Detection of Arbitrary Triangle

Hong-Min Liu
School of Computer Science and Technique
Henan Polytechnic University
Jiaozuo, China

Zhi-Heng Wang
School of Computer Science and Technique
Henan Polytechnic University
Jiaozuo, China

***Abstract*—This paper presents a method for the detection of arbitrary triangle based on the property that the distance of the incenter to the edges of the triangle is equidistant and equal to the radius of the inscribed circle. The method is constructed by the following steps. Firstly, the orientation line of each edge point is computed from the edge-only image. Secondly, the energy and the feature length of each pixel are calculated to obtain the feature length distribution map and energy distribution map of the whole image. Then the latter is made use to detect the candidate incenters of the triangles by local maximum detection. Subsequently, the true incenters are chose from the candidates according to the ratio of the energy and the feature length of the pixel. Finally, the orientation distribution of the equidistant edge points is considered and non-triangle is eliminated. Synthetic images and real images are used to prove the capability of the proposed method for detection of triangles and the experiments show good accuracy despite the presence of noises and Gaussian blurring.**

Keywords- triangle detection; orientation line; inscribed circle; energy distribution.

I. INTRODUCTION

Geometrical shape detection is of importance in many applications, such as automatic inspection and assembly in industrial applications, for locating complex objects or tracking them. Triangle is a common figure and is seen frequently in daily scenes, such as the traffic signs. So the detection of triangle is significant for the objects detection and recognition.

Hough transform (HT) based methods [1-5] is the common method for the detection of the triangle which shows enough robust but along with the significant computation and large storage requirement. Other non-HT based methods were also provided. Barnes et al. presented a robust regular polygon detector [6]. This method first derived the α posteriori probability for a mixture of regular polygons and thus the probability density function for the appearance of a set of regular polygons. Then a complete formulation using maximum likelihood was derived for the locations of the most likely polygons. Siddharth and David proposed an efficient method that finds partial and complete matches to models for families of polygons in fields of corners extracted from images [7]. This method provided the polygon models assigned specific values of acuteness to each corner in a fixed-length sequence along the boundary in advance, then the geometrical similarity of the detected figures were compared with the models and specific polygons were detected. Shi et al. presented an algorithm for recognizing geometrical shapes automatically [8]. This method got the energy image and directional angle image by the tunable filter from the original image, and then the angle and the line number were calculated from the angle and perpendicular histogram to recognize the polygons. In addition, other methods were also provided for the detection of polygons [9-10].

The common methods for detection of the triangles are HT based methods [1-5], which have large computation. Other methods, which either pointed at the detection of specific polygons [6], or needed to be provided the information of the polygons [7-11], are not universal for the detection of triangles. In our work, a universal algorithm is presented for the detection of arbitrary triangles. The method is based on the property of the inscribed circle of the triangle. For an arbitrary triangle, it has one unique inscribed circle. The distance of the incenter to the edges of the triangle is equidistant and equal to the radius of the inscribed circle. Inspired by this, a new technical term, orientation line, is defined, which is consistent with the orientation of the edge and passes through the edge point itself. The edge points constructing the triangle are obtained by means of comparing the distance between its orientation line and the incenter with the radius of the inscribed circle. The detected triangle is represented with the edge points constructing it. Compared with the HT-based algorithm, the method proposed here avoids the voting processing, and raises the computation efficiency and results in an accurate detection of the arbitrary triangles. We present the application results of the provided method to both synthetic and real images.

The remainder of this paper is organized as follows. Section 2 introduces the principle of the method. Section 3 gives a detailed description for this method in steps. Section 4 displays the application results of the method on both synthetic and real images. The conclusions are presented in Section 5.

II. PRINCIPLE OF THE METHOD

The inscribed circle of a triangle is used for the triangle detection. As Fig.1 shows, for an arbitrary triangle, an inscribed circle exists and its center is denoted as c_0. For three random edge points, such as P_1, P_2 and P_3, the orientation lines, defined in detail in the following, of them are denoted as l_1, l_2 and l_3 respectively, which are coincident with the gradient orientations of the edge on which the edge points lying. According to the well-known property that the center of the inscribed circle is equidistant from the sides of the polygon and the definition of orientation line, the distances of the incenter to the orientation lines of edge points of a triangle are also equidistant and equal to the radius of the inscribed circle.

For each pixel in the image, the distances of the pixel to the orientation lines of edge points included in a local neighborhood are computed, and the distance distribution is used to determine the potential incenter of the triangle. Fig. 2 shows the distance distribution histogram of different points. As Fig. 2 (a) shows, the appointed points A, B and C are at different positions of a triangle, and the point C is the incenter of the triangle. The red circles display the pixel support region (PSR) centering on each point. The orientation line of each edge point is computed first. Then, the distances of the center pixel to the orientation lines of the edge points encircled in its PSR are computed. In the PSR of the point A, two edges of the triangle are encircled. For the edge points on one edge of the triangle, the distances computed are equidistant and equal to the distance of the point A to the edge. So two peaks are found in the distance distribution histogram of point A, and one peak is larger than the other obviously due to the different distribution of edge points, as Fig. 2 (b) shows. For the point B, three edges are encircled in its PSR, and three peaks are displayed in the distance distribution histogram of point B as Fig. 2 (c) displays. The three peaks are corresponding to the distances of the point B to the edges of the triangles respectively. For the incenter point C of the triangle, all the edge points are included in its PSR. In theory, the distances of the point C to the orientation lines of the edge points are equidistant and equal to the radius of the circle. In practice, the distance distribution histogram of the point C is shown in Fig. 2 (d) and an obvious peak is observed at the distance equal to the radius. This is coincident with the theoretical deduction. It can be seen that, with the different positions of the points relative to the triangle, the distance distribution histogram of each point is different. When the point is the incenter of a triangle, it can get an obvious maximum at the radius in the distance distribution histogram. This is the basis idea of our method for detecting the arbitrary triangle.

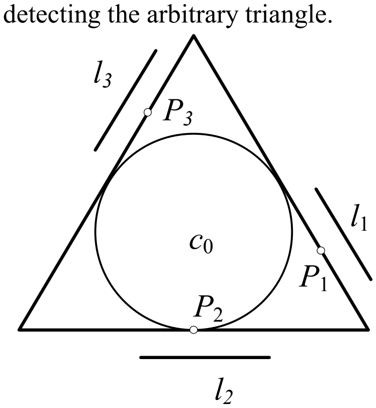
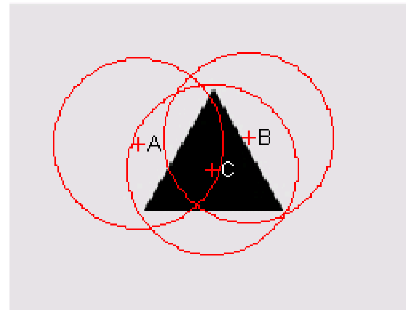


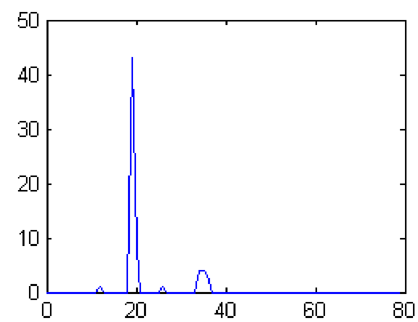
Figure 1 Orientation lines of the edge points

III. ALGORITHM

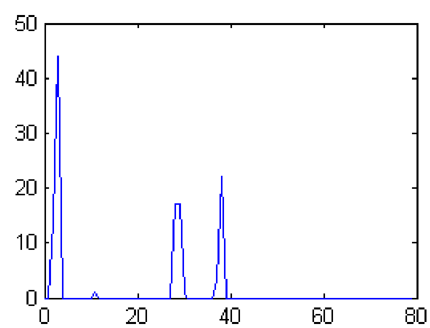
Based on the principle of the method, an algorithm is developed for detecting the arbitrary triangle. Three steps are interpreted in detail as the following.



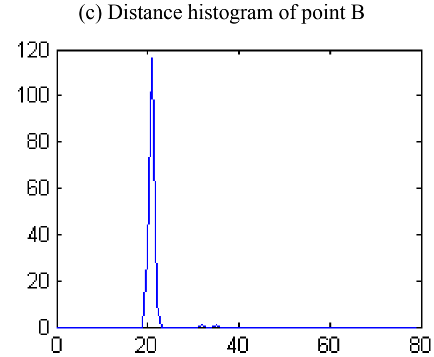
(a) Original triangle and the appointed points



(b) Distance histogram of point A



(c) Distance histogram of point B



(d) Distance histogram of point C

Figure 2 Distance distribution histograms of different points

A. Computing the Orientation Line of Edge Points

For an edge point $X_i(x_i, y_i)$ $(i = 1, 2, ..., N)$, assuming that N edge points are obtained in the image, the gradient of the point $X_i(x_i, y_i)$ is computed and denoted as $grad(X_i) = [d_{ix}, d_{iy}]$, in which the d_{ix} and d_{iy} indicates the gradient along the x-axis and y-axis respectively. The orientation line of point $X_i(x_i, y_i)$ is defined as the line which gets across the point $X_i(x_i, y_i)$ and is perpendicular with the gradient vector of the point, so the orientation line can be represented as $l_i : a_i x + b_i y + c_i = 0$, in which $a_i = d_{ix}$, $b_i = d_{iy}$ and $c_i = -d_{ix} x_i - d_{iy} y_i$. The orientation line of the edge point is coincident with the edge on which the point is located.

B. Computing the Feature Length and the Energy

Given an integer R, for a point $X(x, y)$, the pixel support region (PSR) is defined as a circular area, which is centered on X and of a radius of R. For each edge point $X_i (i = 1, 2, ..., N)$ in the PSR, assuming PSR consists of N edge points, the distance of the center pixel X to the orientation line of one edge point X_i is computed using the following equation (1):

$$d_i = {\left| a_i x + b_i y + c_i \right|}\Big/{\sqrt{a_i^2 + b_i^2}} \tag{1}$$

where a_i, b_i, c_i are the coefficients of the orientation line equation of edge point X_i and equal to d_{ix}, d_{iy} and $-d_{ix} x_i - d_{iy} y_i$ respectively. The distances of the center point X to the orientation lines of all the edge points in the PSR are computed and the distance distribution histogram of the point X is obtained. The maximum in the histogram is defined as the energy of point X denoted as $E(x, y)$ and the corresponding distance is defined as the feature length denoted as $K(x, y)$. For each pixel in the image, the feature length and the energy are computed, and the feature length distribution map and the energy distribution map of the whole image are obtained.

C. Local maxima detection

In order to determine the potential incenter of a triangle, local maxima detection is performed in the feature energy distribution map of the image. First, a threshold T is applied, which is determined with equation (2):

$$T = Mean(E) + k \cdot Std(E) \tag{2}$$

in which the $Mean(E)$ and $Std(E)$ respectively represents the mean and standard deviation of the feature energy distribution map, and the factor k in the equation is a coefficient in the range of 2~5 in a general way. For a point $X(x, y)$, it is considered as a candidate incenter when its energy $E(x, y)$ is larger than T and also larger than its eight neighbors in the distribution map.

For the determination of the true centers, the candidate centers are further validated. For a candidate center $P_1(x_1, y_1)$ with the energy $E(x_1, y_1)$ and feature radius $K(x_1, y_1)$, a smaller PSR denoted as sPSR of the radius $R' = T_{radius} \cdot K(x_1, y_1)$ compared to initial R is assigned. The radius R' is in proportion to the feature radius $K(x_1, y_1)$ and T_{radius} is related to the shape. The set of T_{radius} should make the sPSR include the detected shape, that is to say, the radius R' should at least equal to that of the circumcircle of the shape. Denote the radius of the inscribed circle and circumcircle of a shape as r and r' respectively. For a triangle, due to $2 \le r'/r < \infty$, the set of T_{radius} is equal to 2 when the triangle is equilateral and greater than 2 in other conditions.

Then, the distances between the orientation lines of the edge points in the sPSR of $P_1(x_1, y_1)$ and $P_1(x_1, y_1)$ are calculated by (3). The locations of the edge points whose corresponding distances are between $(K(x_1, y_1) - \Delta)$ and $(K(x_1, y_1) + \Delta)$ are added to set X_{ep}, which is initialized to an empty set. The number of the edge points in set X_{ep} is counted and denoted as N_{ep}. Assuming that there exists a geometrical shape with the incenter $P_1(x_1, y_1)$, the edge points constructing the edges of the shape can be added to set X_{ep}. So if N_{ep} fulfills the following condition

$$N_{ep}\Big/K(x_1, y_1) > T_{validate} \tag{3}$$

the candidate center $P_1(x_1, y_1)$ is added to a set C whose members are the true centers of the circles. Here, the threshold $T_{validate}$ is related to the geometrical shape. When the triangle is an equilateral triangle, the radius of its inscribed circle r has the maximum and the ratio of the perimeter of the triangle L to r will be $6\sqrt{3}$. Due to the discretization of the digital image, the number of the points constructing a line is equal to the length of the line only when the line is vertical or horizontal, and more generally, the former is less than the latter. So a factor of $\sqrt{3}\big/2$ is multiplied by L to approximate the number of the edge points on one side of the equilateral triangle, and the ratio of L to r, $T_{validate}$, can be determined approximately to be 9 in the case of the triangle.

D. Detection of the triangle

For detecting the triangle from other shapes, the gradient orientations of the edge points are used. When a potential incenter of a triangle is determined, the candidate edge points are also obtained. For a triangle, there should have three main orientations in the gradient orientation distribution of the edge points. This restriction is used to eliminate the non-triangle.

Furthermore, to improve the robustness of the proposed method to noise, an additional step is performed to cancel the points from the candidate edge points set which obviously diverge with other edge points. The detected triangle is represented with the remained edge points constructing the triangle.

IV. Experiment Results

A. Synthetic experiments

As shown in Fig. 3 (a), an image of size 404×113 with three different triangles is generated to perform the triangle detection. The energy distribution map of the synthetic image is displayed in Fig. 3 (b), in which the intensities of potential incenters of the triangles are larger obviously than other pixels. More than ten local maxima are obtained after the local maximum detection. Then (3) is used to validate the local maximum and the true incenters of the triangles are remained as Fig. 3 (c) marks. The detected inscribed circles of the triangles are also shown in Fig. 3(c). The final result is displayed in Fig. 3 (d).

The effects of noises and blurring on the method proposed are also evaluated. Fig. 4(a) gives a triangle image contaminated by the Salt & Pepper noise. The corresponding edge image is shown in Fig. 4 (b), in which many non-edge points are displayed. The result of the method is shown in Fig. 4 (c), where the triangle is correctly detected. The detection results under the effect of Gaussian noise are displayed in Fig. 4 (d) ~ (f). The effect of Gaussian blurring is also tested and the results are shown in Fig. 4 (g) ~ (i). From these results, it can be seen that the proposed method has good resisting for the effects of noises and blurring.

B. Real image experiment

Real image is also applied to test the performance of the method. An image of 150 × 108 pixels is used as Fig. 5 (a) shows. The detected results are shown in Fig. 5 (b), in which the triangle objects are detected and the edges points of triangles are shown with red points.

V. Conclusions

This paper has presented a method for automatically detecting the triangles based on the property of the inscribed circle of arbitrary triangle. For edge points of a triangle, the distance of the incenter to orientation line of the points are equidistant and equal to the radius of the inscribed circle, so the energy and feature length relative with the distance information are defined to find the incenter of a triangle, with the orientation distribution of obtained edge points, the triangles are detected finally. From our results, the proposed method has been capable of effectively detecting the triangle objects, and shows good accuracy and consistency despite the presence of noises and blurring.

Acknowledgment

This work is supported by National Natural Science Foundation of China (61005033), Open Projects Program of National Laboratory of Pattern Recognition (20090018) and Doctor Degree Foundation of Henan Polytechnic University (B2010-39, B2010-68).

References

[1] E. R. Davies, Machine vision: theory, algorithms, practicalities, 3rd ed., Elsevier, 2005, pp. 387-410.

[2] S. H. Chiu, J. J. Liaw, An effective voting method for circle detection. Pattern Recognition Letters, Vol 26, pp. 121-133, 2005.

[3] E. R. Davies, A modified Hough scheme for general circle location. Pattern Recognition Letters, Vol 7, pp. 37-43, 1988.

[4] R. K. K. Yip, P. K. S. Tam, D. N. K Leung, Modification of hough transform for circles and ellipses detection using a 2-dimensional array. Pattern Recognition, Vol. 25, pp. 1007-1022, 1992.

[5] A. Torii, A Imiya, The randomized-Hough -transform -based method for great-circle detection on sphere. Pattern Recognition Letters, Vol 28, pp. 1186-1192, 2007.

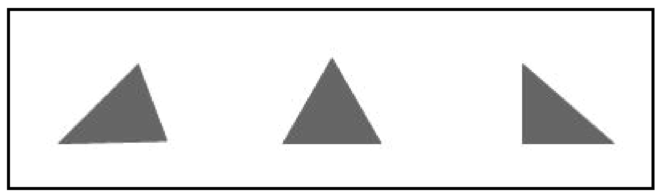
[6] N. Barnes, L. Gareth, S. David, The regular polygon detector. Pattern Recognition, Vol 43, pp. 592–602, 2010.

[7] M. Siddharth, W. P. David, Matching Flexible Polygons to Fields of Corners Extracted from Images. ICIAR 2007, Lecture Notes in Computer Science, 4633, pp. 447–459, 2007.

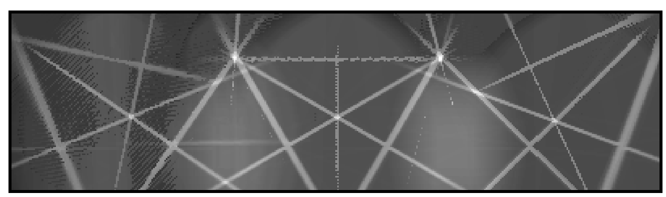
[8] J. Shi, Z. H. Xiao, Q. Chang, An Algorithm for Recognizing Geometrical Shapes Automatically Based on Tunable Filter. Journal of North University of China (Natural Science Edition), Vol 30, No. 5, pp. 467-471, 2009.

[9] A. Croitoru, Y. Doytsher, Right-angle rooftop polygon extraction in regularized urban areas: Cutting the corners. Photogrammetric Record, Vol 18, No. 108, pp. 311–341, 2004.

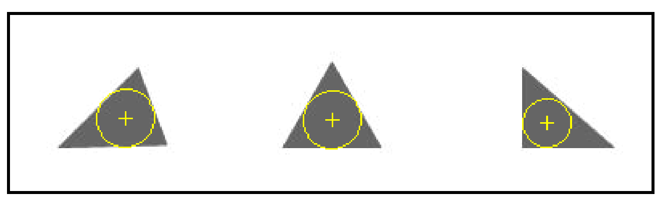
[10] J. W. Gates, M. Haseyama, H. Kitajima. Real-time polygon extraction from complex images. Circuits and Systems, 2000. Proceeding ISCAS 2000 Geneva. The 2000 IEEE Int'l Symp. on, Vol 5, pp. 309–312, 2000.



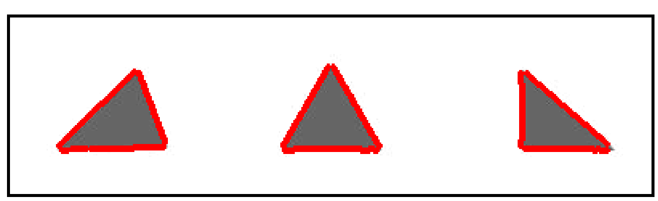
(a) Original image with different triangles



(b) Energy distribution map of the input image

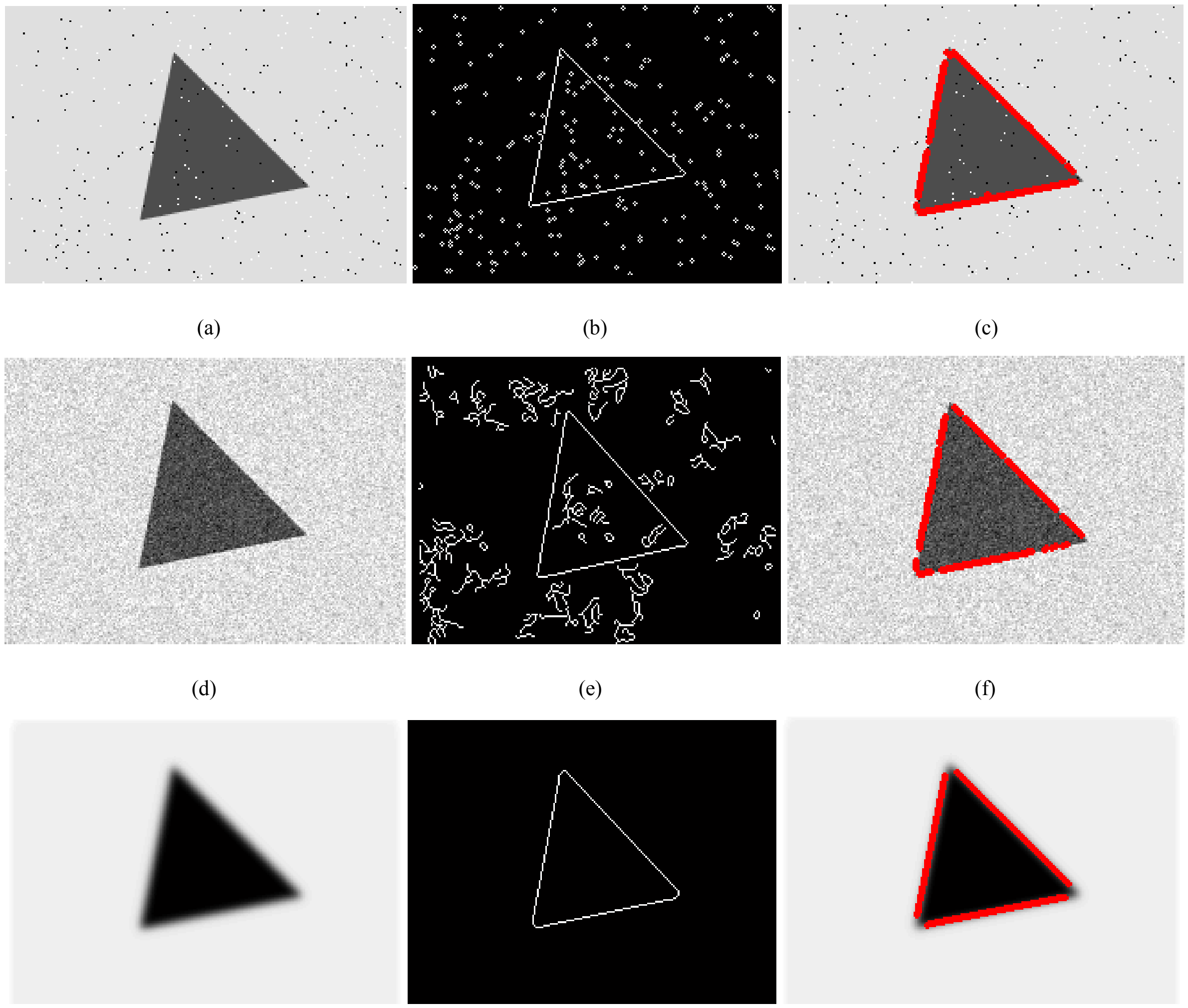


(c) The incenters and inscribed circles of the triangles



(d) Detected triangles

Figure 3. Triangles detection over synthetic image

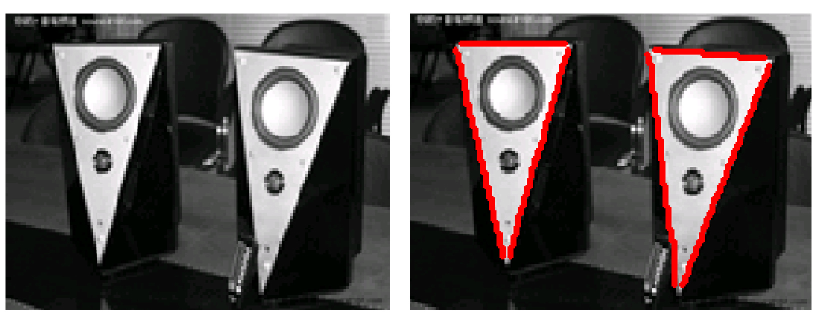


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4. The effects of noises and blurring on triangle detection. (a) is the triangle added Salt & Pepper noise, (b) and (c) is the edge detection and detection result of fig. a. (d) is the triangle added Gaussian noise, (e) and (f) is the edge detection and detection result of fig. d. (g) is the triangle added Gaussian blur, (h) and (i) is the edge detection and detection result of fig. g.



(a) Original image with triangle objects (b) Detection triangles

Figure 5 Triangle detection over real image