

# *K*-Cosine Corner Detection

Te-Hsiu Sun

Department of Industrial Engineering and Management  
Chaoyang University of Technology, Taichung County, Taiwan.  
e-mail: thsun@mail.cyut.edu.tw

**Abstract**—This study presents a boundary-based corner detection method that achieves robust detection for digital objects containing wide angles and various curves using curvature. The boundary of an object is first represented into curvature measured by *K*-cosine. Then, by modifying the corner detection error, this study proposes a suitable *K* value and curvature threshold for robust corner detection. Furthermore, the proposed *K*-cosine corner detection (KCD) was verified with several commonly employed digital objects. The experimental results reveal that the proposed method is free from translation, rotation and scaling, and is superior to Tsai's method [34] in computation speed in discriminating false targets. A simple case study is shown finally to demonstrate the feasibility and applicability for practical use of KCD.

**Index Terms**—*K*-cosine, corner detection, curvature, image processing

## I. INTRODUCTION

Corner detection plays a critical role in image processing and pattern recognition in which many different approaches having been developed. These approaches can be broadly classified into two major categories: gray-level and boundary-based approaches. Gray-level approaches match corners using gray-level corner templates [17][14] or compute the gradient at edge points [28][25] while boundary-based approaches analyze the properties of boundary pixels to identify corners [12][34]. Representing boundaries using different descriptors and then searching for the corner features is common in boundary-based approaches. Various boundary descriptors, such as chain code, curvature, and Fourier Descriptor (FD), have been developed. The Freeman chain code is easy to use and can be treated as a polygonal approximation of a contour, although it is less efficient and accurate [6]. Fourier descriptors have been successfully applied to contour enhancement and object inspection, providing position, scale and orientation invariant properties by normalization [3][26]. However, they require heavy computation when calculating the complex equations of the forward and backward transformations.

Curvature, defined as the change rate of the slope, has

been widely employed in different applications such as shape representation, feature extraction, corner detection and object recognition [3][4][9][15][17][21][32]. Different numeric curvature estimation approaches have been discussed. Rosenfeld and Johnston initially defined curvature as a *K*-cosine function, where *K* denotes a region of support on the boundary [23]. Mokharian and Mackworth [19], Teh and Chin [31], and Sohn et al. [29] expressed curvature with a formula involving its first- and second-order directional derivatives. Liu and Srinath evaluated the curvature by convolving the edge direction function with the first derivative of a Gaussian function at each pixel [16]. Fairney et al. experimented with several different measures of digital curvature and found them to be unreliable in the presence of noise [5]. Tsai and Chen computed directly the curvature by measuring the first- and second-order derivatives of the continuous functions[33]. Tsai measured the curvature by using neural networks to identify the included angles at boundary points [32]. Later, Tsai et al. employed the eigenvalue of covariance matrices to measure the curvature and detect the sharp corners in a contour [34]. Arrebola et al. evaluated the curvature as the correlation factor coefficient of the forward and backward histograms in a *K*-vicinity of a given point [1]. For simplicity, this study adopts the *K*-cosine to measure the magnitude of curvature.

Many algorithms have been developed to locate corners using the local curvature maximum [2][7][8][19]. Rattaransi and Chin found corners by transforming the coded digital scale space map into an organized tree by calculating the maximum of absolute curvature [22]. Sohn et al. applied a constrained regularization method to derive an optimal smoothing factor for curvature estimation, which facilitates the corner detection [29]. Sohn et al. later developed the mean field annealing strategy, which uses simulated annealing to improve the approximation of the curvature estimation for corner detection [30]. Lee et al. developed a multi-scale corner detection algorithm using wavelet transform of contour orientation, utilizing both the local extrema and the modulus of transform results to identify corners and arcs [13]. Sheu and Hu developed a two-phase corner detection scheme [27]. The first phase specifies the points with significant curvature as candidate corners, and then the second phase verifies them using a self-adjusting convolution window. Inesta et al. employed curvature to identify dominant points for polygonal approximations of real imaged objects, and defined

T. H. Sun is currently an associate professor in Department of Industrial Engineering and Management at Chaoyang University of Technology, Taiwan. (e-mail: thsun@mail.cyut.edu.tw).

$K$ -curvature,  $K$ -angular bending and  $K$ -cosine [10]. Mokhtarian and Suomela first extracted edges using a Canny edge detector, and then detected corners based on the maximum of absolute curvature in the curvature scale-space [20].

However, determining a suitable  $K$  value is difficult in practice. A small  $K$  makes the curvature noisy owing to digitization, while a large value loses the relevant information of points within the  $K$ -pixel area. Moreover, determining the  $K$  value and the corresponding curvature threshold for detecting corners are critical, but they have not been discussed in literature. This study presents a boundary-based corner detection method using  $K$ -cosine, and proposes a suitable  $K$  and curvature threshold. This method also aims to attain robust corner detection for objects containing wide-angles and various curves. The remainder of this study is organized as follows. Section 2 describes the proposed  $K$ -cosine Corner Detection method. Section 3 presents the experiments. Application and conclusions are finally presented in Sections 4 and 5.

## II. K-COSINE CORNER DETECTION (KCD)

Let a sequence of  $m$  digital points describe the boundary of an object,

$$S = \{ P_j = (x_j, y_j) | j = 1, 2, 3, \dots, m \}; \quad (1)$$

where  $P_{j+1}$  denotes a neighbor of  $P_j$ , and  $(x_j, y_j)$  represents the Cartesian coordinate of the boundary point  $P_j$ . If  $S$  can be divided into  $n$  subsets, then it can also be described as

$$S_i = \{ (x_{ij}, y_{ij}) | j = 1, 2, 3, \dots, m \}, i = 1, 2, 3, \dots, n; \quad (2)$$

and

$$S = S_1 \cup S_2 \cup S_3 \dots \cup S_n = \bigcup_{i=1}^n S_i. \quad (3)$$

To represent the boundary  $S$  as a curvature,  $K$ -cosine is employed to measure the magnitude of curvature over a given region of support ( $K$ ) in the boundary.

### A. Definitions of $K$ -cosine Curvature and Corner Detection Error

The  $K$ -cosine is defined as below.

<Definition>  $K$ -cosine

Given an object whose boundary is defined by  $S = \{ P_i | i = 1, 2, 3, \dots, m \}$ , the curvature ( $K$ -cosine) of each boundary point  $P_i$  is defined as

$$c_i(K) = \cos \theta_i = \frac{\vec{a}_i(K) \cdot \vec{b}_i(K)}{\|\vec{a}_i(K)\| \cdot \|\vec{b}_i(K)\|}, \quad (4)$$

As depicted in Fig. 1,  $\vec{a}_i(K) = \vec{P}_{i+K} - \vec{P}_i$ ,  $\vec{b}_i(K) = \vec{P}_{i-K} - \vec{P}_i$ ,  $\theta$  denotes the angle between  $\vec{a}_i(K)$  and  $\vec{b}_i(K)$ , and  $K \in \mathbb{N}$ .

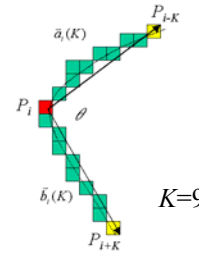


Fig 1. Curvature measurement with  $K$ -cosine

In Eq. (4),  $c_i(K)$  denotes the cosine of the angle between vectors  $\vec{a}_i(K)$  and  $\vec{b}_i(K)$  for a given pixel  $P_i$ .  $K$  represents a region of support that is the number of pixels between the starting and ending points of a given boundary point. The  $K$ -cosine contains the curvature information such that  $-1 \leq c_i(K) \leq 1$ . When  $c_i(K) = -1$ ,  $\theta = 180^\circ$ , indicating that the corresponding point is on a flat segment. When  $c_i(K)$  is close to 1, then the angle  $\theta$  is approximately  $0^\circ$ , implying the corresponding point is on a sharp angle. Thus, when  $c_i(K)$  is far from  $-1$ , then  $\vec{a}_i(K)$  and  $\vec{b}_i(K)$  are in two different segments of which point  $P_i$  is a corner point.

The chord property of a digital line stipulates that, "The line segment joining any two points of a digital straight line lies everywhere within a pixel distance,"[24]. According to this property, the corner (or joint) detection error [11] was defined as follows.

<Definition> Corner Detection Error

Given a corner with its angle  $\theta$ , its corner detection error is defined as

$$e_i = \tan\left(\frac{\theta}{2}\right). \quad (5)$$

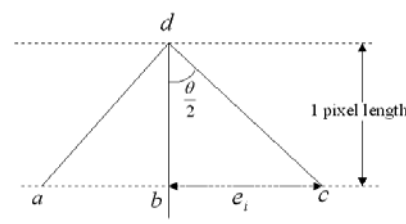


Fig 2. Corner Detection Error

As depicted in Fig. 2,  $\overline{ab}$  denotes the bisector of  $\angle adc$ , and  $e_i$  denotes the distance that the boundary moves along  $\overline{ac}$  until the distance of  $\overline{ab}$  equals one pixel length. Since  $\overline{ac}$  denotes a line, any two points on this line should lie within a pixel difference such that  $\overline{ab}$  is defined as one pixel length, indicating that  $K$  depends on the corner error  $e_i$ . However, this definition does not address digitization errors, and produces unreliable corner detection in practice.

### B. Properties of $K$

$K$  is the region of support which smoothes the curvature functions. According to the corner detection error,  $e_i$ , the lower bound is derived as follows.

<Property 1> Lower Bound ( $K_L$ ) of  $K$

Given an object with included angles  $\angle A_1, \angle A_2$ ,

...,  $\angle A_n$ ,  $K_L \in \mathbb{N}$  exists such that  $K_i \geq K_L$ ,  $\forall i = 1, 2, \dots, m$   
 where  $K_L =$

$$\begin{cases} \text{Max}\{e_1, e_2, e_3, \dots, e_n\} = \tan\left(\frac{\text{Max}\{A_1, A_2, \dots, A_n\}}{2}\right), & \text{Max}\{e_i\} \in \mathbb{N} \\ \text{Int}[\text{Max}\{e_1, e_2, e_3, \dots, e_n\}] + 1, & \text{otherwise.} \end{cases} \quad (6.a)$$

$$(6.b)$$

In Eq. (6),  $m$  denotes the number of points on the boundary;  $n$  is the number of corners in the object, and  $\text{Int}[*]$  denotes a function converting a real number to a natural number ( $N$ ). This study suggests that the value of  $K$  must be greater than the corner detection error of the largest angle ( $\text{Max}\{A_i\}$ ) to compensate for the digitization errors. That is, the lowest value of  $K$  should be the maximum of all corner detection errors. When  $\text{Max}\{e_i\}$  is not an integer, it is rounded up to the closest integer according to Eq. (6b). Meanwhile, the maximum of  $K$  should be the shortest segment of the boundary. The upper bound of  $K$  can be derived as follows.

<Property 2> Upper Bound ( $K_U$ ) of  $K$

Given an object with its boundary  $S$ , the upper bound of the  $K$ -cosine curvature is given by

$$K_U = \text{Min}_{i=1,2,\dots,n} \{ \text{Leng}(S_i) \} \quad (7)$$

In Eq. (7), the function  $\text{Leng}(S_i)$  denotes the number of points in segment  $S_i$ . The domain of  $K$  is located in  $[0, \text{Leng}(S_i)]$ .  $\text{Min}_{i=1,2,3,\dots,n} \{ \text{Leng}(S_i) \}$  denotes the boundary

segment with the lowest number of boundary points, which is the upper bound of  $K$ .

As mentioned previously, Eq. (5) does not address the effect of digitization errors, making the corner detection process unreliable. This effect is clearly demonstrated in Fig. 3. The distance  $\overline{ab}$  is over one pixel length when the corner point is missing because of digitization errors. Therefore, the corner detection error is modified by including a coefficient  $\gamma$  into the equation, and hence is derived below.

$$\begin{aligned} \tan\left(\frac{\theta}{2}\right) &= \frac{\overline{ac}}{\overline{ab}} = \frac{\overline{ac}}{\gamma} \\ \therefore \frac{e_m}{\gamma} &= \tan\left(\frac{\theta}{2}\right) \\ \therefore e_m &= \gamma \tan\left(\frac{\theta}{2}\right). \end{aligned} \quad (8)$$

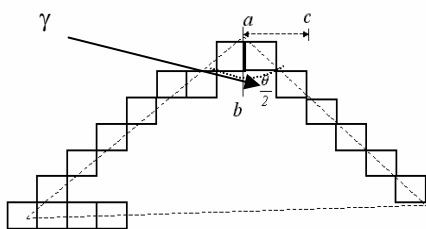


Fig 3. Modified Corner Detection Error

In accordance with the modified corner detection error,  $e_m$ , a suitable  $K$  is proposed as follows.

<Property 3>  $K_p$

Given an object with angles  $\angle A_1, \angle A_2, \dots, \angle A_n$ , and  $\theta = \text{Max}\{A_1, A_2, A_3, \dots, A_n\}$ , a  $K_p \in [K_L, K_U]$  exists such that

$$K_p = \text{Int}\left[\gamma \tan\left(\frac{\theta}{2}\right)\right] + 1, \quad (9)$$

where  $\gamma \in \mathbb{N}$ .

Accordingly,  $K_p$  is defined as the rounded-up value of  $e_m$  with various  $\gamma$  values. When the system has few digitization errors,  $\gamma$  is set to 1 ( $K = K_L$ ). Thus, a large  $\gamma$  provides reliable and consistent detection for wide-angle corners or corners with heavy digitization errors.

### C. Determination of Curvature Threshold

The objective of the curvature threshold process is to extract the boundary points with  $K$ -cosine values larger than a pre-specified threshold, and then identify the corners as the points with the highest  $K$ -cosine curvature within the extracted areas. Figure 4 shows the boundary representation of a digital object in which four corner areas are detected because the curvatures of boundary points are above a threshold,  $T$ , as depicted in Fig. 4. Accordingly, using a lower threshold discards more boundary pixels, leading to the loss of critical information, while a higher threshold may not reliably detect corners.

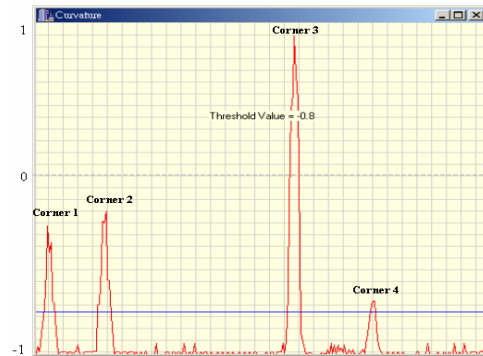


Fig 4. An example of the  $K$ - cosine threshold

To retain as much information as possible, this study derives the curvature threshold  $T$  according to  $K_p$  with various  $\gamma$  values as

$$T = \cos \left[ 2 \tan^{-1} \left( \frac{K_p}{\gamma} \right) \right], \quad (10)$$

where  $K_p$  is given in Eq. (9).

### D. KCD Algorithm

This study proposes the KCD algorithm according to the properties described. The prior information in the algorithm is the number of corners,  $n$ , and widest angle  $\theta$  in the digital image. Users must identify the value of  $\theta$  arbitrarily, or set it to an approximate large value. The

proposed algorithm is designed recursively to ensure that the correct number of corners is discovered. The algorithm is described below.

- Step 1.** Input the number of corners,  $n$ , and the estimated widest angle  $\theta$ , in the digital object.
- Step 2.** Identify the boundary point  $P_i$  in the object.
- Step 3.** Obtain  $K_L$  and  $K_U$ , and let the detected corner  $n_{detect}=0$ ,  $\gamma=1$ , and  $K=K_L$ .
- Step 4.** Calculate the  $K$ -cosine using Eq. (4).
- Step 5.** Determine the threshold  $T$  using Eq. (10).
- Step 6.** Threshold the curvature as  
 if  $c_i(K) > T, \forall i=1, 2, 3, \dots, n$ ,  
 then  $n_{detect} = n_{detect} + 1$ .
- Step 7.** Verify the number,  $n_{detect}$ , with  $n$ .  
 if  $n_{detect} \neq n$  and  $K \leq K_U$ ,  
 then  $K = K + 1$  and go to step 4,  
 else if  $n_{detect} \neq n$  and  $K > K_U$ ,  
 then  $\gamma = \gamma + 1$ ,  
 else go to Step 8.
- Step 8.** Stop.

This algorithm first sets  $K$  as the lower bound and increases  $K$  by 1 within the range of upper and lower bounds ( $K_L$  and  $K_U$ ). When  $K$  reaches the upper bound without detecting the correct number of corners, the algorithm restarts, incrementing  $\gamma$  by 1 and resetting  $K$  as the lower bound. The proposed algorithm runs until it identifies the required number of corners. Therefore, the proposed algorithm not only preserves the important information, but also reliably detects the corners.

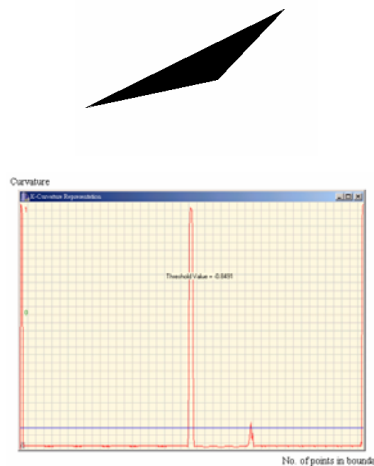
III. IMPLEMENTATION

The proposed  $K$ -cosine corner detection (KCD) method was implemented in C++ using the Borland C++ Builder on a Personal Computer with a Pentium III-750 CPU. Experiments and applications were demonstrated as shown in the following sections.

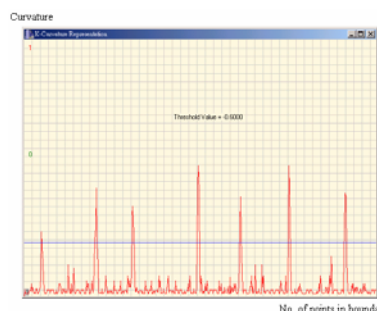
A. Experiment I – Verification

The purpose of the first experiment was to explore the ability of KCD to detect wide-angle corners and the invariant properties of translation, rotation and scaling. This experiment used six triangular objects with the widest angles,  $\theta$ , as  $60^\circ, 90^\circ, 110^\circ, 120^\circ, 145^\circ$  and  $160^\circ$ , respectively for verification. These objects were digitized in different orientations and positions using a frame grabber mounted with a CCD camera, and then segmented into binary images. The corresponding curvature boundary representations and their thresholding processes are illustrated respectively in Fig. 5. Taking the widest angle  $145^\circ$  as an example, Eq. (9) derives  $K_p = 6$ , and the threshold  $T = -0.8491$  when  $\gamma = 2$ . The proposed KCD found the three corners reliably using the derived parameters  $K$  and  $T$ . Figures 6 and 7 show the two real circular objects, which were digitized in 100%, 75% and 50% scales at randomly selected positions. The

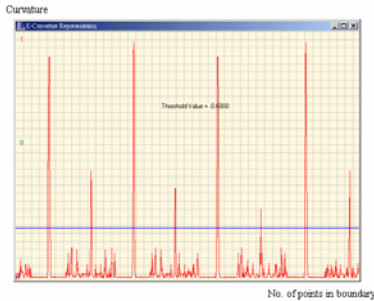
experimental results indicate that the proposed method successfully detected seven corners of the circular-curve objects in the first set of images, and eight corners in the second set of images. Additionally, two real circular objects were scanned with rotating  $30^\circ$  and  $60^\circ$  as shown in Figs. 8 and 9. Their peaks shift without altering their shapes when the objects are rotated in the curvature boundary representation. The proposed method also successfully detected the corners in spite of rotations.



(a) Triangle with  $\theta=145^\circ$  (b)  $K=6, \gamma=2$ , and  $T= -0.8491$   
 Fig 5. KCD in triangular objects with different widest angles

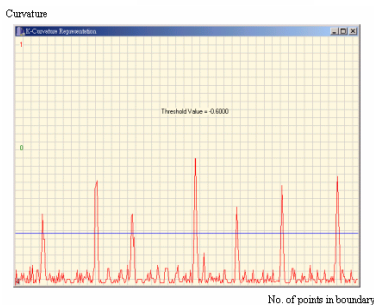


(a) Scale=100 % (b)  $K=3, \gamma=2$ , and  $T= -0.600$   
 Fig 6. KCD in circular objects (7 corners) with different scales



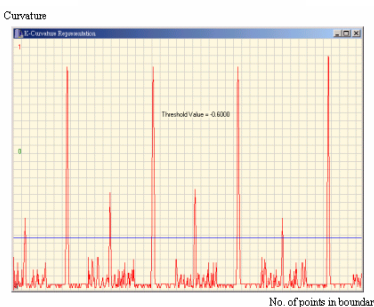
(a) Scale=100% (b)  $K=3$ ,  $T=-0.600$ ,  $\gamma=2$ , and No. of pixel=914

Fig 7. KCD in circular objects (8 corners) with different scales



(a) Rotation with 30° (b)  $K=3$ ,  $\gamma=2$ , and  $T= -0.600$

Fig 8. KCD in circular objects (7 corners) with different orientations



(a) Rotation with 30° (b)  $K=3$ ,  $T=-0.600$ ,  $\gamma=2$ , and No. of pixel=724

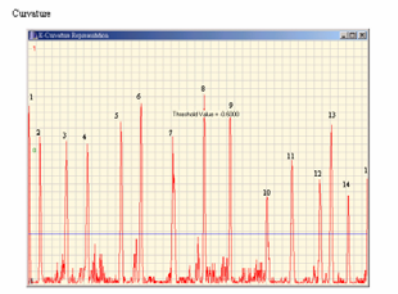
Fig 9. KCD in circular objects (8 corners) with different orientations

B. Experiment II — Benchmarks

The proposed method was benchmarked using Tsai’s methods [34]. Tsai *et al.* utilized the eigenvalue of covariance matrices to estimate the curvature of a two-dimensional boundary point over a small support region [34]. The method measured curvature information by calculating the smaller eigenvalue,  $\lambda_s$ , of the covariance matrix of a given region of support and then identified corners by setting a self-determined threshold of the curvature information of boundary points at  $T-\lambda_s$ . However, users must obtain the workable values of  $s$  and  $T-\lambda_s$  empirically. A leaf digitized with various scales was used for verification and comparison of efficiency. The corners of the leaf image, depicted in Fig. 10a, b, were both successfully identified by Tsai’s method and the proposed KCD, but KCD provides  $K$ ’s and curvature thresholds as shown in Fig. 10c. Considering computation speed, the complexity of Tsai’s method is  $O(N \times s^2)$ , where  $N$  denotes the number of points on the boundary, and  $s$  denotes the number of points in the region of support, when calculating the covariance matrix and eigenvalues. When computing the  $K$ -cosine, the complexity of KCD is  $O(N \times K)$ , where  $K$  denotes the previously defined region of support. Figure 11 compares KCD and Tsai’s method in terms of computation speed given the same region of support and varying numbers of boundary points.



(a) Original image of a leaf (b) Image after threshold



(c) KCD with  $K=6$ ,  $T=-0.600$ ,  $\gamma=3$

Fig 10. KCD with a leaf image

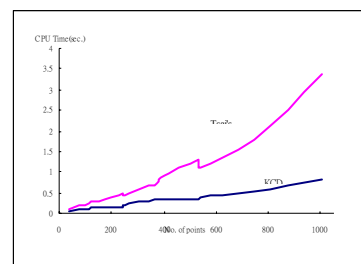
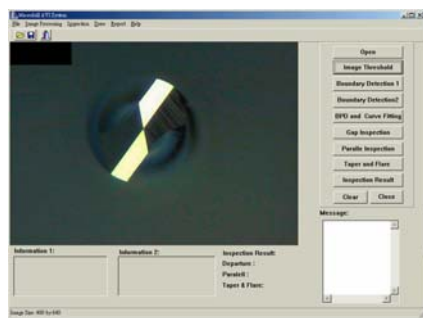


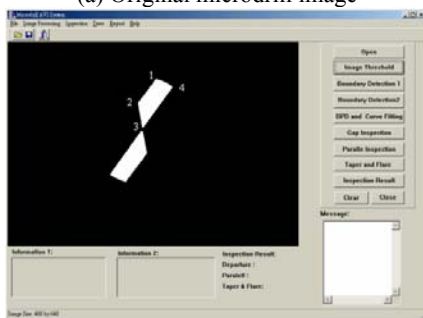
Fig 11. Comparison of KCD and Tsai’s method in computation speed

IV. APPLICATIONS

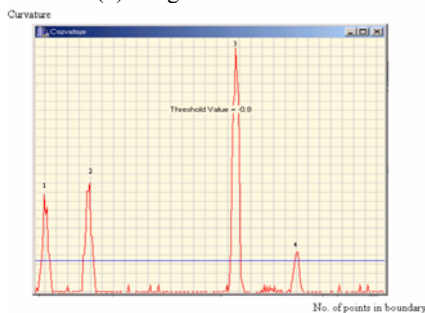
The proposed method has been practically implemented in C++ on the BCB 6.0 environment under the Windows XP platform. It was applied to the inspection of microdrills to identify the defects of flare, taper, gap, and overlap on the microdrill blades. Owing to the microdrill size, alignment is difficult in practice. Therefore, the microdrill is placed under a CCD camera and digitized without a fixture. The derived image was first thresholded to derive the blade of the microdrill, and then a boundary following process was applied to obtain the boundary of blades. Finally, the KCD method was used to separate the boundary into segments with  $K=5$ ,  $T=-0.800$ , and  $\gamma=2$ . In this case study, the proposed KCD method reliably detected the eight corners of microdrill blades, and detected the defects with pre-defined specifications. An original microdrill is shown in Fig. 12a. It was thresholded with a predetermined threshold and separated the blade and its background as shown in Fig. 12b and c.



(a) Original microdrill image



(b) Image after threshold

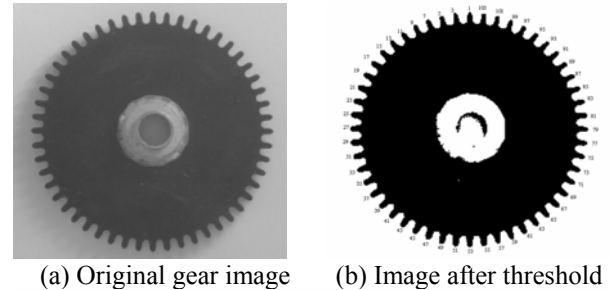


(c) KCD with  $K=5$ ,  $T=-0.800$ ,  $\gamma=2$

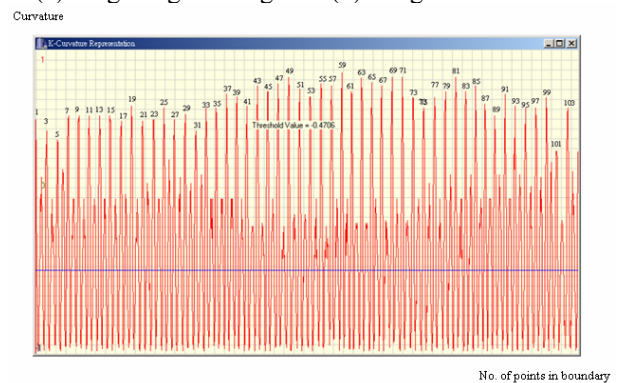
Fig. 12. Application of KCD to microdrill inspection

Moreover, the proposed KCD has been also for a gear inspection task, which was inspected manually by sampling in the production line. The proposed method was implemented to examine the 104 gear teeth automatically as shown in Figs. 13 a, b and c. The proposed KCD used  $K=4$ ,

$T=-0.4706$ , and  $\gamma=3$  to identify each individual tooth of a gear, and subsequently inspection the required features of gears with specification. The proposed method successfully released the human inspector in tedious tasks, and increased the reliability of inspection.



(a) Original gear image (b) Image after threshold



(c) KCD with  $K=4$ ,  $T=-0.4706$ ,  $\gamma=3$

Fig. 13. Inspection of Gear

V. CONCLUSIONS

This study presents a position, orientation and scale invariant boundary-based corner detection method. For effectiveness and simplicity, the boundary of an object is first represented with curvature, which is measured using  $K$ -cosine. Then, the corner detection error is modified by including an effective coefficient  $\gamma$ , and recommends the use of  $K_p$  and the curvature threshold ( $T$ ) to detect corners reliably.

Experiments were performed to demonstrate the effectiveness and efficiency of the proposed KCD method. KCD's ability to detect wide-angle corner points was tested using a set of triangular objects. The experimental results show that the proposed method successfully identifies the corners. Next, the proposed method was verified with a set of circular-curved objects, used in [34] as benchmarks, for the effect of various positions, scales and orientations. The experimental results reveal that KCD reliably detected the desired corners under various positions, scales and orientations. Eventually, two practical case studies -- microdrill inspection and gear inspection are presented.

REFERENCES

[1] F., Arrebola ,A., Bandera, P., Camacho, and F., Sandoval, "Corner detection by local histograms of the contour chain code," *Electronics Letter*, vol. 33 (21), pp. 1769-1771, 1997.

- [2] H., Asada, and M., Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8 (1), pp. 2-14, 1986.
- [3] A. Bandera, C., Urdiales, F., Arrebola, and F., Sandoval, "2D object recognition based on curvature functions obtained from local histograms of the contour chain code," *Pattern Recognition Letters*, vol. 20, pp. 49-55, 1999.
- [4] J.-E., Byun, and T., Nagata, "Determining the 3-D pose of a flexible object by stereo matching of curvature representations," *Pattern Recognition*, vol. 29 (8), pp. 1297-1307, 1996.
- [5] D.P., Fairney, and P.T., Fairney, "On the accuracy of point curvature estimators in a discrete environment," *Image and Vision Computing*, vol. 12 (5), pp. 259-265, 1994.
- [6] H., Freeman, and L.S., Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. Comput. C-26*, pp. 297-303, 1977.
- [7] N., Friedland, and A., Rosenfeld, "An integrated approach to 2D object recognition," *Pattern Recognition*, vol. 30 (3), pp. 525-535, 1997.
- [8] J.A., Garcia, J., Fdez-Valdivia, and R., Molina, "A method for invariant pattern recognition using the scale-vector representation of planar curves," *Signal Processing*, vol.43, pp. 39-53, 1995.
- [9] M.H., Han, and D., Jang, "The use of maximum curvature points for the recognition of partially occluded objects," *Pattern Recognition*, vol. 23, pp. 21-22, 1990.
- [10] J.M. Inesta, M., Buendia, and M.A., Sarti, "Reliable polygonal approximations of imaged real objects through dominant point detection" *Pattern Recognition* vol. 33 (6), pp. 685-697, 1998.
- [11] J.I. Ker, "Automated Profile Inspection Methods by Computer Vision Methods," Ph.D. Dissertation, Department of Industrial Engineering, University of Missouri-Columbia, 1989.
- [12] J.H., Kim, S.H., Yoon, and K.H., Sohn, "A robust boundary-based object recognition in occlusion environment by hybrid Hopfield neural networks," *Pattern Recognition*, vol. 29 (12), pp. 2047-2060, 1996.
- [13] J.S. Lee, Y.N., Sun, and C.H., Chen, "Multiscale corner detection by using wavelet transform," *IEEE Transactions on Image Processing*, vol. 4 (1), 100-104, 1995.
- [14] L., Li, and W., Chen, "Corner detection and interpretation on planar curves using fuzzy reasoning," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 (11), 1024-1210, 1999.
- [15] K.B., Lim, K. Xin, and G.S., Hong, "Detection and estimation of circular arc segments," *Pattern Recognition Letters*, vol. 16, pp. 627-636, 1995
- [16] H.C. Liu, and D., Srinath, "Corner detection from chain-code," *Pattern Recognition*, vol. 23, pp. 51-68, 1990.
- [17] X., Li, M., Parizeau, and R. Plamondon, "Segmentation and reconstruction of on-line handwritten scripts," *Pattern Recognition*, vol. 31 (6), pp. 675-684, 1998.
- [18] R., Mehrotra, and S., Nichani, "Corner detection," *Pattern Recognition*, vol. 23 (11), pp. 1223-1233, 1990.
- [19] F. Mokhtarian, and A., Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8 (1), pp. 34-43, 1986.
- [20] F., Mohktarian, and R., Suomela, "Robust image corner detection through curvature scale space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20 (12), pp. 1376-1381, 1998.
- [21] J.S., Park, and J.H., Han, "Contour matching: a curvature-based approach," *Image and Vision Computin*, vol. 16, pp. 181-189, 1998.
- [22] A., Rattarangsi, and R.T., Chin, "Scale-based detection of corners of planer curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14 (4), pp. 430-449, 1992.
- [23] A., Rosenfeld, and E., Johnson, "Angle detection on digital curves," *IEEE Transactions on Computers C-22*, pp. 875-878, 1973.
- [24] A., Rosenfeld, "Digital straight line segments," *IEEE Transaction on Computer C23* (12), vol. 1264-1268, 1974.
- [25] P.L., Rosin, "Augmenting corner descriptors," *Graphical Models and Image Processing*, vol. 58, pp. 286-294, 1996.
- [26] F.J., Sánchez-Marín, "Automatic recognition of biological shapes with and without representations of shape," *Artificial Intelligence in Medicine*, vol. 18 (2), 173-186, 2000.
- [27] H.T., Sheu, and W.C., Hu, "A rotationally invariant two-phase scheme for corner detection," *Pattern Recognition*, vol. 29 (5), 819-828, 1996.
- [28] A., Singh, and M., Shneier, "Gray level corner detection a generalization and a robust real time implementation," *Comput. Vision Graphics Image Process*, vol. 51, pp. 54-69, 1990.
- [29] K., Sohn, W.E., Alexander, J.H., Kim, and W.E., Snyder, "A constrained regularization approach to robust corner detection," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24 (5), pp. 820-828, 1994.
- [30] K., Sohn, J.H., Kim, W.E., Alexander, "A mean field annealing approach to robust corner detection," *IEEE Trans. System Man Cyber.*, Part B: Cybernetics, vol. 18 (1), pp. 82-90, 1998.
- [31] C.H., Teh, and R.T., Chin, "On the detection of dominant points of digital curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11 (8), pp. 859-872, 1989.
- [32] D.M., Tsai, "Boundary-based corner detection using neural networks," *Pattern Recognition*, vol. 30 (1), pp. 85-97, 1997.
- [33] D.M., Tsai, and M.F., Chen, "Curve fitting approach for tangent angle and curvature measurements," *Pattern Recognition*, vol. 27, pp. 699-711, 1994.
- [34] D.M., Tsai, H.T., Hou, and H. J., Su, "Boundary-based corner detection using eigenvalues of covariance matrices," *Pattern Recognition Letters*, vol. 20, pp. 31-40, 1999.

**Te-Hsiu Sun** is currently an associate professor in the Department of Industrial Engineering and Management at Chaoyang University of Technology, Taiwan. He received the Bachelor's degree in Mechanical Engineering from the Chung Yuan University, Taiwan, in 1987, and the Ph.D. degree in Industrial Education from the National Changhua University of Education in 2002. His present research interests include applications of machine vision, image processing, human resource management, and organizational management.