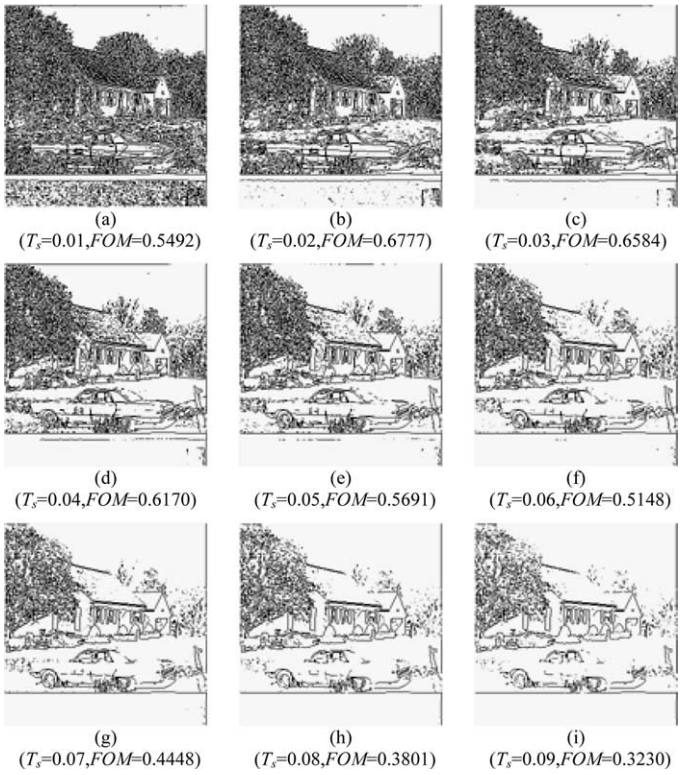

Fig. 15. Edge maps obtained by using the Sobel edge detector with various thresholds (a) ($T_s=0.01$, $FOM=0.5492$) (b) ($T_s=0.02$, $FOM=0.6777$) (c) ($T_s=0.03$, $FOM=0.6584$) (d) ($T_s=0.04$, $FOM=0.6170$) (e) ($T_s=0.05$, $FOM=0.5691$) (f) ($T_s=0.06$, $FOM=0.5148$) (g) ($T_s=0.07$, $FOM=0.4448$) (h) ($T_s=0.08$, $FOM=0.3801$) (i) ($T_s=0.09$, $FOM=0.3230$).

many insignificant edges. In the proposed edge detection approach, an adequate percentage of edge points can be chosen to pleasingly present the object contour. If necessary, adaptive choice of parameters can also be automatically determined by using the information of edge point numbers in the given image to avoid trial-and-error testing occurred frequently in a conventional edge detector.

4. Experimental results

To evaluate the performance of the proposed approach, several experiments based on some standard test images were conducted. Four color host images—'Lena,' 'Baboon,' 'Peppers,' and 'House' of 512×512 pixels are shown in Fig. 7.

In our experimental approach, we employed subjective and objective performance measures (visual evaluation and Pratt's figure of merit) to evaluate the proposed method and other comparable methods. Pratt's figure of merit (FOM) presented in [7] is a widely used objective standard to rate the quality of edge detection systems and is defined as

$$FOM = \frac{1}{\max(N_I, N_T)} \sum_{i=1}^{N_T} \frac{1}{1 + \alpha d_i^2}, \tag{10}$$

where N_I and N_T are the numbers of ideal and detected edge pixels, respectively. α denotes a penalty constant (often 1/9) used to penalize displaced edges. d_i represents the distance between an alleged edge point and the nearest ideal edge pixel. Domain experts provide ideal edge maps. The value of FOM is a number between 0 and 1. The larger the FOM value is, the better the performance will be, and the converse applies.

The original color images are first transformed into grayscale images. Then each grayscale image is fed into the GAP predictor to produce predictive error values. Those predictive error values whose absolute values are greater than or equal to TH_1 are classified as edge points. Otherwise, they are classified as non-edge points.

To demonstrate the adaptive threshold selection in the proposed approach, one of the four images (House) shown in Fig. 7, was chosen to make comparisons of the visual

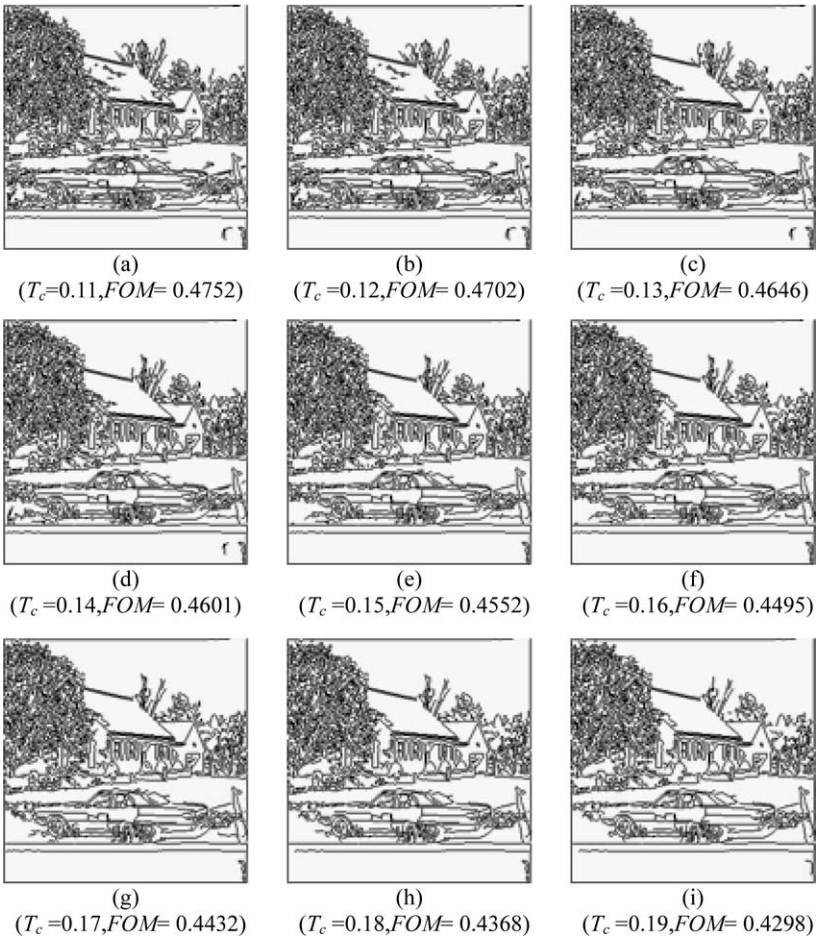

Fig. 16. Edge maps obtained by using the Canny edge detector with various thresholds (a) (T_c=0.11, FOM=0.4752) (b) (T_c=0.12, FOM=0.4702) (c) (T_c=0.13, FOM=0.4646) (d) (T_c=0.14, FOM=0.4601) (e) (T_c=0.15, FOM=0.4552) (f) (T_c=0.16, FOM=0.4495) (g) (T_c=0.17, FOM=0.4432) (h) (T_c=0.18, FOM=0.4368) (i) (T_c=0.19, FOM=0.4298).

evaluation and *FOM*. In this experiment, the threshold TH_1 is set to 5. The edge maps with 20, 30, 40, 50, 60, 70, 80, 90 and 100% edge points are shown in Fig. 8.

The edge map with the highest *FOM* value of 0.8509 is the edge map shown in Fig. 8(b). Visual evaluation of Fig. 8(a) shows that the use of TH_2 at 10% is not sufficient to present the objects. In contrast, object contours can be extracted when larger TH_2 values are used such as in Fig. 8(c)–(i). However, exceedingly fine textures appear in those edge maps making contours difficult to observe.

In order to observe the influence of changing the TH_2 on different types of images, we performed several experiments with $TH_1=5$ and the threshold TH_2 set at 10, 30, and 50%, respectively. Three of the four images shown in Fig. 7 were used in the experiments. Fig. 9 shows the experimental results.

The results indicate the object contours of a smooth image containing simple edge structure such as the Peppers image can be well identified with the use of $TH_2=10\%$. Both the visual evaluation and the *FOM* value are satisfactory. For a figure image such as Lena, and a complex-scenes image such as Baboon, using $TH_2=10\%$, is insufficient. By increasing TH_2 to 30 or 50%, the object contours become clearly identified. Both the visual evaluation and the *FOM* value are satisfactory. These experimental results are roughly in accordance with the observation presented in [10].

Hundreds of experiments were performed to explore the effects of varying the TH_1 and TH_2 parameters on different types of images. We increased TH_2 in steps of 10% starting from 10% through 100%, and simultaneously varied TH_1 from 5 to 25 in step of 5. The results of these variations for the House, Baboon, Lena, and Peppers images are reported in *FOM* values, and depicted in Figs. 10–13. In the proposed approach, users can choose the best pair of TH_1 and TH_2 thresholds based on the *FOM* values shown in Figs. 10–13 to adapt to application requirements. Figs. 10–13 reveal that satisfactory edge maps can be achieved with the parameter TH_2 set in the range between 30 and 50% for non-smooth images such as House, Lena, and Baboon, and with the TH_2 set to 10% for smooth images such as Peppers, when TH_1 is set to 5.

To verify this observation, we tested several other standard images using the parameters $TH_1=5$, and $TH_2=30\%$. Visual evaluation and *FOM* values are satisfactory. Table 1 shows the obtained *FOM* values from edge maps of these images. Examples of the four obtained edge maps are shown in Fig. 14.

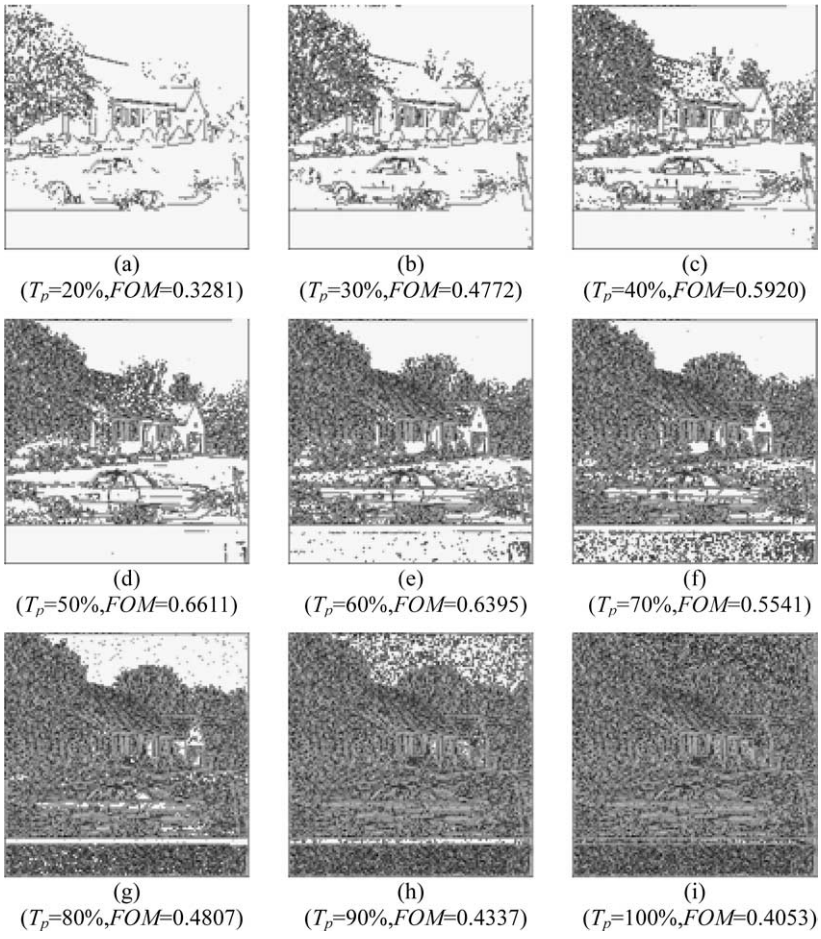

Fig. 17. Edge maps obtained by using the edge detection scheme in [10] with various thresholds (a) ($T_p=20\%$, $FOM=0.3281$) (b) ($T_p=30\%$, $FOM=0.4772$) (c) ($T_p=40\%$, $FOM=0.5920$) (d) ($T_p=50\%$, $FOM=0.6611$) (e) ($T_p=60\%$, $FOM=0.6395$) (f) ($T_p=70\%$, $FOM=0.5541$) (g) ($T_p=80\%$, $FOM=0.4807$) (h) ($T_p=90\%$, $FOM=0.4337$) (i) ($T_p=100\%$, $FOM=0.4053$).

Fig. 18. Gaussian noise corrupted image.

To compare the performance of the proposed approach with three others, namely Sobel, Canny, and the edge detection scheme in [10], several experiments were conducted with various parameters. These experiments show the best edge maps produced by the proposed approach are superior to the other methods as measured by visual evaluation and *FOM* values. Figs. 15–17 show some of the edge maps produced from Sobel, Canny, and the edge detection scheme in [10]. We employed Matlab software to obtain the Sobel and Canny edge maps. The parameters T_s and T_c shown in Figs. 15 and 16 represent the thresholds used in the Matlab Sobel and Canny 'edge' function, respectively. The parameter T_p shown in Fig. 17 stands for the percentage of edge blocks used in the edge detection scheme in [10].

After carefully observe the edge maps in Fig. 15, we find that some significant textures are missing when the Sobel edge detector is used. Furthermore, the detected edges are thicker. Using the Canny edge detector gives distorted edges (Fig. 16). Too many edges around objects are detected. Thus, it is not easy to identify, which lines are real contour edges. Using the scheme in [10] results in the edges around objects being distorted. The best *FOM* values from the edge maps produced

Fig. 19. Edge map of Fig. 18 obtained from the Sobel edge detector (*FOM* = 0.1050).

Fig. 20. Edge map of Fig. 18 obtained from the Canny edge detector (*FOM* = 0.3191).

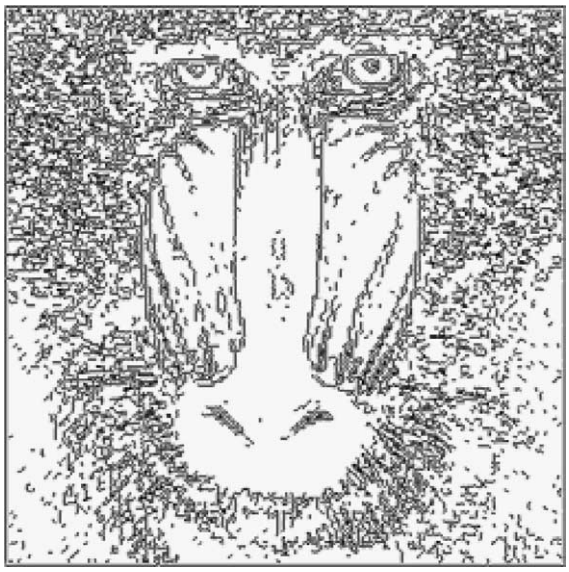

Fig. 21. Edge map of Fig. 18 obtained from the scheme in [10] ($FOM = 0.4107$).

by Sobel, Canny, and the edge detection scheme in [10], are unsatisfactory. Visual comparison as well as comparing *FOM* values, clearly show that the results of the proposed approach are superior to any of the other methods.

To explore the noise effect on edge detectors, several experiments were conducted on Gaussian noise corrupted images. Once again the Sobel edge detector, the Canny edge detector, the edge detection scheme in [10] and the proposed approach were compared. In one example, the image shown in Fig. 7(d), was corrupted with Gaussian noise by applying the 'imnoise' function with a mean of 0 and variance of 0.01 in Matlab. The Gaussian noise corrupted image is shown in Fig. 18.

The parameters TH_1 and TH_2 were set to 5 and 50%, respectively, and used in the computation of the edge map shown in Fig. 22. By increasing TH_2 to 60, 70, 80, 90, and 100%, the *FOM* values are promoted to 0.6267, 0.6967, 0.7636, 0.8326, and 0.8134, respectively. Experimental results shown in Figs. 19–22 indicate both the visual evaluation and the *FOM* evaluation of the proposed approach are better than the edge detection methods of Sobel, Canny or that found in [10].

Fig. 22. Edge map of Fig. 18 from the proposed approach with $TH_1 = 5$ and $TH_2 = 50\%$ ($FOM = 0.5342$).

5. Conclusions

A new adaptive edge detection approach based on image context analysis is presented in this paper. The proposed approach uses the GAP predictor to produce predictive error values, which can indicate edge points if the absolute value of the predictive error values are greater than or equal to a predefined threshold. The context of the neighboring pixels near the current pixel to be predicted in a casual template, is used to analyze whether the current pixel is an edge point or not. The approach can precisely locate object contours in the image, especially for complex scenes. This advantage is critical to some applications such as data hiding, watermarking, morph, and pattern recognition.

Experimental results demonstrate that both the subjective and objective (visual evaluation and Pratt's Figure of Merit) measurements of the detected image in the proposed approach are better than the edge detectors of Sobel, Canny, and that presented in [10]. The proposed approach has the flexibility of allowing users to choose appropriate parameters to meet application requirements.

Furthermore, the approach can be integrated into a prediction-based lossless image compression scheme to provide both the lossless compression codes and the edge maps of objects, which facilitate the image transmission and objects recognition for medical diagnoses and other applications. Since, the proposed approach is simple and effective, it is as efficient to implement as the Sobel edge detector, the Canny edge detector, and the edge detection scheme in [10].

References

[1] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.
[2] S.C. Cheng, Content-based image retrieval using moment-preserving edge detection, Image and Vision Computing 21 (9) (2003) 809–826.
[3] M. Heath, S. Sarkar, T. Sanocki, K. Bowyer, Comparison of edge detectors: a methodology and initial study, Computer Vision and Image Understanding 69 (1) (1998) 38–54.
[4] Z.J. Hou, G.W. Wei, A new approach to edge detection, Pattern Recognition 35 (7) (2002) 1559–1570.
[5] M.D. Lema, O.R. Mitchell, Absolute moment block truncation coding and its application to color images, IEEE Transactions on Communications 32 (10) (1984) 1148–1157.
[6] X. Li, M.T. Orchard, Edge-directed prediction for lossless compression of natural images, IEEE Transactions on Image Processing 10 (6) (2001) 813–817.
[7] W.K. Pratt, Digital Image Processing, Wiley, New York, 1978.
[8] J. Sun, D. Gu, Y. Chen, S. Zhang, A multiscale edge detection algorithm based on wavelet domain vector hidden markov tree model, Pattern Recognition 37 (7) (2004) 1315–1324.

[9] W.H. Tsai, Moment-preserving thresholding: a new approach, Computer Vision, Graphics, and Image Processing 29 (3) (1985) 377–393.
[10] P. Tsai, C.C. Chang, Y.C. Hu, An adaptive two-stage edge detection scheme for digital color images, Real-Time Imaging 8 (4) (2002) 329–343.
[11] G.W. Wei, Y.Q. Jia, Synchronization-based image edge detection, Europhysics Letters 59 (6) (2002) 814–819.
[12] S. Wongthanavasu, R. Sadananda, A CA-based edge operator and its performance evaluation, Journal of Visual Communication and Image Representation 14 (2) (2003) 83–96.
[13] X. Wu, N. Memon, Context-based, adaptive, lossless image coding, IEEE Transactions on Communications 45 (4) (1997) 437–444.
[14] T.J. Yen, A qualitative profile-based approach to edge detection, PhD Thesis, New York University, 2003.
[15] S. Zheng, J. Liu, J.W. Tian, A new efficient SVM-based edge detection method, Pattern Recognition Letters 25 (10) (2004) 1143–1154.
[16] D. Ziou, S. Tabbone, Edge detection techniques—an overview, Pattern Recognition and Image Analysis 8 (4) (1998) 537–559.