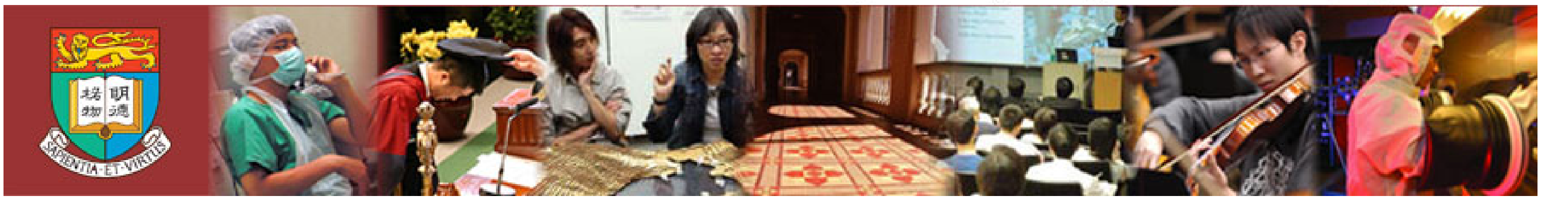


# Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support

*X.C. He and N.H.C. Yung*
Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong
Email: { xche, nyung } @eee.hku.hk

## Abstract

*Corners play an important role in object identification methods used in machine vision and image processing systems. Single-scale feature detection finds it hard to detect both fine and coarse features at the same time. On the other hand, multi-scale feature detection is inherently able to solve this problem. This paper proposes an improved multi-scale corner detector with dynamic region of support, which is based on Curvature Scale Space (CSS) technique. The proposed detector first uses an adaptive local curvature threshold instead of a single global threshold as in the original and enhanced CSS methods. Second, the angles of corner candidates are checked in a dynamic region of support for eliminating falsely detected corners. The proposed method has been evaluated over a number of images and compared with some popular corner detectors. The results showed that the proposed method offers a robust and effective solution to images containing widely different size features.*

## 1. Introduction

Corners in images represent a lot of useful information and they play an important role in describing object features for recognition and identification. Applications that rely on corners include motion tracking, object recognition, stereo matching, among many others. For this reason, a number of corner detection methods have been proposed [1-3]. Most of them are single-scaled and work well if the object has similar size features, but are ineffective for objects with multiple-size features. As such, either the fine or coarse feature is missing, which is unacceptable because objects, in general, cannot be assumed to have only features of a single size. To alleviate this problem, Rattarangsi and Chin [4] proposed a multi-scale algorithm based on Gaussian scale space, which can detect corners of planar curves. Although it can detect multiple-size features, the algorithm is computationally intensive due to the huge number of scales it requires. On the other hand, the curvature scale space (CSS) technique is more suitable for recovering invariant geometric features of a planar curve at multiple scales. Mokhtarian and Suomela [5,6] proposed two CSS corner detectors for gray-level image. 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 $\sigma$ is large and detects a number of false corners when $\sigma$ is small. Second, its performance is sensitive to a global threshold. Their second (enhanced) algorithm attempts to eliminate the 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, we propose a new and improved corner detection method based on the CSS corner detector. Different from the CSS methods in [5, 6], 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 global 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 above checking is based on dynamic region of support, which changes according to features' local size. The proposed detector has been evaluated over a number of images with multiple-size and compared with the two CSS methods as well as other popular corner detectors. 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. In Section 2, we give an overview of the original and enhanced CSS methods and describe their merits and limitations. In Section 3, our proposed corner detection method is described in detail. Section 4 describes the experiment result and provides an analytical discussion of the results.

## 2. CSS corner detectors

In this section, we discuss the original and enhanced CSS corner detectors. To begin the discussion, we quote the definition of curvature, $\kappa$, from [6] as:

$$\mathcal{K}(u,\sigma)=\frac{\dot{X}(u,\sigma)\ddot{Y}(u,\sigma)-\ddot{X}(u,\sigma)\dot{Y}(u,\sigma)}{(\dot{X}(u,\sigma)^2-\dot{Y}(u,\sigma)^2)^{1.5}},$$

where $\dot{X}(u,\sigma)=x(u)\otimes\dot{g}(u,\sigma)$, $\ddot{X}(u,\sigma)=x(u)\otimes\ddot{g}(u,\sigma)$, $\dot{Y}(u,\sigma)=y(u)\otimes\dot{g}(u,\sigma)$, $\ddot{Y}(u,\sigma)=y(u)\otimes\ddot{g}(u,\sigma)$, and $\otimes$ is the convolution operator while $g(u,\sigma)$ denotes a Gaussian function with deviation $\sigma$, and $\dot{g}(u,\sigma)$, $\ddot{g}(u,\sigma)$ are the 1st and 2nd derivatives of $g(u,\sigma)$ respectively.

The following steps are used by the original CSS algorithm [5] to detect corners of 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, $\sigma_{high}$, for each edge contour.
4. Consider those local maxima as initial corners whose absolute curvature are above threshold *t* 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.

In this algorithm, 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 algorithm [6] 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 consequentially 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.

## 3. Proposed method

To address the above issues, the proposed algorithm differs from the original and enhanced CSS in Steps 3&4:

3. Compute curvature at a fixed low scale for each contour to retain all true corners.
4. All of the curvature local maxima are considered as corner candidates, including the false corners. By classifying the false corners into rounded and due to boundary noise and details [4], two criteria (as described in 3.1 and 3.2) are used to remove them.

### 3.1 Adaptive local threshold

Among the corner candidates, although some points are detected numerically as the local absolute maximum, the measurable differences in curvature between this maximum and its neighbors in the region of support are often very small. This is the case of rounded corners, and fortunately we can remove them by using an adaptive local curvature threshold. In principle, we set the threshold for a candidate according to its neighborhood region's curvature. The local maxima whose absolute curvatures are under its local threshold are eliminated. This adaptive threshold is given by:

$$T(u)=C\times\overline{\mathcal{K}}=1.5\times\frac{1}{L1+L2+1}\sum_{i=u-L2}^{u+L1}\mathcal{K}(i)$$

where the mean value, $\overline{\mathcal{K}}$, is used to represent the curvature of a neighborhood region. In this case, a region of support (ROS) is defined as from one of the neighboring local curvature minima to the next, where the curvature is strictly decreasing from the candidate point to both ends. In the equation above, *u* is the position of corner candidate in the curve, *L1* and *L2* are sizes of the ROS, and *C* is a coefficient. It should be noted that if *C* is set to 1, no corner is removed, and for the purpose of retaining a corner whose curvature function waveform is a standard triangle, the boundary value of *C* is 2. By observation, the round corner has a convex waveform in absolute curvature function, and is not sharper than a triangle. So, theoretically *C* should be greater than 1 and less than 2. A median value of 1.5 is chosen for the proposed method, and it works well for almost all images. Through various testing, we found that this value is robust and changing of it does not affect the corner detection performance significantly.

### 3.2 Angle of corner

In general, a well-defined corner should have a relatively sharp angle. As argued in [7], if we know the angle of each point on a curve, it would be easy to differentiate true corners from false corners. The key to this approach is to use a proper ROS, i.e. a proper scale. 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. 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.

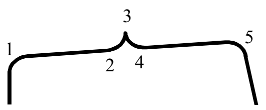


**Figure 1. Illustration of an ambiguous case**

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 candidate 3's ROS 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. And it is likely regarded as a false corner because of the nearly straight line between 1&5. Therefore, this corner checking criterion is given by:

If $160° \le \angle C_i \le 200°$, then $C_i$ is a **False Corner**

Else $C_i$ is a **True Corner**.

$\angle C_i$ is given by $\angle C_i = \left|\tan^{-1}(\Delta Y1/\Delta X1) - \tan^{-1}(\Delta Y2/\Delta X2)\right|$, where

$$\Delta X1 = \frac{1}{L1}\sum_{i=u+1}^{u+L1} X(i) - X(u);\ \Delta Y1 = \frac{1}{L1}\sum_{i=u+1}^{u+L1} Y(i) - Y(u);$$

$$\Delta X2 = \frac{1}{L2}\sum_{i=u-L2}^{u-1} X(i) - X(u);\ \Delta Y2 = \frac{1}{L2}\sum_{i=u-L2}^{u-1} Y(i) - Y(u).$$

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. Experimental results and analysis

In this section, the experimental results of the proposed corner detector are presented. We attempted many different images, but only two are being depicted in this paper: Blocks image in Fig.2 and House image in Fig.3. The results of the proposed method and five other corner detectors (Plessey [1], Kitchen and Rosenfeld [2], SUSAN [3], original CSS [5], and enhanced CSS [6]) are shown. For the purpose of evaluation, a reference solution for each image is manually generated, where corners are identified in appropriately magnified version of the image. Since it is often difficult to decide whether or not a point should be classed as a corner, only entirely obvious corners are included in the reference solutions.

The method of performance evaluation adopted in this research is described below. Let $C_{REF}$, $C_{DET}$ denote the set of corners from the reference solution and the set of corners found by a particular detector, respectively. Let $D_{max}$ be the maximal admissible distance between the detected and reference corner locations for which the detection is considered to be correct (we set $D_{max}$=4 pixels). For corner points $C_i \in C_{REF}$, $C_j \in C_{DET}$, if the distance $d_{i,j}$ between $C_i$ and $C_j$ is minimum for $\forall$i, j, and if $d_{i,j} \le D_{max}$, then $C_j$ labeled as a 'correct' detection of $C_i$, otherwise $C_i$ is labeled as 'missed' corners. The corners labeled as 'missed' in $C_{REF}$ are considered as true corners not detected and the remaining corners in $C_{DET}$ are considered to be the 'false' detections. The localization error is the average of all the distances $d_{i,j}$ for the corners detected correctly.

The results are summarized in Tables 1, 2 and Fig. 2, 3. It can be seen from Table 1 that the number of true corners is 60. The proposed method detected the most number of true corners, has the least number of missed and false corners, although the localization error is only the second best. The CSS methods are reasonable here, with the CSS performs slightly better on correct and missed corners, but with a lot more false corners. On the other hand, CSS has a large error, while the enhanced CSS has the best error. The other corner detectors performed substantially poorer than the three CSS-based detectors. Similar results are shown in Table 2, except that CSS now is the best in detecting correct and missed corners, but has a lot of false corners and high error, while the proposed method is best in error, least number of false corners and slightly worse than CSS in detecting correct and missed corners.

Subjective observation of Fig. 2 and 3 shows that the proposed method indeed resulted in corners that closely resemble the reference corner list and has very little false corners detected. The main contribution of this paper is in the use of adaptive local threshold and dynamic ROS to identify corners. Different parameters are automatically set for not only different images, different curves, but also different corner candidates. As a result, the proposed method increases the number of true corners detected and reduces the number of false corners detected. We can conclude that the proposed method is more efficient and accurate than the two CSS methods.

**Table 1. Evaluation results of the Blocks image**

| Detector | Correct corners | Missed corners | False corners | Localization error |
|---|---|---|---|---|
| Plessey | 41 | 19 | 17 | 1.6487 |
| Kitchen/Rosenfeld | 48 | 12 | 14 | 1.5389 |
| SUSAN | 44 | 16 | 19 | 1.5992 |
| CSS | 56 | 4 | 14 | 1.8542 |
| Enhanced CSS | 54 | 6 | 9 | 0.9941 |
| **Proposed** | **57** | **3** | **4** | **1.3902** |

**Table 2. Evaluation results of the House image**

| Detector | Correct corners | Missed corners | False corners | Localization error |
|---|---|---|---|---|
| Plessey | 55 | 19 | 48 | 1.6015 |
| Kitchen/Rosenfeld | 61 | 13 | 34 | 1.6131 |
| SUSAN | 61 | 13 | 28 | 1.7506 |
| CSS | 63 | 11 | 18 | 1.5728 |
| Enhanced CSS | 48 | 26 | 13 | 1.3048 |
| **Proposed** | **62** | **12** | **4** | **1.0085** |

## Reference

[1] C. Harris. "Determination of ego-motion from matched points". In *Proc. Alvey Vision Conf.*, Cambridge, UK, 1987

[2] L. Kitchen and A. Rosenfeld. "Gray level corner detection". *Pattern Recognition Letters*, pp. 95-102, 1982

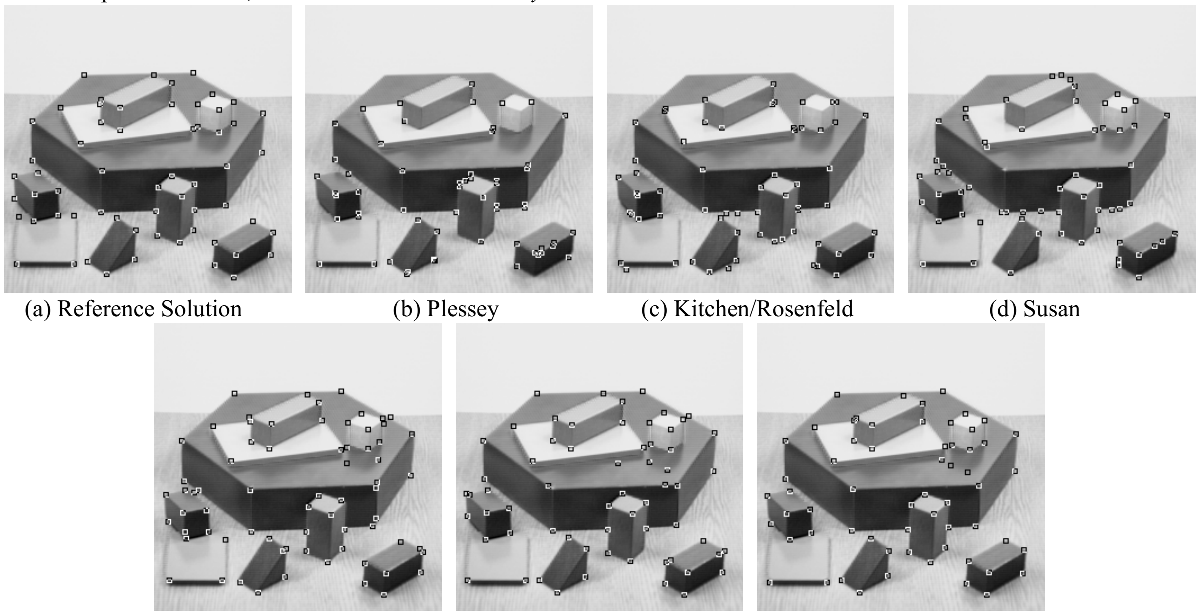
[3] S. Smith and J. Brady. "SUSAN—A new approach to low-level image processing". *International Journal of Computer Vision*, 23(1):45-48, 1997.

[4] A. Rattarangsi and R. T. Chin, "Scale-based detection of corners of planar Curves", *IEEE Trans on Pattern Analysis and Machine Intelligence,* 14(4): 430-449. 1992

[5] F. Mokhtarian and R. Suomela. "Robust image corner detection through curvature scale space". *IEEE Trans on Pattern Analysis and Machine Intelligence,* 20(12): 1376-1381, 1998

[6] F. Mokhtarian, and F. Mohanna, "Enhancing the curvature scale space corner detector", *Proc. Scandinavian Conf. on Image Analysis,* pp. 145-152, Bergen, Norway, 2001

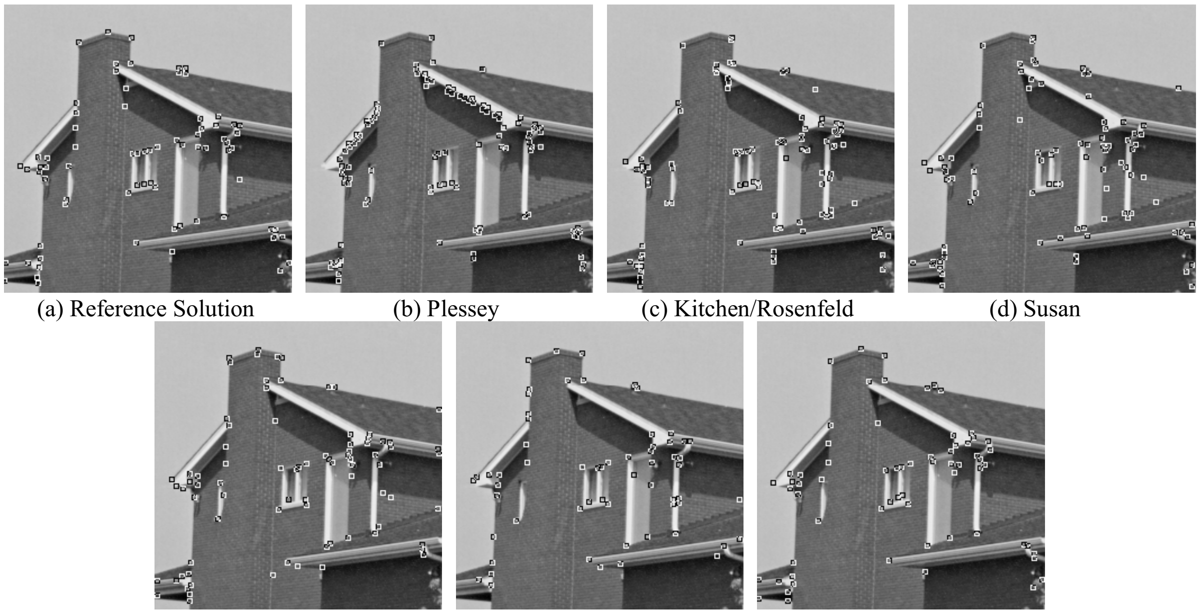
[7] A. Rosenfeld, and E. Johnston, "Angle detection on digital curves", *IEEE Trans. Computer.*, 22: 875-878 1973.



(a) Reference Solution (b) Plessey (c) Kitchen/Rosenfeld (d) Susan

(e) Original CSS (f) Enhanced CSS (g) Proposed Method

**Figure 2. Blocks image**



(a) Reference Solution (b) Plessey (c) Kitchen/Rosenfeld (d) Susan

(e) Original CSS (f) Enhanced CSS (g) Proposed Method

**Figure 3. House image**