

A Fast Adaptive Corner Detection Based on Curvature Scale Space

Van Hau Nguyen[†], Kyung Haeng Woo^{**}, Won Ho Choi^{***}

ABSTRACT

Corners play an important role in describing object features for pattern recognition and identification. This paper proposed a fast and adaptive corner detector in both coarse and fine scale, followed by the framework of the curvature scale space (CSS). An adaptive curvature threshold and evaluating of angles of corner candidates are added to original CSS to remove round corners and false corners in the detecting process. The efficiency of proposed method is compared to other popular detectors in both accuracy criteria, stability and time consuming. Results illustrate that the proposed method performs extremely surpass in both areas.

Key words: CSS Corner Detection, Low Computational Time, Accuracy Criteria (ACC)

1. INTRODUCTION

Corner detection is an extremely key task in many applications based on image processing systems. Applications include motion tracking, object recognition, stereo matching [1-3]. Therefore, there have been many corner detection methods proposed up to now.

Many algorithms for image corner detection [4-8] have been developed. However, they are sensitive to noise, suffer from missing junctions [3], suffer from a loss in localization accuracy or much time consuming [4,8] and so on. Other corner detectors are stated in [9-12]. Most of them are single scale algorithms that work well if the object has

similar size features, but are ineffective for objects with multiple size features. Thus, single scale detection usually leads to either missing small corners or overlooking coarse features, which is unacceptable because objects, in general, cannot be assumed to have only features of a single size. To solve this problem, multi scale detection of corners is needed. Rattarangsi and Chin [7] proposed a multi scale algorithm based on Gaussian scale space. Although it can successfully detect multiple size features, the algorithm was computationally heavy due to the huge number of scales it requires. Mokhtarian and Suomela [6,13] proposed two CSS corner detectors for gray-level image, which were better than the existing corner detectors, because the CSS technique was more suitable for getting invariant geometric features of a planar curve at multiple-scales. However, in both algorithms, multi-scale is used only for localization while detection is still in single scale. Their first algorithm suffers from two problems: first it fails to detect true corners when deviation σ is large and detects a number of false corners when σ is small. Second, its performance is sensitive to a global threshold. Their second (enhanced) algorithm attempts to eliminate the

* Corresponding Author: Won Ho Choi, Address: (680-749) 93 Daehakro, Namgu, Ulsan, TEL: +82-52-259-1627, FAX: +82-52-259-1686, E-mail: whchoi@mail.ulsan.ac.kr

Receipt date: Jan. 31, 2011, Revision date: Mar. 17, 2011
Approval date: Apr. 13, 2011

[†] School of Electrical Engineering, University of Ulsan (E-mail: leo.phys@yahoo.com)

^{**} School of Electrical Engineering, University of Ulsan (E-mail: wkh2000@mail.ulsan.ac.kr)

^{***} School of Electrical Engineering, University of Ulsan

* This work was supported by 2009 Research Fund of University of Ulsan

above problems. By using different scales of the CSS for contours with different lengths before computing the absolute curvature and without using a global threshold, it offers a better set of detected corners.

In this paper, a fast and adaptive corner detection method based on the CSS corner detector is proposed. Different from the CSS methods in [6,13], curvature of each contour is first computed at a relatively low scale to retain all true corners. After determining the corner candidates by the local maxima of absolute curvature function, the curvature of corner candidates are compared with an adaptive local threshold instead of a single hard threshold to remove the rounded corners. Then the angles of candidate corners are checked to remove any false corners due to boundary noise and trivial details. The proposed detector has been evaluated over a number of images and compared with the two CSS methods as well as other popular corner detectors. It has also been evaluated using accuracy (ACU) criteria and computation time. It is found that it performs better than any of the existing corner detectors for objects with multiple size, and is more robust and reliable from image to image.

The following is the organization of the remainder of this paper. In Section 2, we briefly describe the original CSS and enhanced threshold CSS, and then analyze their limitations. Section 3 presents an overview of the proposed method. The performance of the speed and adaptive corner detection is evaluated in terms of the ACU and time requirement in section 4. Finally the conclusion is presented in section 5.

2. CSS CORNER DETECTOR

In this section, the original CSS and enhanced CSS [6,13] will be discussed. First we quote the definition of curvature K of the contour in CSS.

Let Γ represent a regular planar curve which is

parameterized by the arc length u . $\Gamma(u, \sigma)$ is an evolved version of the curve Γ

$$\Gamma(u) = (x(u), y(u)) \tag{1}$$

$$\Gamma(u, \sigma) = (X(u, \sigma), Y(u, \sigma)) \tag{2}$$

where $X(u, \sigma) = x(u) \oplus g(u, \sigma)$, $Y(u, \sigma) = y(u) \oplus g(u, \sigma)$ and \oplus is the convolution operator and $g(u, \sigma)$ denotes a Gaussian function with deviation σ .

The curvature of curve $\Gamma(u, \sigma)$ is defined as

$$K(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{(X_u^2(u, \sigma) + Y_u^2(u, \sigma))^{1.5}} \tag{3}$$

where

$$X_u(u, \sigma) = \frac{\partial}{\partial u}(x(u) \oplus g(u, \sigma)) = x(u) \oplus g_u(u, \sigma)$$

$$X_{uu}(u, \sigma) = \frac{\partial^2}{\partial u^2}(x(u) \oplus g(u, \sigma)) = x(u) \oplus g_{uu}(u, \sigma)$$

$$Y_u(u, \sigma) = y(u) \oplus g_u(u, \sigma), Y_{uu}(u, \sigma) = y(u) \oplus g_{uu}(u, \sigma)$$

and $g_u(u, \sigma)$, $g_{uu}(u, \sigma)$ denote the first and the second derivatives of $g(u, \sigma)$ respectively.

The following steps are used by the original CSS algorithm to detect corners an image:

1. Apply the Canny edge detector to the gray level image and obtain a binary edge-map.
2. Extract the edge contours from the edge-map, fill the gaps in the contours and find the T-junctions.
3. Compute curvature at a high scale, σ_{high} , for each edge contour.
4. Consider those local maxima as initial corners whose absolute curvature are above threshold and twice as much as one of the neighboring local minima.
5. Track the corners from the highest scale to the lowest scale to improve localization.
6. Compare the T-junction to other corners and remove one of the two corners which are very close.

3. PROPOSED METHOD

In the original CSS, a single scale is used in the detection procedure, and multi scale is used only for localization. As mentioned, it fails to detect true corners when σ is large and detects false corners when σ is small, where σ presents the scale. If this is applied to a complex image, the conflict between missing true corners versus detecting false corners become more severe. Another problem is its sensitivity to a global threshold value, t , which creates undesirable generalization of detection.

The enhanced CSS deals with these problems, by using different scales of the CSS for contours with different lengths, and smoothing the absolute curvature function for long contours to remove the false maxima. However, the criterion for selecting contour lengths is not explicit. Such criterion is obviously important as it determines the success of the algorithm. On the other hand, it is reasonable to believe that proper scale value does not sequentially depend on the contour length. The contour length is not a major attribute of a curve, since the algorithm for edge contour extraction can alter it. Actually, different size feature, which need different scale, can exist in the same contour. Although the enhanced CSS offers better results than the original CSS, there are rooms for improvement.

To solve the above problems, the proposed algorithm will be changed. The curvature is computed at a hard low scale for each contour to retain all true corners. Then, all of curvature local maxima are considered as corner candidates, including the false corner, by classifying the false corners into rounded and boundary noise [7]. The adaptive threshold and angle of corner are considered to remove them.

3.1 Adaptive Threshold to Remove Round Corner

Ideally, a corner is an intersection of two

straight lines. However, in reality, corners are often deformed with ambiguous shapes. Consider the ambiguous case as illustrated in Fig. 1, there are five points labeled on the curve, all of which represent maximal local curvature values and can be regarded as corner candidates. The region of support ROS of a corner is defined by the segment of the contour bounded by the two nearest curvature minima of corner. For corner candidate point 3, the ROS spans from point 2 to point 4.

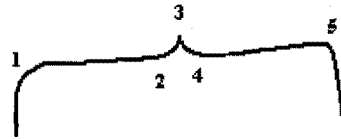


Fig. 1. Illustration of an ambiguous case.

Although the curvature of a round corner is the largest among its neighbors, the actual difference may not be significant. On the other hand, the curvature of an obtuse corner may have similar or even lower absolute maximum than a round corner. In order to utilize this curvature characteristic of the neighbors to eliminate round corners but the obtuse corners, the ROS of each corner is used to calculate a fine threshold adaptively

$$T(u) = R \times \bar{K} = R \times \frac{1}{L_1 + L_2 + 1} \sum_{i=u-L_2}^{u+L_1} |K(i)| \quad (4)$$

where u is the position of the corner candidate on the contour, L_1 and L_2 are two neighbor corner candidate points, $L_1 + L_2$ is the size of the ROS centered at u , and R is a coefficient, \bar{K} is the mean curvature of the ROS. In case of Fig. 1, for point 3, size of the ROS is the line from point 2 to point 4. If the curvature of the corner candidate is larger than $T(u)$, then it is declared a true corner; otherwise it is eliminated from the list. The round corners can be eliminated because for an obtuse corner, the curvature drops faster over $L_1 + L_2$ than that of a round corner over a similar ROS.

In theory, by controlling R appropriately we can differentiate round corners from obtuse corners

and eliminate various kinds of round corners as well. However, round corners are ill defined by nature, and there is no explicit criterion to distinguish them. For instance, using a circle to model, every point on a circle has the same curvature, so a circle has no obvious corner. However, for an ellipse, it could be argued that the vertices may be considered as true corners. Therefore, whether a round corner should be regarded as a true corner is determined by how round or sharp it is.

Suppose an ellipse is given by

$$f(x) = [b^2 - (\frac{bx}{a})^2]^{\frac{1}{2}} \tag{5}$$

with $x \in [-a, a]$ and $b > a$. The vertex $(0, b)$ of the ellipse will get a curvature maximum, so it is a true corner. The absolute curvature of every point which belongs to ellipse is calculated by substituting eq. (5) into eq. (3)

$$K(x) = \left| \frac{f'(x)}{[1 + f'(x)^2]^{\frac{3}{2}}} \right| = \frac{ba^4}{[(bx)^2 - (ax)^2 + a^4]^{\frac{3}{2}}} \tag{6}$$

and we have $K_{max} = b/a^2, K_{min} = a/b^2$.

The area under the curvature function given by

$$\int K(x)dx = \int \frac{ba^4}{[(bx)^2 - (ax)^2 + a^4]^{\frac{3}{2}}} dx = \frac{bx}{[(b^2 - a^2)x^2 + a^4]^{\frac{1}{2}}} \tag{7}$$

We have mean curvature given by

$$\bar{K} = \int_{-a}^a K(x)dx / 2a = 1/a = K_{max} \cdot a/b \tag{8}$$

From eq. (4), a adaptive threshold is given by

$$T(u) = R \times \bar{K} = R \cdot K_{max} \cdot a/b \tag{9}$$

$$K_{max} \begin{cases} < T \text{ if } b/a < R \\ > T \text{ if } b/a > R \end{cases}$$

In other words, if the ratio of its major axis to its minor axis is lower than R , it is a round corner. We use R to define round corners to be eliminated them out the corner candidates.

3.2 False Corner Removal

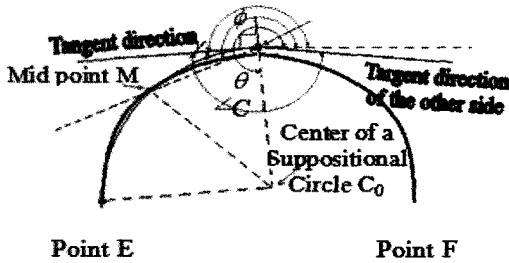
A well defined corner should have a relatively

sharp angle. If we know the angle of each point on a curve, it would be easy to differentiate true corners from false corners. The key success to this approach is to define angle of corner correctly. In Fig. 1, there are five corner candidates. If a small ROS is adopted, they all are true corners. If a larger ROS is considered, corners 2, 3, 4 may be regarded as false corners. When the feature size is not known a priori, it can be challenging to find the right corners.

This inspires us to use a dynamic ROS, which is determined by the property of the corner candidates themselves. To a corner candidate, its ROS should be defined by its two adjacent corner candidates. In Fig. 1, if all the five point labeled are corner candidates, then ROS of candidate 3 should span from points 2 to 4. It is then judged as a true corner according to its sharp angle. On the other hand, if only points 1, 3, 5 are retained as corner candidates after the adaptive local thresholding, the ROS for candidate 3 would span from points 1 to 5. Then it is likely regarded as a false corner because of the nearly straight line between 1 and 5.

We redefine the angle of a corner using tangents instead. For any point in the arc, the tangent directions on its two sides form an angle at that point. Similarly, a straight line can be regarded as an arc with infinite radius of curvature, so the tangent direction of any point on the line is the same as the line direction. In this respect, straight lines and arcs can be treated in exactly the same way. To calculate the tangent, a circle is best-fitted to the pixels on each arm of the ROS of the corner candidate, as shown in Fig. 2. A simple three-point method is employed to determine the circle. This three-point method is detailed below, with reference to Fig. 2.

First, on one branch of an ROS, for example from C to E , three points (C , mid-point M and E) are selected. If three points are collinear, the tangent direction of this branch is defined from C to



Point E Point F

Fig. 2. Tangent definition of corner.

E , else a center point C_0 is deduced. Let $C(x_1, y_1), M(x_2, y_2), C_0(x_0, x_0)$, we have

$$x_0 = \frac{(x_1^2 + y_1^2)(y_2 - y_3) + (x_2^2 + y_2^2)(y_3 - y_1) + (x_3^2 + y_3^2)(y_1 - y_2)}{2[x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)]}$$

$$y_0 = \frac{(x_1^2 + y_1^2)(x_2 - x_3) + (x_2^2 + y_2^2)(x_3 - x_1) + (x_3^2 + y_3^2)(x_1 - x_2)}{2[y_1(x_2 - x_3) + y_2(x_3 - x_1) + y_3(x_1 - x_2)]}$$

(10)

Second, a line is draw from C to C_0 , and θ is used to represent the direction from C to C_0 , which can be calculated by a four-quadrant inverse tangent function. Similarly, ϕ denotes for the direction from C to M . Then, the tangent of C at this side of ROS is given by

$$\gamma_1 = \theta + \text{sign}(\sin(\phi - \theta)) \cdot \frac{\pi}{2}$$

(11)

where *sign* is a signum function.

Third, the tangent of the ROS from C to F is determined similarly, and is denoted by γ_2 .

Fourth, the two tangent lines form the angle of the corner as following

$$\angle C = \begin{cases} |\gamma_1 - \gamma_2| & \text{if } |\gamma_1 - \gamma_2| < \pi \\ 2\pi - |\gamma_1 - \gamma_2| & \text{otherwise} \end{cases}$$

(12)

Finally, the corner checking criterion is given as follow

$$C_i = \begin{cases} \text{true corner} & \angle C_i \leq \theta_{obtnuse} \\ \text{false corner} & \angle C_i > \theta_{obtnuse} \end{cases}$$

The parameter $\theta_{obtnuse}$ denotes the maximum obtuse angle that a corner can have and still be considered as a true corner. False corners are marked and removed after all corner candidates have been checked.

As the set of corner candidates may change after this step, further iterations are required until it converges. Using this criterion, the isolate corner candidates due to boundary noise and trivial details can be removed, and the dominant features of multiple size can be retained.

4. EXPERIMENT RESULTS AND DISCUSSION

In this section, performance of detection results is compared with popular detectors on planar curves as well as on gray-level images. In order to evaluate its performance, we use accuracy (ACU) criteria [14] for measuring the localization accuracy of corner detectors. The final subsection discusses the computational time requirements.

4.1 Test Results on Planar Curves

To evaluate the performance of the proposed method on planar curves, we have chosen some verified test shapes from Chetverikov and Szabo [15] and compared with the two best performing detectors, Beus-Tiu and Chet-Sza from their comparative work on the following corner detectors: Chet-Sza [15], Rosen-John [9], Rosen-Wes [16], Freeman-David [17] and Beus-Tiu [12]. Then, we compared them with our proposed method. The test results are shown in Fig. 3. It is assumed that in Fig. 3(a)-(b), parameter values are tuned to get the best result of each shape. Therefore, in Fig. 3(c), the same parameter values are set for the proposed method to process the whole image sets of input.

Among all these detectors, Beus-Tiu tends to miss some of the obvious true corners, as can be seen shape of plane in Fig. 3(a), and is not able to suppress some of the round corners, as illuminated in fifth shape from the left of Fig. 3(a). On the other hand, Chet-Sza can suppress round corners effectively in simple shapes. However, in order to keep obtuse corners, it has to adjust parame-

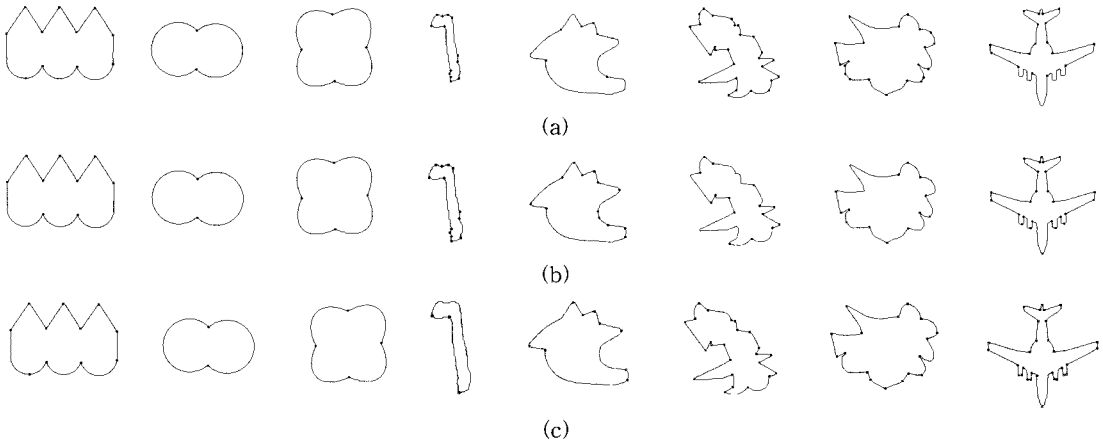


Fig. 3. Detecting results on planar images (a) Beus-Tiu (b) Chet-Sza (c) Our proposed method.

ters which can detect local variations (false corners) as corners. This can be obviously seen in the fourth shape from the left of Fig. 3(b). The proposed method has the best corner detection performance visually, in that it detects almost all the dominant features, including the fine features as in the shape of plane of Fig. 3(c) and suppresses noise and round corners. In summary, the proposed detector works quite well on different size features using the same R and $\theta_{obtuseness}$, the maximum obtuse angle that a corner can have and still kept as a true corner.

4.2 Test Results on Gray Images

For the purpose of evaluation, reference solutions for the testing images in are manually generated according to the ground-truth, as illustrated in Fig. 4(a) and Fig. 5(a) depicting the reference corners solutions of the "Blocks" and "House" image respectively.

Let assume that C_{ref} and C_{detect} denote the set of corners in the reference and set of corners detected by using proposed method. $d_{i,j}$ denotes distance between i^{th} corner in the reference and j^{th}

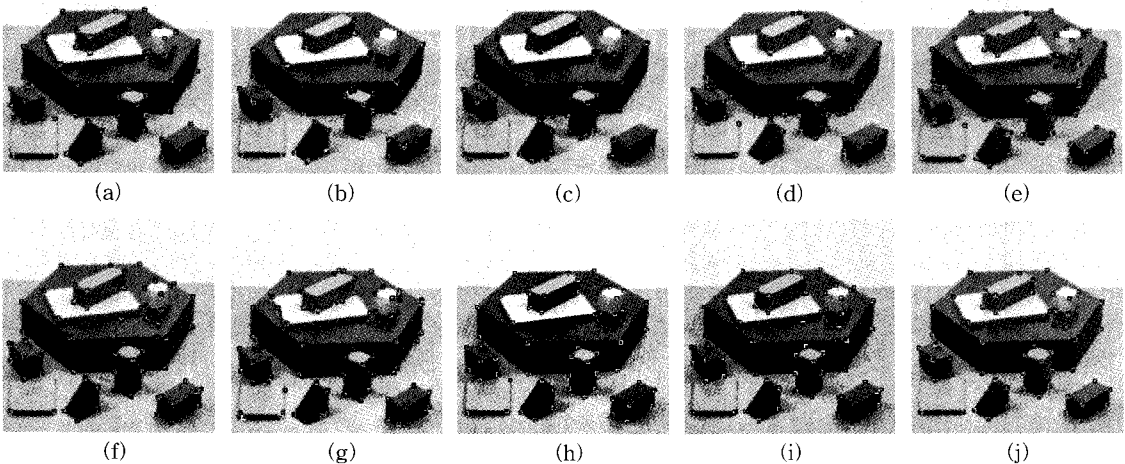


Fig. 4. Detecting results on the "Blocks" image (a) reference image (b) Plessey (c) Kitchen-Rosenfeld (d) SUSAN (e) original CSS (f) enhanced CSS (g) COP (h) CS (i) EV (j) our proposed method.

corner in the detected list. The true corner C_j is marked if $\min d_{i,j} < D_{max}$, otherwise C_i is marked as missed corner. The remaining corners in the detected corners are marked as false corners. Here, D_{max} is the maximum admissible distance, which is set to 4 pixels for our processing.

Following this evaluation method, the results of proposed method are compared to six other detectors that includes Plessey [4], Kitchen-Rosenfeld [5], SUSAN [8], original CSS [13], enhanced CSS [6], COP [18], CS [19] and EV [20]. The summarized results are shown in Table 1, Table 2 and Fig. 4, Fig. 5.

From Table 1, the number of real corners is 60 in the "Blocks" reference solution. The proposed method detected 56 true corners, which is about 93.3% while it has the least number of false corners and missed corners. The proposed method is also achieve the best rate of ratio of correct corners over total detected corners (90.3%). The other detectors performed much poorer in that their percentage of true corners ranges from 68.3% to 90% and Original CSS is 93.3%. However, their ratios of correct corners over total detected corners range from 66.7% to 85.7%. That should be inferred other methods detected many false corners.

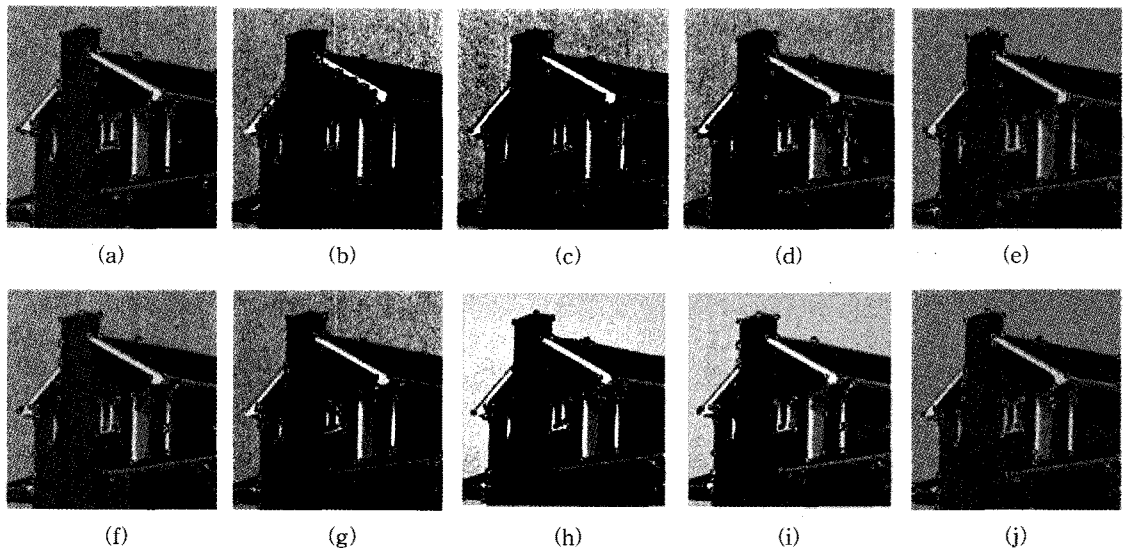


Fig. 5. Detecting results on the "House" image (a) reference image (b) Plessey (c) Kitchen-Rosenfeld (d) SUSAN (e) original CSS (f) enhanced CSS (g) COP (h) CS (i) EV (j) our proposed method.

Table 1. Evaluation results on the "Blocks" Image

Detector	True corners	Missed corners	False corners	Correct corners/ Total detected corners (%)	Correct corners/ Total referenced corners (%)
Plessey	41	19	17	70.7%	68.3%
Kitchen-Rosenfeld	48	12	14	77.4%	80.0%
SUSAN	44	16	19	69.8%	73.3%
Original CSS	56	4	14	80.0%	93.3%
Enhanced CSS	54	6	9	85.7%	90.0%
COP	52	8	26	66.7%	86.7%
CS	54	6	8	87.1%	90.0%
EV	52	8	10	83.9%	86.7%
Proposed method	56	4	6	90.3%	93.3%

Table 2. Evaluation results on the "House" Image

Detector	True corners	Missed corners	False corners	Correct corners/ Total detected corners (%)	Correct corners/ Total referenced corners (%)
Plessey	53	27	50	51.4%	66.3%
Kitchen-Rosenfeld	59	21	36	62.1%	73.8%
SUSAN	60	20	29	67.4%	75.0%
Original CSS	63	17	18	77.7%	78.9%
Enhanced CSS	49	31	12	80.3%	61.3%
COP	52	28	18	74.3%	65.0%
CS	53	27	8	86.7%	66.3%
EV	58	22	6	90.7%	72.5%
Proposed method	66	15	6	91.7%	82.5%

Similar results are shown in Table 2. The number of real corners is 80 in "House" image. However, because of the complexity, all the corner detectors had a lower success rate in detecting true corners. The best is 82.5% for the proposed method, followed by the original CSS at 78.75%. The ratios of true corners over detected corner are poorer than one of proposed method, which means many false corners were detected. It is also compared that this ratio is decreased when the complexity of image is increased. That means more and more false corners will be detected if test image is complex, which is similar to natural images. In summary, the proposed method remains the most successful in detecting true corners, with the least number of false corners and stability to complexity of image.

4.3 Accuracy

Accuracy means corners should be detected as close as possible to the correct position. In one image, the corner positions and the numbers can be different according to different people. Also as there is no standard procedure to measure accuracy of corners detectors. Therefore, we adopted a new approach for the creating ground truth to evaluate the accuracy of corner detectors as in [5].

Let N_0 be the number of the corners in the original image (note that $N_0 \neq 0$), N_a the number of the matched corners in the original image when com-

pared to the ground-truth corners and N_g the number of corners in the ground-truth. The criterion of accuracy is

$$ACU = 100 \times \frac{\frac{N_a}{N_0} + \frac{N_a}{N_g}}{2} \tag{13}$$

where ACU stands for accuracy. The value of ACU should be close to 100%. In the following, the comparative experiments are carried out among Plessey, Kitchen-Rosenfeld, SUSAN, OCSS, ACSS in terms of ACU criteria.

The comparison of accuracy for tested detectors have been illustrated in Fig. 6. Over all, the results of these comparisons show that our corner detector has the better accuracy and the stability among these corner detectors.

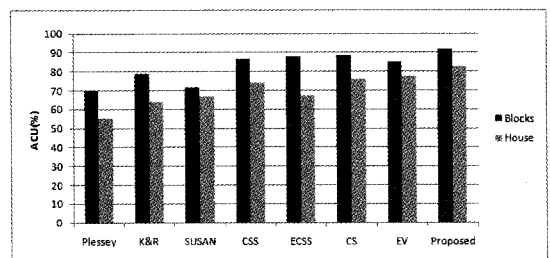


Fig. 6. Comparison of accuracy for tested detectors.

4.4 Computation Time

In order to compare the running time of these detectors more reasonably, the experiments were carried out ten times. Then, we calculate their

Table 3. Efficiency evaluation results

Detectors	Plessy	K&S	SUSAN	CSS	ECSS	CS	EV	Proposed
Corners/sec	297	772	1132	1025	907	1011	1108	1145

average time. Because the numbers of corners detected are different, we use the ratio of the number of detected corner to time that they spent to evaluate their efficiency.

Table 3 shows the efficiency of proposed clearly higher than others. It is note that although method CS and EV have detection rates are lower than those of CSS or ECSS, computation time is better. However, our proposed method achieves better than detection rate and speed. This enables the proposed method to be deployed in real time application.

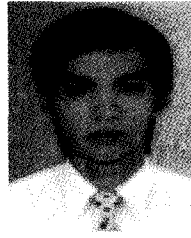
5. CONCLUSION

In this paper, the coarse and fine curvature of corners are considered in the detection, which the use of adaptive threshold and dynamic region to find corners makes them become clearly. The proposed method performs the comparative results such as increasing the number of true corners, decreasing the missed and false corners... in comparison to other popular methods. The experiments also demonstrate the good quality and robustness but only low computational cost. All these advantages make the proposed scheme potential to apply to real time systems.

REFERENCE

- [1] B. Serra and M. Berthod, "3-D Model Localization Using High Resolution Reconstruction of Monocular Image Sequences," *IEEE Trans. Image Proces.*, pp.175-188, 1997.
- [2] G. S. K. Fung, N. H. C. Yung, and G. K. H. Pang, "Vehicle Shape Approximation from Motion for Visual Traffic Surveillance," in *Proc. IEEE 4th Int. Conf. on Intelligent Transportation Systems*, pp.201-206, 2001.
- [3] G. S. Manku, P. Jain, A. Aggarwal, A. Kumar, and L. Banerjee, "Object Tracking Using Affine Structure for Point Correspondence," in *Proc. 1997 IEEE Comput. Soc. Conf. on Computer Vision and Patter Recognition*, pp. 704-709, 1997.
- [4] Harris C. and Stephens M., "A Combined Corner and Edge Detector," in *Proc. of Fourth Alvey Vision Conf.*, pp.147-151, 1988.
- [5] L. Kitchen and A. Rosenfeld, "Gray Level Corner Detection," *Pattern Recogn. Lett.*, pp. 95-102, 1982.
- [6] Mokhtarian, F. and Suomela, R., "Enhancing the Curvature Scale Space Corner Detector," In: *Proc. Scandinavian Conf. on Image Analysis*, pp.145-152, 2001.
- [7] Rattarangsi, A. and Chin, R.T., "Scale-Based Detection of Corners of Planar Curves," *IEEE Trans. Pattern Anal. Mach. Intell.* 14, pp.430-449, 1992.
- [8] Smith S. and Brady J., "SUSAN-A New Approach to Low-Level Image Processing," *Int. J. Comput. Vis.* 23, pp.45-48, 1997.
- [9] A. Rosenfeld and E. Johnston, "Angle Detection on Digital Curves," *IEEE Trans. Comput.* 22, pp.875-878, 1973.
- [10] 신충호, 오무송, "The Optimal Skeleton Method of an Image," *Journal of Korea Multimedia Society*, Vol.6, No.2, pp. 224-229, 2003.
- [11] H. C. Liu and M. D. Srinath, "Corner Detection from Chain-Code," *Pattern Recogn.* 23, pp.51-68, 1990.
- [12] H. L. Beus and S. S. H. Tiu, "An Improved Corner Detection Algorithm Based on Chain-Coded Plane Curves," *Pattern Recogn.* 20, pp. 291-296, 1987.

- [13] Mokhtarian F. and Suomela R., "Robust Image Corner Detection Through Curvature Scale Space," *IEEE Trans. Pattern Anal. Mach. Intell.* 20, pp. 1376-1381, 1998.
- [14] Mohanna Farahnaz and Mokhtarian Farzin, "Performance Evaluation of Corner Detection Algorithms Under Similarity and Affine Transforms," *BMVC*, pp. 353-362, 2005.
- [15] D. Chetverikov and Z. Szabo, Detection of High Curvature Points in Planar Curves, <http://visual.ipan.sztaki.hu/corner/index.html>, 1999.
- [16] A. Rosenfeld and J. S. Weszka, "An Improved Method of Angle Detection on Digital Curves," *IEEE Trans. Comput.*, pp. 940-941, 1975.
- [17] H. Freeman and L. S. Davis, "A Corner Finding Algorithm for Chain Coded Curves," *IEEE Trans. Comput.*, pp. 297-303, 1977.
- [18] S. C. Bae, I. S. Kweon, and C. D. Yoo, "COP: A New Corner Detector," *Pattern Recogn. Lett.*, pp. 1349-1360, 2002.
- [19] Ming Li, Jia-Zhuo Wang, Ling-Ling Li, and Cui-Hua Li, "Curvature Statistic Corner Detection," 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), Vol.1, pp. 309-312, 2009.
- [20] Byung-Seung Jeon, Dong-Gi Woo, Young-Hak Mo, and Myo-Taeg Lim, "An Improved Corner Point Detection Using Extreme Value of Susan Method for Measuring a Displacement," *ICCAS-SICE*, pp. 5392-5396, 2009.



Van Hau Nguyen

received the B.S. degree in electronic physics from Vietnam National University- Hochiminh City, University of Natural Sciences in 2006. He is currently working towards M.S degree on electrical engineering. His research interests include the areas of image processing and digital signal processing.



Kyung Haeng Woo

received the B.S., M.S. degree in electronic engineering from University of Ulsan in 1993 and 1995, respectively. From 1998 to 2000 she worked at Samchang co. Ltd., as a R&D member. From 2005 to 2010 she worked at E&I tech as a president. From 2010 until the present she has been with the University of Ulsan, and now she is a guest professor at the School of Electrical Engineering. Her research interests include the image processing and computer vision.



Won Ho Choi

received the B.S., M.S., and Ph. D. degree in electronic engineering from Yonsei University in 1978, 1980, and 1990, respectively. From 1980 to 1985 he worked at Jeil Precision and Hewlett-Packard as a R&D member. From 1986 until the present he has been with the University of Ulsan, and now he is a professor at the School of Electrical Engineering. His research interests include the image processing algorithm/hardware design.