



Effective and efficient contour-based corner detectors

Shyh Wei Teng ^{a,*}, Rafi Md. Najmus Sadat ^b, Guojun Lu ^a

^a School of Engineering and Information Technology, Faculty of Science and Technology, Federation University Australia, Gippsland campus, Churchill, Australia

^b Faculty of Information Technology, Monash University, Churchill, Australia

ARTICLE INFO

Article history:

Received 13 March 2014

Received in revised form

23 December 2014

Accepted 21 January 2015

Available online 29 January 2015

Keywords:

Corner detector

Contour

Curvature

ABSTRACT

Corner detection is an essential operation in many computer vision applications. Among the contour-based corner detectors in the literature, the Chord-to-Point Distance Accumulation (CPDA) detector is reported to have one of the highest repeatability in detecting robust corners and the lowest localization error. However, based on our analysis, we found that the CPDA detector often fails to accurately detect the true corners when a curve has multiple corners but the sharpness of one or a few of them is much more prominent than the rest. This detector also might not perform well when the corners are closely located. Furthermore, the CPDA detector is also computationally very expensive. To overcome these weaknesses, we propose two effective and efficient corner detectors using simple triangular theory and distance calculation. Our experimental results show that our proposed detectors outperform CPDA and nine other existing corner detectors in terms of repeatability. Our proposed detectors also have a relatively low or comparable localization error and are computationally more efficient.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Detecting corners is an essential operation in many computer vision and image processing applications such as motion tracking, shape representation, image registration, camera calibration, object recognition and stereo matching. A corner can be defined as a location on an edge where the angle of the slope changes abruptly i.e. where the absolute curvature is high [1].

Corners are important features in two-dimensional (2D) images as they can represent the shape of an object very well. Therefore, corner detection plays an important role in image matching and pattern recognition. We can broadly classify the corner detectors into two groups [2] – intensity-based and contour-based. Intensity-based corner detectors [3–6] directly deal with the intensity values (or grey level pixel values) but not with the shape or any edge information of the image. On the other hand, contour-based detectors [7–12] first extract the curves (or contours) from the image by using an edge detector, and then identify the locations which have salient information or maximal curvature. Most intensity-based corner detectors are based on image derivatives that is why they are more sensitive to noise. Contour-based corner detectors, however, are generally less sensitive to noise as they are not based on image derivatives and they also apply Gaussian smoothing to remove the noises from the contours. This paper focuses on

contour-based detectors as they are generally more effective compared to intensity-based detectors [2].

The Chord-to-Point Distance Accumulation (CPDA) technique [13] is a way of estimating the curvature values of 2D planar curve using a single chord. Later, Awrangjeb and Lu proposed a strong angle detector based on the CPDA technique with multiple chords and this detector is one of the best contour-based corner detectors reported in the literature [14]. This CPDA detector uses the chords, which intersect curve segments of different lengths, to estimate curvature values on each point along the curves extracted by an edge detector. The estimated curvature values of each chord are then normalized. Next, the curvature values estimated using the chords at each point are multiplied to obtain the final curvature values. Finally, points corresponding to the local maxima of the multiplied values are chosen as candidate corners and these corners are further refined to determine the final set of corners.

Although the CPDA detector is reported to achieve one of the highest repeatability and the lowest localization error among existing compatible detectors in the literature, we found that the CPDA detector has the following weaknesses. Firstly, the curvature values estimated by the CPDA detector are not proportional to the original angle of the corner. Secondly, it fails to perform well when a curve has multiple corners but the sharpness of one or a few of them is much more prominent than the rest. Thirdly, it has the potential to miss some corners on curves which have several corners closely located to each other. The second and third weaknesses are mainly caused by the process in the CPDA detector whereby the curvature values derived by the chords are combined

* Corresponding author. Tel.: +61 3 51226851.

E-mail addresses: shyh.wei.teng@federation.edu.au (S.W. Teng), guojun.lu@federation.edu.au (G. Lu).

to derive the final curvature values. This might cause the loss of maxima curvature values at some corner locations. Finally, the CPDA detector is computationally expensive due to the complexity of its curvature estimation and its refinement processes. How the CPDA detector works and its weaknesses will be discussed in greater detail in [Sections 3 and 4](#) respectively.

In this paper, we have proposed two approaches for contour-based corner detection to address the potential weaknesses of the CPDA detector and various other existing contour-based corner detectors. One of the proposed corner detectors uses a simple triangular theory for curvature estimation and the other is based on the ratio of the curve length distance between two pixels to the direct Euclidean distance of these two pixels on the curve. Our proposed detectors are able to overcome all the aforementioned weaknesses of the CPDA detector. Our experimental results also show that our proposed detectors not only achieve the best repeatability and comparably low localization errors in comparison with the CPDA detector as well as the other existing compatible detectors. They are also the most efficient corner detectors.

The rest of this paper is organized as follows. A selection of the main contour-based corner detectors is described in [Section 2](#). An overview of the CPDA detector and its weaknesses are discussed in [Sections 3 and 4](#) respectively. [Section 5](#) presents the proposed corner detectors. [Section 6](#) discusses how the proposed techniques overcome the weaknesses of the CPDA detector and the complexity of the corner detectors. Finally, [Section 7](#) presents the experimental results and [Section 8](#) concludes the paper.

2. Related works

In this section, we discuss a selection of the main contour-based corner detectors and their weaknesses. The performance of our proposed detectors will be evaluated against majority of these detectors in [Section 7](#).

Dominant points detectors, e.g. [15–17], are one of the earliest group of techniques which extract the strong corners from digital curves. This group of techniques refers to the locations of the strong corners as dominant points on the curve. Generally, for each candidate corner, these techniques first establish the region of supports (RoSs) which are the two sides of the corner. Then based on the RoSs, the curvatures of these candidate corners are measured. Finally, the strong corners are determined using certain rule-based or optimization algorithms. Ref. [17], proposed by Poyato, is one of the more recent and best performing techniques in this group. It establishes the RoSs using chain-code and then uses an optimization algorithm to detect the stronger corners that can still retain the general shape of the curve. This group of techniques works well if the curve representing the boundary of each object in an image can first be accurately and completely extracted. Unfortunately, this is difficult to achieve in many applications which process real images.

The Curvature Scale Space (CSS) [7] corner detector is also one of the earliest contour-based detectors. It first uses a coarse smoothing scale to estimate a curvature value for each pixel along the curve and then identifies approximate locations of the corners. Next, it uses a finer scale to track these locations to improve the localization of these corners. The main weakness of this CSS detector is in selecting an appropriate scale for identifying the approximate locations of the corners. If a coarser scale was used, the detector would be more robust to noise, but might miss many potential corners. However, if a finer scale was used, the detector would be sensitive to noise and would detect many spurious corners. The enhanced CSS [8] detector attempted to solve this weakness by using different scales for curves with different lengths. However, choosing the right set of scales for various curves' length is still difficult. The use of coarse scales on the initial

step for curvature estimation is still causing high localization errors. Another weakness of these detectors is that they are not invariant to affine transformation as both detectors use the arc-length parametrization for every point on an extracted curve. Apart from these weaknesses, these CSS detectors estimate curvature values using the derivatives which are computed based on a very small neighbourhood. This makes the detectors very sensitive to the local variations and noise.

To overcome the weaknesses of the CSS detectors described above, several detectors which use multiple scales for curvature estimation at each point on a curve are proposed. Awrangjeb et al. proposed a multiscale detector (ARCSS) [11] which uses three different scales and affine-length parametrizations instead of the arc-length to detect the corners. However, this detector is computationally very expensive due to the calculation of the affine-length parametrizations. The multiscale curvature product (MSP) [12] detector is another CSS-based detector which multiplies the curvature values derived using three scales to make the strong corners more distinguishable from the noise and weak corners. He and Yung proposed two variations [9,10] of a CSS-based detector that use an adaptive local threshold according to its neighbourhood region's curvature and then detect the angle on a proper RoS.

There are also detectors that apply other mechanisms to process the curve before deriving the corners. Zhang et al. [18] proposed a detector which applies multiple levels of Difference of Gaussian (DoG) on a curve to obtain several corresponding planar curves. These planar curves are then used for detecting the corners. Zhang's detector is reported to have higher corner detection rate compared to other CSS detectors. However, as derivative is used for curvature estimation, this detector is still sensitive to noise, thereby lowering its repeatability. Another group of detectors [19–21] applies wavelet transform to the curve to derive multiple wavelets for representing its contour orientation. However, as the wavelets acquired from wavelet transform are similar to the second derivative of the curve, these detectors can still be sensitive to noise. The last group of detectors we discuss here uses various forms of matrix manipulation, e.g. eigenvalues of the covariance matrix [22] and gradient correlation matrix [23], for processing the curve to locate the corners. Generally, these detectors are computationally highly complex due to the matrix manipulation. A more detailed review of all the techniques discussed in this section can be found in [24].

3. Overview of CPDA detector

Similar to CSS-based corner detectors, CPDA detector [14] also starts by detecting curves from the images and finding the T-junctions. Each extracted curve is smoothed with an appropriate Gaussian kernel (i.e. $\sigma=1, 2$, or 3) depending on its length to remove the noise from it.

Next, three chords which are defined as L_i , where $i \in \{10, 20, 30\}$, are moved along each curve. In [Fig. 1](#), let $P_1, P_2, P_3, \dots, P_N$ be the N points on a curve. So, value i of chord L_i defines the number of points (or pixels) of the curve segment between points P_j and P_{j+i} . Chord L_i is the straight line that intersects points P_j and P_{j+i} on the curve. To estimate the curvature value $h_{L_i}(q)$ at point P_q using chord L_i , the chord is moved on each side of P_q for at most i points while keeping P_q as an

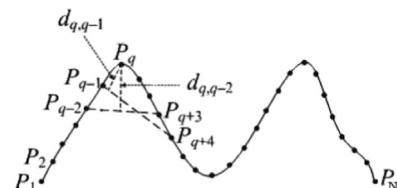


Fig. 1. Curvature estimation at a point using CPDA with chord of length L .

interior point and the distances $d_{q,j}$ from P_q to the chord are calculated. Finally, the CPDA detector accumulates the curvature estimation using

$$h_{L_i}(q) = \sum_{j=q-i+1}^{q-1} d_{q,j} \quad (1)$$

The curvature values estimated using each chord are normalized using Eq. (2) and then multiplied together using Eq. (3):

$$h'_{L_i}(q) = \frac{h_{L_i}(q)}{\max(h_{L_i})}, \quad \text{for } 1 \leq q \leq N, \quad i \in \{10, 20, 30\} \quad (2)$$

$$H(q) = h'_{L_{10}}(q) \times h'_{L_{20}}(q) \times h'_{L_{30}}(q), \quad \text{for } 1 \leq q \leq N \quad (3)$$

where h_{L_i} is the set of curvature values estimated for all points on the curve using chord L_i . The local maxima of $H(q)$ determine the locations of the candidate corners. Finally, to filter out the weak and false corners among the candidate corners, a two-step refinement process is used.

The first refinement step is to remove all weak corners among the candidate corners. This is done by removing all candidate corners which have their final curvature values lower than a curvature threshold (T_h). Fig. 2(b) shows an example of the corners after removing the weak corners. The second step is to remove all false corners. Here, the angle from a candidate corner to its two neighbouring candidate corners (or curve-ends when insufficient neighbouring candidate corners are present) is calculated and compared with an angle-threshold. The candidate corner is considered as a false corner if the angle is greater than the angle-threshold.

4. Weaknesses of CPDA detector

It is reported that the CPDA detector outperforms many other existing compatible detectors [14]. However, from our analysis, we found that it has the following weaknesses.

An important property of a corner detector is that the curvature values estimated to measure the sharpness of the corners must be proportional to the angle of the corners. This means the estimated curvature value should increase (or decrease depending on the measure used) proportionally with the increased angle of the corner. However, the CPDA detector could not achieve this property. Fig. 3 shows three different angles and the first three rows of Table 1 present the curvature values for the corner locations estimated by three different chords. According to these curvature values, the smallest angle has the highest curvature values and the second smallest angle has the lowest value. The reason is the CPDA detector that uses the perpendicular distance from the candidate location to the chord which can be the same for many angles. In Fig. 4, we can relate the angles $\angle BAC$, $\angle CAD$ and $\angle BAD$ to the angles

of Fig. 3(a), (b) and (c) respectively. Now, Lines BC and BD are the corresponding chords for the angles $\angle BAC$ and $\angle BAD$, considering that there is the same number of pixels in AC and AD . Now, both angles have the same perpendicular distance, AE , to their respective chords. For this reason, the corners of Fig. 3(a) and (c) have similar CPDA curvature values.

Besides the above-mentioned weakness related to individual chords used by the CPDA, we have also found the following weaknesses in how the CPDA detector combines the individual curvature estimations. CPDA detector fails to detect corners on a curve if some of them are very sharp while the sharpness of the other corners are relatively much smaller. In this scenario CPDA detector could fail to detect the less sharp corners. This is because the curvature values estimated using a chord are first normalized by using the highest value of that chord. Then when the final curvature values are derived by multiplying the corresponding curvature values from the three chords together, the differences between the final curvature values of the very sharp and less sharp corners could be exaggerated so much that the final curvature values of the less sharp corners fall below the curvature threshold set. This is even though such corners are visually prominent. To illustrate this, we will use the Curve A of Lena image [25] shown in Fig. 2(c). Curve A has two visually prominent corners (i.e. x and y); one of them is very sharp (i.e. x) and the other is relatively less sharp (i.e. y). CPDA detector fails to detect the Corner y as shown in Fig. 2(c). This is due to the reason we stated above and is reflected in Fig. 5: (a), (b) and (c) show the curvature values of the three chords L_{10} , L_{20} and L_{30} respectively. (d) shows the final curvature values and we can see that the value for Corner y is lower than the curvature threshold (i.e. 0.2 as recommended in [14]) even though it is visually prominent.

We have also found that, the CPDA detector might potentially miss obvious corners if they were located closely. This is due to the use of chords intersecting curve segments with more number of pixels apart. For example, Fig. 2(c) and (d) shows that the Corner w is not detected by the CPDA detector. As shown in Fig. 6(a), Chord L_{10} can detect the local maxima at Corner w , however, the second and third chords (L_{20} and L_{30}) cannot (Fig. 6(c) and (d)). After multiplying these curvature values, the final curvature value representing Corner w will be too low to be detected as a corner by the CPDA detector. There are also some scenarios where the multiplication of three chords' estimated curvature values is responsible for losing the extrema for the corner location. For example, the corner at the bottom of the shape in Fig. 17(c) is not detected. In this case, the chords generate the maxima in three different locations and the product of the curvature values becomes less than the specified threshold. A faster version of the CPDA detector [26] has also been proposed where the curvature values are estimated on selected DoG points only. Although the

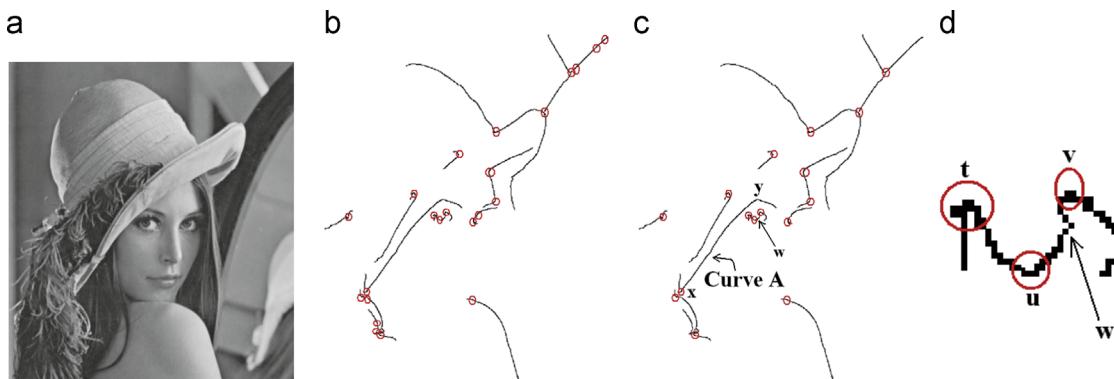


Fig. 2. (a) Lena image (used with permission from authors of [14]); (b) and (c) show the corners (denoted by 'o') detected by the CPDA after the first and second refinement steps respectively; (d) shows the enlarged version of the curve where Corner w (which CPDA has not detected) is located.

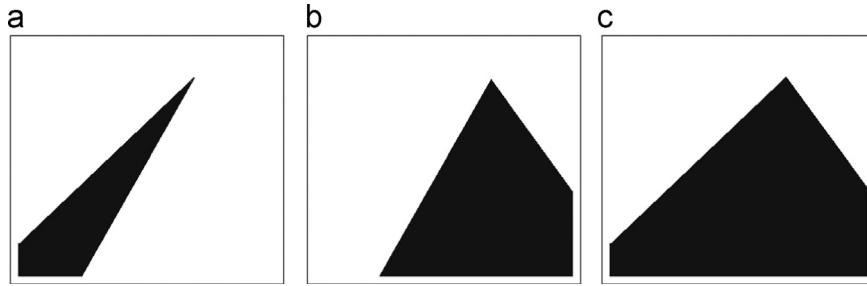


Fig. 3. Three different angles to show the proportionality of the CPDA curvature estimation.

Table 1

Estimated curvature values by the CPDA detector and our proposed detectors.

Detector	Chord length (L_i)	Fig. 3(a)	Fig. 3(b)	Fig. 3(c)
CPDA	10	4.44	3.62	4.00
	20	10.21	8.59	9.00
	30	15.66	13.56	14.00
CTAR	7	0.70	0.89	0.94
CCR	7	0.67	0.87	0.92

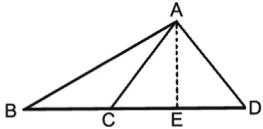


Fig. 4. Perpendicular distances to the CPDA chord for different angles.

Fast CPDA detects the corners relatively quicker than the original version, it achieves relatively lower repeatability.

5. Proposed detectors

In this section, we propose two corner detectors that can overcome the weaknesses of the CPDA (a chord-based corner detector) and various CSS-based corner detectors we have discussed in previous sections. For both proposed detectors, the measures used to estimate the curvature value are discussed first. Next, the corner detectors using the curvature estimation measures are proposed.

5.1. Proposed detector using triangular theory

5.1.1. Curvature estimation measure

Instead of using distance accumulation, a simple but yet effective measure based on a triangular theory is used to estimate the curvature values. To apply this measure, a chord is first moved along the curve in a way similar to the CPDA curvature estimation. Every time the chord is placed on the curve, a new triangle can be formed using the two ends of the chord and the midpoint on the curve segment between the two ends of the chord. The ratio of the length of the chord to the summation of the length of the other two arms of the triangle, which are from the midpoint of the curve segment to each respective ends of the chord, is computed. The value of this ratio is the estimated curvature value for the midpoint on the curve segment. This measure is less sensitive to noise which is one of the weaknesses of CSS-based corner detectors. This is because it does not use any derivative based measurements. Furthermore, it also uses a relatively bigger neighbourhood.

We illustrate the above measure with an example in Fig. 7. Let P_1, P_2, \dots, P_N be the N points of a curve and P_i be the point where the curvature value is to be estimated. Now, we traverse k pixels from P_i in the right direction to pixel P_{i+k} and then, k pixels from P_i in the reverse direction to pixel P_{i-k} . If the three pixels P_{i-k}, P_i and P_{i+k} are collinear, the ratio of the length of the chord from P_{i-k} to P_{i+k} to the summation of the length of the other two arms of the triangle from P_i to P_{i-k} and P_{i+k} respectively is 1; otherwise the ratio is less than 1. The value of the ratio will decrease as the sharpness of the corner at P_i increases. Now, the curvature value of point P_i on the curve is estimated using

$$R_1(P_i) = \frac{d_1}{d_2 + d_3} \quad (4)$$

where

$$\begin{aligned} d_1 &= \sqrt{(x_{p_{i-k}} - x_{p_{i+k}})^2 + (y_{p_{i-k}} - y_{p_{i+k}})^2} \\ d_2 &= \sqrt{(x_{p_i} - x_{p_{i-k}})^2 + (y_{p_i} - y_{p_{i-k}})^2} \\ d_3 &= \sqrt{(x_{p_i} - x_{p_{i+k}})^2 + (y_{p_i} - y_{p_{i+k}})^2} \end{aligned}$$

5.1.2. Proposed corner detector

In this section, we describe how the curvature estimation measure described above is used in our proposed detector. The proposed curvature estimation measure is more intuitive for determining the cornerness of a location on a curve. Furthermore, the calculation of the ratio is also more simple and efficient than the CPDA technique which requires multiple distances for estimating curvature of a single point.

Similar to the CSS-based corner detectors, our proposed detector also starts with detecting the curves from the image and finding out the T-junctions. Next, we apply the Gaussian smoothing ($\sigma=3$) to reduce the noise on the curve.

Unlike the CPDA detector which uses multiple chords, we traverse only one chord (L_7) along the curve so that the detector does not lose the maxima at two nearby corners (see the weaknesses discussed in Section 4). Note that, L_i denotes i number of pixels on the curve segment intersected by the chord. We have chosen the value of k as 3. After estimating the curvature values, the local minima are found from each curve's estimation. We consider the locations of the minima as corners if the estimated curvature value is less than a threshold ($T=0.989$). Although the threshold value is close to 1, this value corresponds to around 163° . The CPDA detector [14] also uses a similar angle threshold which is 157° . How we mathematically derived the angle size from our proposed threshold is shown in the appendix. The empirical results of the parameter settings are shown in Section 7.2. Finally, the T-junctions are added to the final set of corners if any location near (5×5 window) the T-junctions is not detected as a corner [7]. We name our proposed detector as Chord to Triangular Arms Ratio (CTAR) detector.

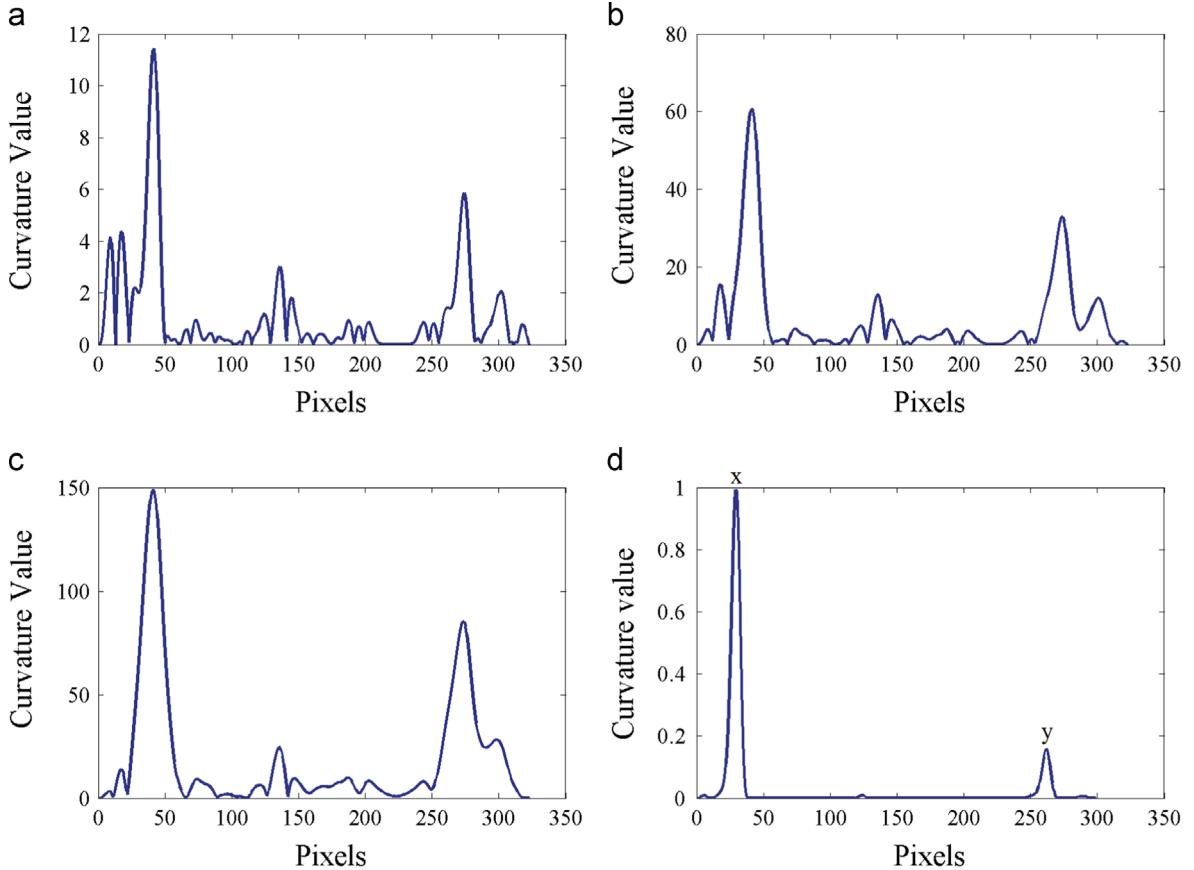


Fig. 5. (a), (b) and (c) Curvature estimation of Curve A of Lena image [25] using chords L_{10} , L_{20} and L_{30} respectively; (d) combined curvature estimation using Eqs. (2) and (3).

5.2. Proposed detector using distance ratio

In this section, we propose another approach to estimate the curvature value on the curve. The corners are usually found at locations where the slopes of the contour change significantly. Therefore the basic task to detect the corners is to measure the flatness of the curve. One way is to put a straight line segment along the curve in such a way that the two ends of the line touch the curve and calculate the ratio of the actual length of that line to the curve length between two ends of that line. A perfect flat curve will give the ratio as 1 for this measurement. The more bent the curve, the lower the ratio will be.

To illustrate how this detector works, we will again use the example in Fig. 7. Similar to CTAR, to estimate the curvature value at location P_i , we traverse k pixels in the right direction of pixel P_i to pixel P_{i+k} and then in the reverse direction, k pixels of the pixel P_i to pixel P_{i-k} . The straight line segment is the chord which connects the pixels P_{i-k} and P_{i+k} . Therefore, the number of pixels within these two ends on the curve is $2k+1$. Primarily, we assume the number of pixels ($L=2k+1$) on the curve segment intersected by the chord as the length of the curve segment. So, we calculate the ratio of the Euclidean distance of pixels P_{i-k} and P_{i+k} to the number of pixels using

$$R_2(P_i) = \frac{\text{distance}(P_{i-k}, P_{i+k})}{L} \quad (5)$$

Here, the function `distance` gives the Euclidean distance between two given locations. The ratio calculated using Eq. (5) is assigned to pixel P_i which is the midpoint within the two ends of the chord. By moving the chord along the curve, each pixel will have an estimated curvature value which reflects the sharpness of the slope change at that location. Now, each local minimum which

is less than a threshold represents a corner. In Fig. 8, the ratio values are the same for the pixels which are on the straight lines and each minimum is found for each corner in the image. There are 577 pixels in the curve and the calculation starts from the filled circle in the direction indicated in the figure. The corresponding corners in Fig. 8(a) and the corresponding minima in Fig. 8(b) are denoted with same labels.

Since, Fig. 8(a) does not have any noise on the curve, it is easy to detect the corner locations. No additional minima are found without the actual corner locations. However, it will not be easy to locate the corners using this approach in case of a bent curve having noises. For example, Fig. 9 shows a shape having bent curve and the corresponding ratio measurement of the shape using Eq. (5). Although peaks at the corner locations are found, many other peaks are also found. More importantly, some of the points have curvature values greater than 1, which is not expected. This observation reveals that the calculation of the ratio using the above-mentioned approach will not work on real life images.

To overcome this weakness, we calculate the distance between each consecutive location on the curve before moving the chord along the curve so that the curve length of any two locations can be perfectly measured:

$$D(P_i) = \sqrt{(x_{p_i} - x_{p_{i+1}})^2 + (y_{p_i} - y_{p_{i+1}})^2} \quad (6)$$

We can derive the distances of all the consecutive pixels on the curve using Eq. (6). Now, the ratio of the length of the chord to the curve length of the chord can be accurately calculated using

$$R_3(P_i) = \frac{\text{distance}(P_{i-k}, P_{i+k})}{\sum_{i=k}^{i+k-1} D(P_i)} \quad (7)$$

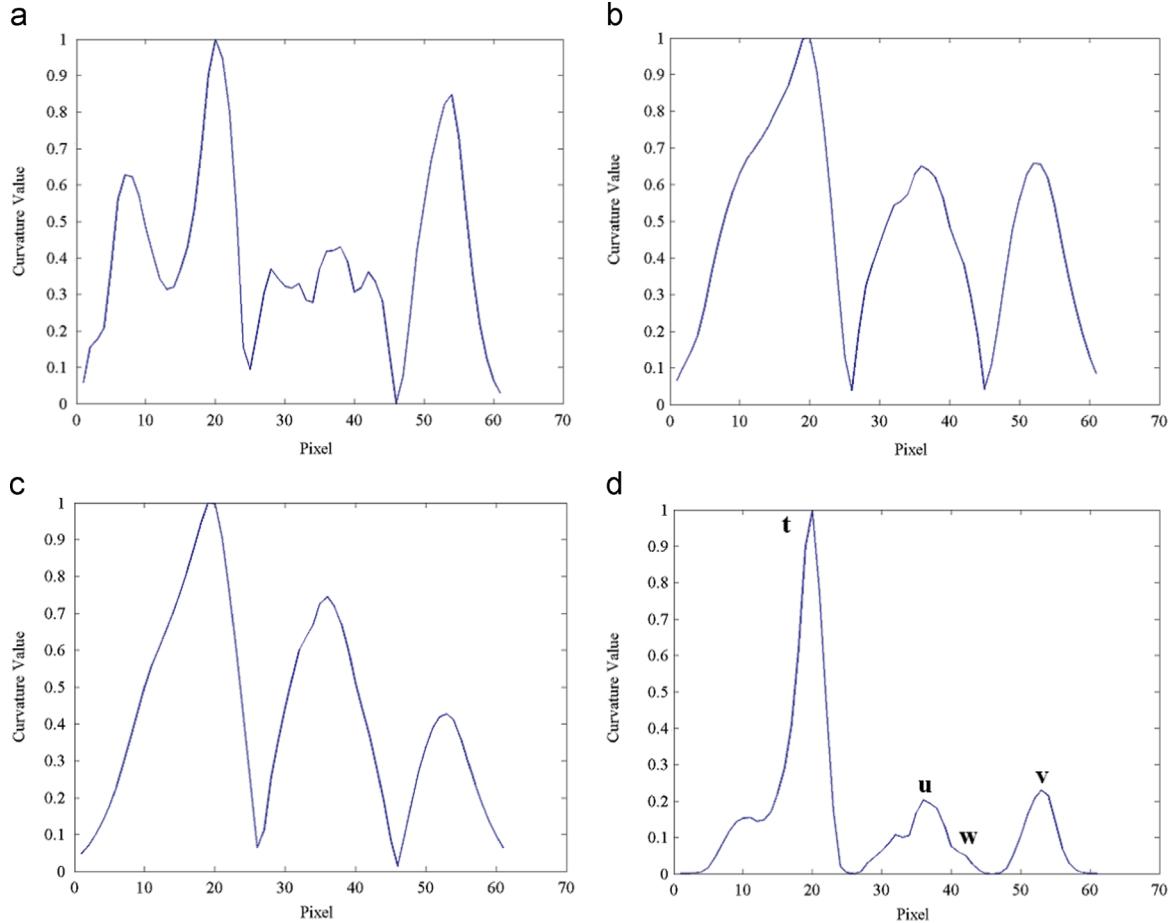


Fig. 6. Curvature values of the curve in Fig. 2(d); (a), (b) and (c) Curvature values estimated using chords L_{10} , L_{20} and L_{30} respectively; (d) the final curvature values.

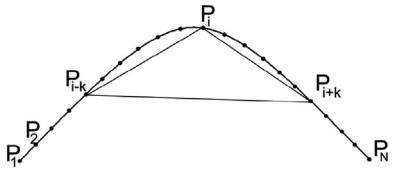


Fig. 7. Curvature estimation measure used in proposed detectors.

Fig. 10 shows the ratio of Fig. 9(a) using the approach mentioned above. One of the goals of having the ratio not greater than 1 has been achieved. However, there are still a few more unwanted extrema. So, Gaussian smoothing is applied to reduce the unwanted extrema before calculating the ratio. Fig. 11 shows the calculated ratio of Fig. 9(a) after applying the Gaussian filter and its corresponding curvature estimation using Eq. (7). The corner locations in Fig. 11 are marked on the curve as well as the corresponding points on the estimated curvature measurement. We clearly see that, the minima are found on the corner locations and there are no more extrema for any noise on the curvature values. We name this detector as Curve to Chord Ratio (CCR). Similar to CTAR, the number of pixels within the curve segment intersected by the chord is 7 and Gaussian smoothing $\sigma = 3$. The threshold for the corners is defined as $T_h = 0.986$. Note that, all the parameters have been determined empirically. The empirical results are shown in Section 7.2. Fig. 12 shows the detected corners (denoted as red circles) by the proposed corner detectors CTAR and CCR on two real images. Both detectors detect the same number of corners from these two images. As we can see

from Fig. 12(a), Corners w and y, which the CPDA detector fails to detect (see Fig. 2) due to its second and third weaknesses discussed in Section 4, are detected by both CTAR and CCR detectors.

6. Discussion on accuracy and complexity of the proposed detectors

In this section we discuss the strengths of the proposed detectors over the CPDA detector. Firstly, how the proposed detectors overcome the weaknesses of the CPDA detector is described. Next, we compare the complexity of the CPDA detector with our proposed detectors.

Firstly, the estimated curvature values by CTAR and CCR for the corner locations in Fig. 3(a)–(c) are proportional to the angle (Table 1). Unlike the CPDA detector, the curvature values decrease with the decreased value of the angle. Next, the CPDA detector needs to normalize the estimated curvature values. However, the normalization process later causes false corner detection. In contrast, the estimated curvature values of CTAR and CCR are already normalized. Therefore, there is less possibility of having false corner detection. Again, CTAR and CCR use only one chord to detect the corners and there is no need to multiply the other chords' value to make a combined estimation. So, there is less chance to lose the extrema for a corner location. Finally, unlike the CSS-based corner detectors [7,8,11,12] our proposed detectors do not use any higher order derivative-based curvature estimation which considers very small neighbourhood and makes the detectors sensitive to noise.

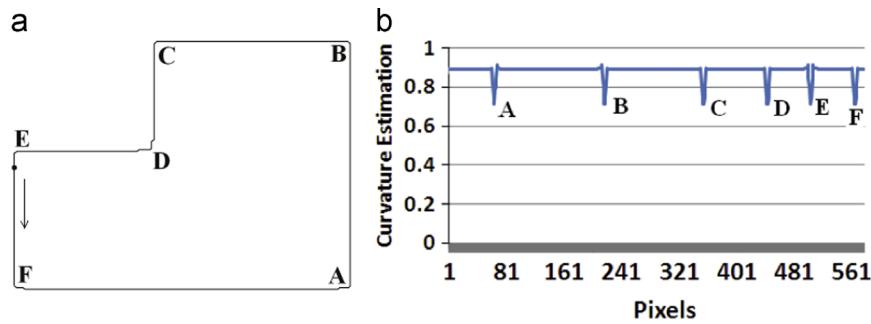


Fig. 8. Ratios representing curvatures of points on a curve having straight lines. (a) The curve of the shapes and (b) calculated ratio.

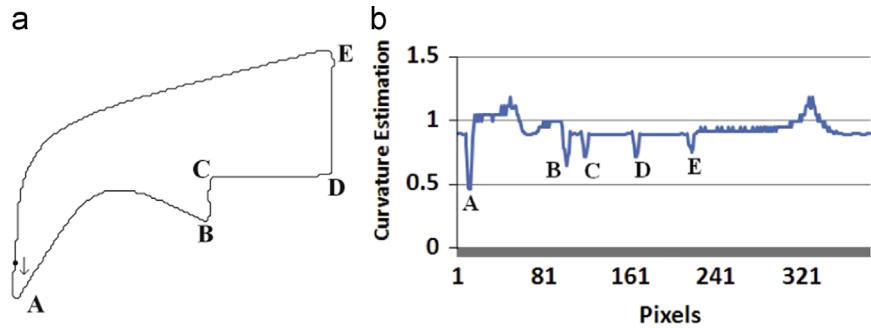


Fig. 9. Ratios representing curvatures of points on a curve. (a) The curve of the shape and (b) calculated ratio.

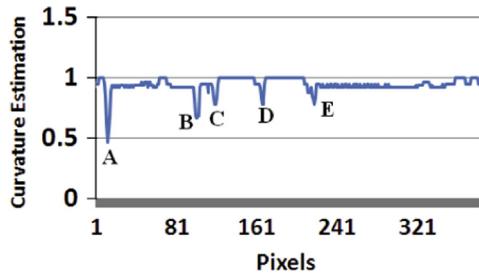


Fig. 10. Ratios representing curvatures of points on a curve without applying Gaussian smoothing.

The initial steps of all the contour-based corner detectors are generally very similar. Most of them use the Canny edge detector to extract the curves from the image and apply Gaussian smoothing to reduce the noise. So, these steps are excluded from the complexity comparison. Basically, we compare the complexity of the CPDA detector with our proposed detectors in this section.

The CPDA detector [14] utilizes the Euclidean distance to estimate the curvature. The Euclidean distance measure uses the square root operation which is computationally very expensive. CTAR and CCR also use this measure to calculate the triangular ratio and distance ratio respectively. So, we compare the complexity of these two detectors by counting the number of square root operations used in these three detectors.

Assume that curvature values on a curve with n points are estimated using three chords L_{10} , L_{20} and L_{30} . On average, the CPDA detector needs to calculate $(L_i - 2)$ square root operations (Eq. (1)) for each point of a curve. So, the total number of square root operations used by the CPDA is

$$n \times (L_{10} - 2) + n \times (L_{20} - 2) + n \times (L_{30} - 2) \quad (8)$$

In contrast, CTAR uses far fewer square root operations compared to the CPDA detector. Only $3n$ operations are required to calculate the ratio for a curve with n points and the number of pixels on the intersected curve segment does not affect the number of operations. CCR needs even fewer operations than the

CTAR detector. This detector needs $n - 1$ square root operations to calculate the distance between each consecutive location on the curve and another $n - 1$ square root operations to calculate the distance between two ends of the chord. So, the total number of operations is $2n - 2$. To illustrate the difference in the number of square root operations with a real image, the CPDA detector needs 84,582 square root operations for processing the "Lena" image (Fig. 2), whereas CTAR and CCR need only 5418 and 3672 square root operations respectively.

The CPDA detector uses a normalization process to convert all its derived curvature values to between 0 and 1. This process is already reported as a weakness of the CPDA detector (Section 4). The normalization process takes n operations [26]. On the other hand, the proposed techniques' (CTAR and CCR) ratios are already normalized. Finally, both proposed detectors do not need a second refinement step which is used in the CPDA detector since our proposed detectors are less prone to detect false corners.

7. Experimental results

This section presents the experimental setup to evaluate the performance of the proposed detectors and the results from four different experiments conducted. First, the characteristics of the parameters of the proposed corner detectors are presented. Second, the robustness of the corners detected by the proposed detectors is compared with that of the main corner detectors we have discussed in Section 2. Next, we compare our proposed corner detectors with the CPDA detector in terms of the number of detected corners. Finally, the efficiency of corner detectors is compared based on the time taken to detect corners by the detectors.

7.1. Experimental setup

The comparison measures and the experimental setup will be described in this section. The robustness of the detectors is evaluated based on the Average Repeatability [27,28] and Average Localization Error [14,28]. The repeatability measure used here is

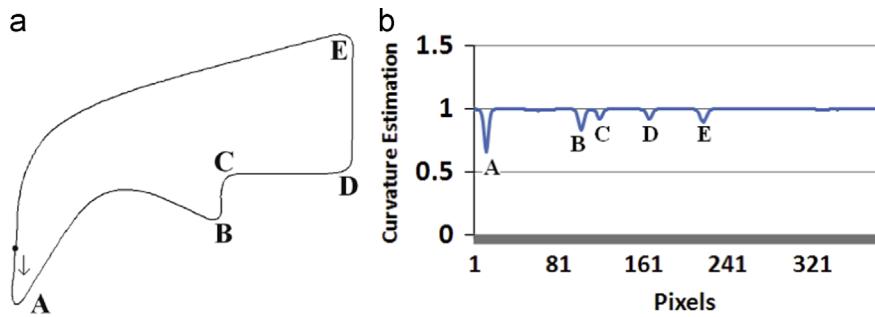


Fig. 11. Ratios representing curvatures of points on a curve after applying Gaussian smoothing. (a) The curve of the shape and (b) calculated ratio.

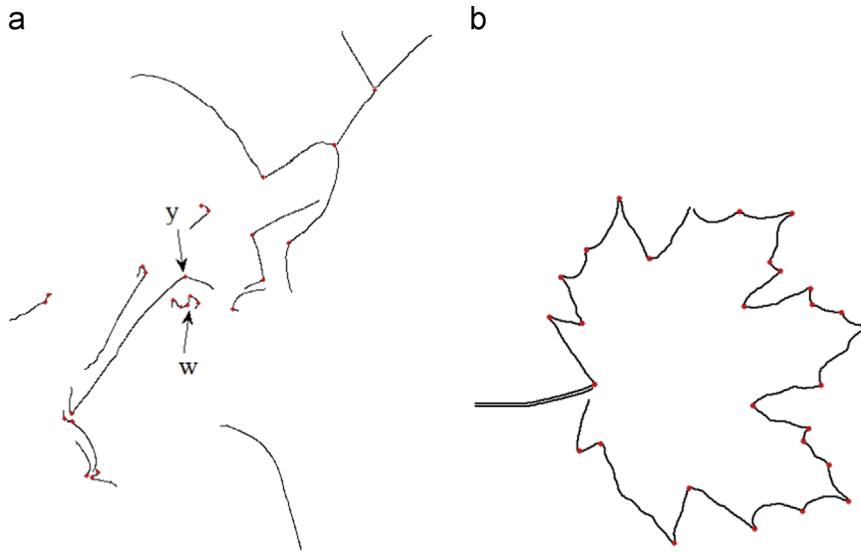


Fig. 12. The detected corners by CTAR and CCR. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

proposed by Mohanna and Mokhtarin [27]. The measure takes into consideration the number of corners from the original as well as corners from the transformed image, and it is computed as follows:

$$\text{Average Repeatability} = 100\% \times \frac{\frac{N_m}{N_o} + \frac{N_m}{N_t}}{2} \quad (9)$$

where N_o is the number of reference corners from the original image, N_t is the number of detected corners in the transformed image, and N_m is the number of matched corners between the detected and reference corners. The corners detected in the original images are used as reference points so that human intervention, which is very subjective, is not needed to determine the reference points [14].

Localization error (LE) [14] shows the distances between the detected corner locations and their reference locations on the image. It is computed as follows:

$$LE = \sqrt{\frac{1}{N_m} \sum_{i=1}^{N_m} (x_{oi} - x_{ti})^2 + (y_{oi} - y_{ti})^2} \quad (10)$$

where (x_{oi}, y_{oi}) and (x_{ti}, y_{ti}) are the i th matched corners from the corners of the reference image (N_o) and the test image (N_t) respectively. Thus a detector with a lower LE rate is better than another detector with a higher LE rate.

We have used two image datasets to evaluate the average repeatability and the average localization error of the corner detectors. The first dataset consists of over 8700 images. They

are derived by applying a wide range of geometric transformations on 23 different base images which include real and synthetic images. These base images, e.g. Lena, House, and Lab, are the benchmark images commonly used in the literature for evaluating the performance of other existing corner detectors [11,12,14]. The second dataset contains 20 base images which include a number of well-known hand-drawn shapes as well as real life images commonly used in several other researches on contour-based corner detection [23,9,10,29].

Seven different sets of geometric transformations are applied on the base images of both datasets. *Rotation*: The original images are rotated with 18 different angles $[-90^\circ, +90^\circ]$ excluding 0° ; *Uniform scale*: The original images are scaled with the interval $[0.5, 2.0]$ at 0.1 apart, excluding 1.0; *Non-uniform scaling*: The changes of scale in the x and y directions are different here. So, the images are scaled with $[0.7, 1.5]$ in the x direction and $[0.5, 1.8]$ in the y direction except when the scale changes in the x and y directions are same; *Rotation and scale*: Both transformations have been applied at the same time. The images are rotated with $[-30^\circ, +30^\circ]$ angle excluding 0° as well as the scale factor changes with $[0.8, 1.2]$ at 0.1 apart; *JPEG compression*: The images are compressed at 20 quality factors in $[5, 100]$ at 5 apart; *Gaussian noise*: The Gaussian noise with a mean 0 at 10 variances in $[0.005, 0.05]$ at 0.005 apart is applied on the images; *Shear transform*: The images are transformed with the shear factors with $[0, 0.012]$ at 0.002 apart both in the x and y directions except when the shear factors in the x and y directions are same.

7.2. Parameter settings

The parameter settings we recommend for our proposed corner detectors are stated in [Section 5](#). These are also the parameter settings we have used in our experiments presented in this paper. In this section, we discuss how these parameters of the CTAR and CCR detectors are determined.

There are three parameters for each corner detector: (1) Gaussian smoothing (σ), (2) number of pixels (i) on the curve segment intersected by the chord L_i , and (3) curvature threshold (T_h). Gaussian smoothing (σ) is used to smooth the contours of the images. Although the σ value used in most of the detectors ranges between 2 and 3, 3 is the most commonly used [23]. Therefore, this value is also chosen as the σ value for both CTAR and CCR.

The curvature threshold (T_h) and the number of pixels (i) between the two ends of the curve segment intersected by the chord are closely correlated to each other. To show such correlation, [Figs. 13 and 14](#) are used. The results in these figures are derived by varying the number of pixels (i) of chord L_i and the thresholds for CTAR and CCR respectively. [Figs. 13 and 14](#) show the average repeatability and average localization error of all the transformations described earlier applied on the first dataset. [Fig. 13\(a\)](#) and (b) shows the results of CTAR for different number of pixels (i) of chord L_i against a series of curvature thresholds. Although the average repeatability derived using chord L_9 is slightly higher than that of L_7 at a few thresholds, the average localization error of chord L_9 is not better than that of L_7 . On the other hand, the average localization error of chord L_5 is lower, however, the average repeatability of chord L_5 is not good enough. Similar scenario is found in case of CCR shown in [Fig. 14\(a\)](#) and (b).

The corner detectors using a high curvature threshold value may detect more number of corners which may not be robust. In contrast, smaller threshold value will reduce the number of detected corners. Therefore, the curvature threshold (T_h) needs to be fixed carefully so that the number of robust corners is detected as much as possible. With respect to achieving higher average repeatability with lower localization error, we can see that CTAR and CCR perform best when the curvature threshold 0.989 in [Fig. 13](#) and 0.986 in [Fig. 14](#) respectively using the chord L_7 .

7.3. Performance comparison based on repeatability and localization error

We compare the repeatability of CTAR and CCR with 10 existing compatible detectors – Poyato [17], ARCSS [11], He&Yung04 [9], He&Yung08 [10], Zhang [23], MSCP [12], EigenValues [22], DoGDetector [18], Fast CPDA [26], and CPDA [14]. Since both our proposed detectors have a single chord, we have also included a CPDA version with one chord (L_{10}) with a curvature threshold

$T_h=0.5$. We have selected chord L_{10} as it performs best among chords ($L_7, L_{10}, L_{20}, L_{30}$) which we have tested.

[Fig. 15\(a\)](#) and (b) shows the average repeatability and the average localization error of each evaluated detector respectively. Except the shear transformation, the average repeatability of CCR is the best among all the corner detectors, followed by CTAR. The average repeatability of each detector over all transformations is shown in the last column which shows CCR and CTAR are the two top performers, followed by DoGDetector and CPDA. As for localization error, the CPDA detector has the lowest average localization error, closely followed by CCR and CTAR.

The same transformations have been applied on the second image dataset. [Fig. 16\(a\)](#) and (b) shows the average repeatability and average localization error of the detectors when evaluated on this dataset. The average repeatability of the detectors on this dataset is generally similar to the first dataset. Here, MSCP, rather than DoGDetector, has the third best average repeatability. Similar to the first dataset, CCR has the best average repeatability followed by CTAR. However, the proposed detectors achieve relatively lower localization error compared to other detectors in this dataset. In this case, CCR has the least amount of localization error followed by CTAR. Since our proposed detectors use only one chord, instead of three in CPDA, we would also like to highlight that the average repeatability and average localization error of the single-chord CPDA is not better than the original CPDA in any of the datasets.

7.4. Performance comparison based on the robustness and quantity of corners detected

The repeatability of the corner detectors indicates the robustness of the detected corners. That means that the corner detectors can detect the reference corners in the test images regardless of any geometric image transformations. However, the repeatability alone is not enough to show the strength of a corner detector. The number of corners is also important. Therefore in this section, we compare the number of prominent corners detected by our proposed detectors to those detected by the CPDA detector. These corners are detected from nine different shapes from the second dataset we used in [Section 7.3](#). [Fig. 17](#) shows the four shapes which have different corner locations detected by the CPDA detector as well as the ones by the CTAR and CCR detectors. Note that, CTAR and CCR detect the same locations from all the nine shapes. The false corners are marked with plus signs (+) and the additional corners detected by our proposed detectors have been marked with star signs (*). [Fig. 17](#) shows that the CPDA detector detects a false corner in the first shape and misses many strong corners in the other three shapes. However, the proposed detectors detect most of the strong corners (14 more than CPDA) on the shapes. These results illustrate that compared to the CPDA detector, the

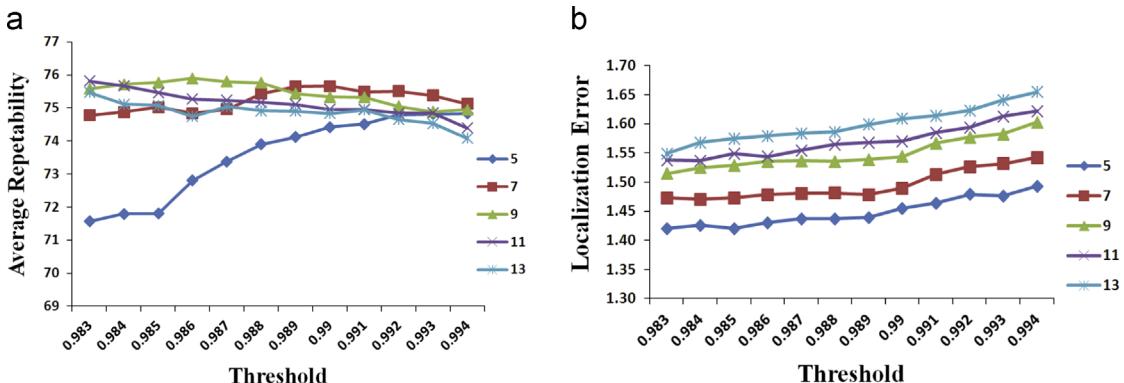


Fig. 13. (a) Average repeatability and (b) average localization error of CTAR by varying the threshold for different number of pixels between the two ends of the curve segment intersected by the chord.

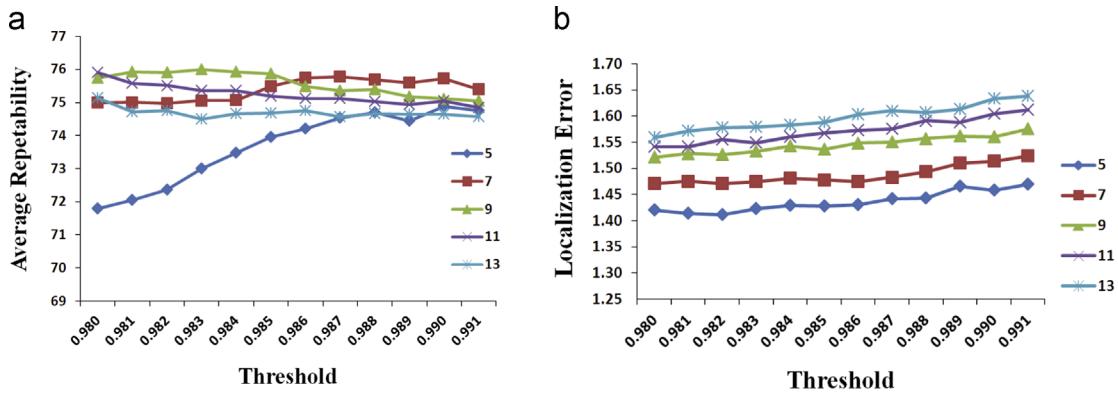


Fig. 14. (a) Average repeatability and (b) average localization error of CCR by varying the threshold for different number of pixels between the two ends of the curve segment intersected by the chord.

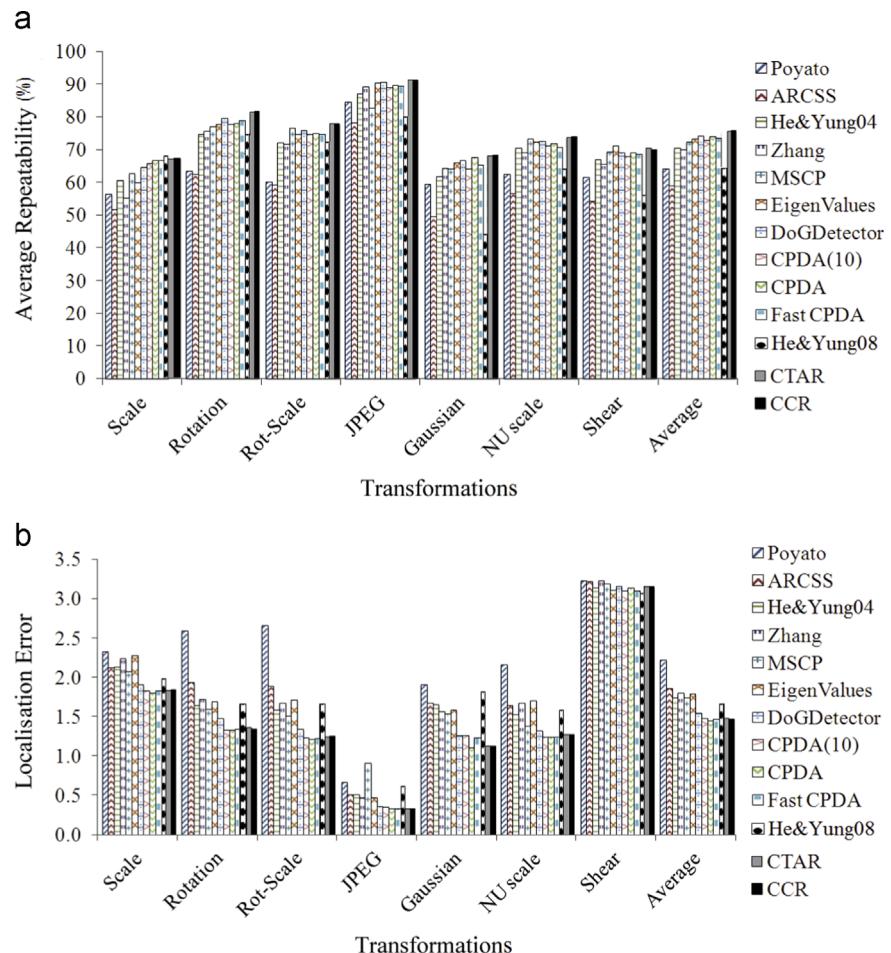


Fig. 15. (a) Repeatability by the corner detectors for different geometric transformation and (b) localization error by the corner detectors for different geometric transformation on the CPDA dataset [14].

proposed detectors not only detect corners which are more robust, they also detect higher number of corners which can be important for many computer vision applications, e.g. image registration and motion tracking.

7.5. Efficiency comparison

Table 2 shows every evaluated detector's total computational time for detecting the corners in the 23 base images from the first

dataset. All detectors are implemented in Matlab and executed in a Windows machine with a Core2Duo 2.0 GHz processor and 3 GB RAM. The time for curve extraction of every detector is the same, so this time is excluded from the time presented. Table 2 clearly shows that our proposed corner detectors are substantially more efficient than the other detectors. CCR is the fastest corner detector followed by CTAR. The Poyato detector is the slowest corner detector among all the corner detectors as it has to repeatedly calculate the geometric relationships of the candidate corners on

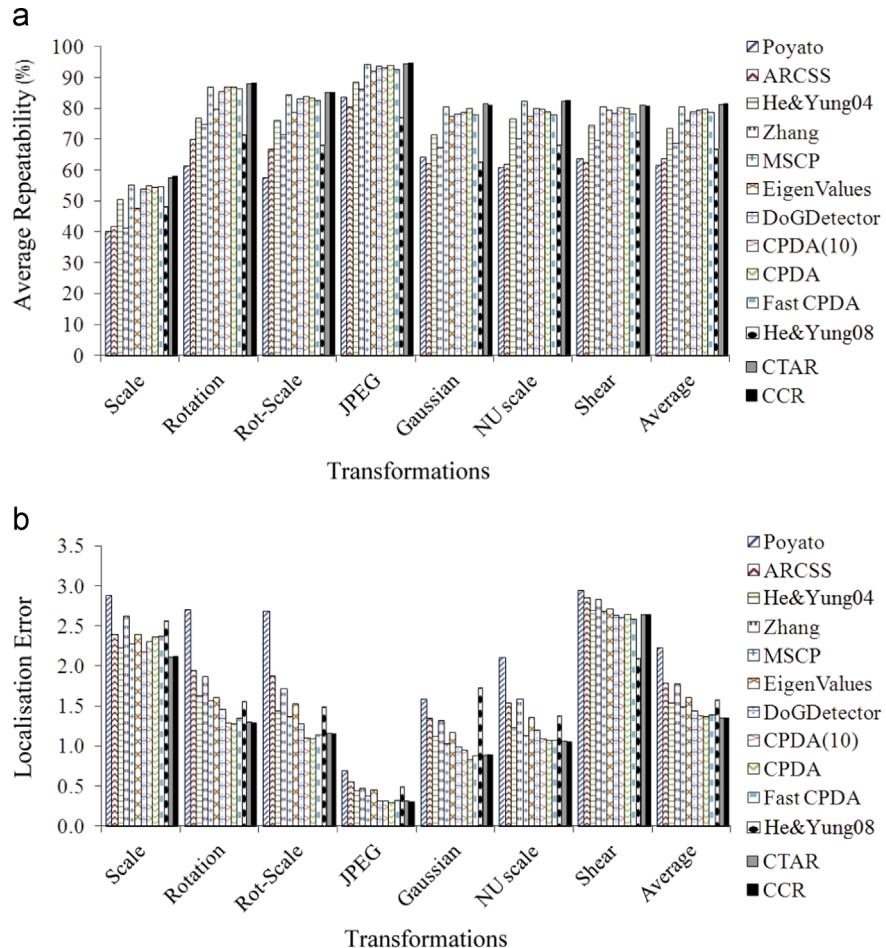


Fig. 16. (a) Repeatability by the corner detectors for different geometric transformation (b) Localization Error by the corner detectors for different geometric transformation on our proposed dataset.

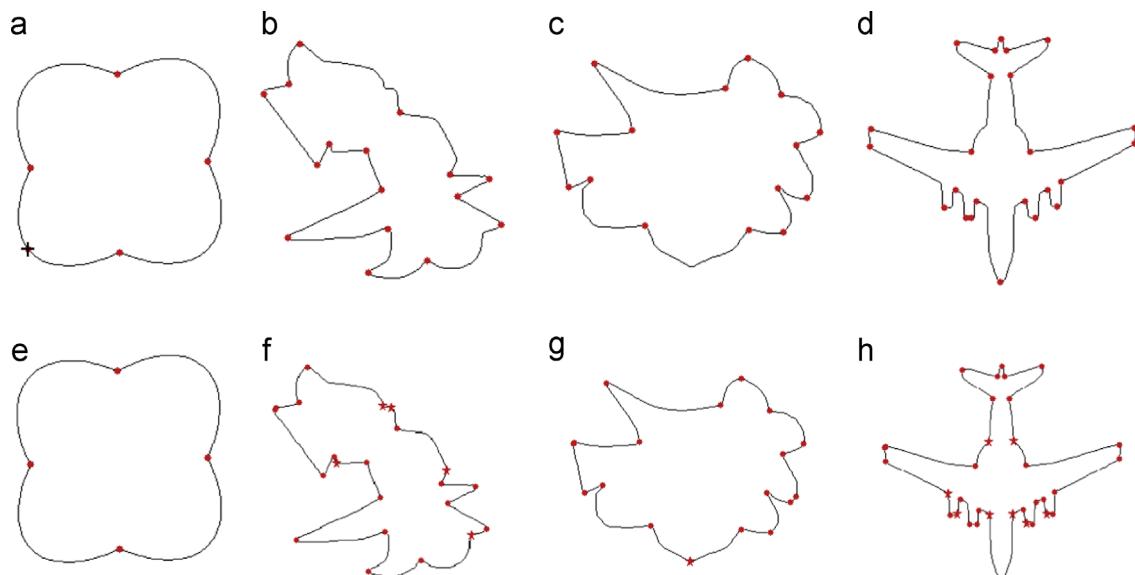


Fig. 17. (a)–(d) Corners detected by the CPDA detector; (e)–(h) Corners detected by CTAR detector; same locations (e)–(h) are detected by CCR detector [+ corresponds to a false corner, * to addition corners detected by CCR and CTAR, rest of them are common corners detected by the corner detectors].

the curve in order to assess which set of strong corners can still represent the general shape of the curve. CCR and CTAR are more than 8.5 times and 7 times respectively faster than the CPDA [14] detector, and 2.6 times and 2.13 times respectively faster than the Fast CPDA [26].

8. Conclusions

In this paper, we have proposed two effective and efficient contour-based corner detectors. Compared to the CPDA-based and eight other existing corner detectors, the proposed detectors

Table 2
Total time to detect corners in 23 images by different corner detectors.

Detector	Total execution time (s)
CCR	0.0701
CTAR	0.0857
Zhang	0.1395
Fast CPDA	0.1825
MSCP	0.2509
CPDA(10)	0.3095
He&Yung04	0.4397
CPDA	0.5965
DoGDetector	0.6152
ARCSS	0.6558
He&Yung08	0.8024
EigenValues	1.7414
Poyato	4.3760

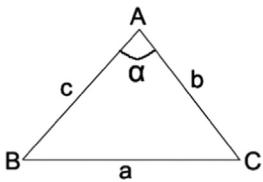


Fig. 18. Justification of CTAR threshold.

achieve the best average repeatability in detecting robust corners. Their average localization error is also comparable to the CPDA corner detector. Our proposed detectors are also substantially more efficient than other evaluated detectors.

Conflict of interest

None declared.

Acknowledgements

We would like to thank Mohammad Awrangjeb – the author of [14], for sharing his source code and the dataset.

Appendix A. Justification of CTAR threshold

CTAR takes the midpoint of the curve segment as a candidate location. Therefore, we can say in Fig. 18

$$b \approx c \quad (11)$$

The threshold of CTAR is 0.989. So,

$$\frac{a}{b+c} = 0.989 \quad (12)$$

$$\Rightarrow \frac{a}{2b} = 0.989 \quad (13)$$

$$\Rightarrow \frac{a}{b} = 1.978 \quad (14)$$

Now, in Fig. 18

$$\alpha = \cos^{-1} \left(\frac{b^2 + c^2 - a^2}{2bc} \right) \quad (15)$$

$$\alpha = \cos^{-1} \left(\frac{2b^2 - a^2}{2bc} \right) \quad (16)$$

$$\alpha = \cos^{-1} \left(1 - \frac{a^2}{2b^2} \right) \quad (17)$$

$$\alpha = 162.98^\circ \quad (18)$$

References

- [1] F.-H. Cheng, W.-H. Hsu, Parallel algorithm for corner finding on digital curves, *Pattern Recognit. Lett.* 8 (1) (1988) 47–53.
- [2] F. Mokhtarian, F. Mohanna, Performance evaluation of corner detectors using consistency and accuracy measures, *Comput. Vis. Image Underst.* 102 (April (1)) (2006) 81–94.
- [3] H. Moravec, Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, no. CMU-RI-TR-80-03, September 1980.
- [4] C. Harris, M. Stephens, A combined corner and edge detection, in: Proceedings of the Fourth Alvey Vision Conference, 1988, pp. 147–151.
- [5] S.M. Smith, J.M. Brady, Susan: a new approach to low level image processing, *Int. J. Comput. Vis.* 23 (1) (1997) 45–78.
- [6] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision, 2006, pp. 430–443.
- [7] F. Mokhtarian, R. Suomela, Robust image corner detection through curvature scale space, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (12) (1998) 1376–1381.
- [8] F. Mokhtarian, F. Mohanna, Enhancing the curvature scale space corner detector, in: SCIA01, 2001, pp. O-M4B.
- [9] X.C. He, N.H.C. Yung, Curvature scale space corner detector with adaptive threshold and dynamic region of support, in: Proceedings of 17th International Conference on Pattern Recognition (ICPR'04), vol. 2, 2004.
- [10] X. He, N. Yung, Corner detector based on global and local curvature properties, *Opt. Eng.* 47 (5) (2008) 057008-1–057008-12.
- [11] M. Awrangjeb, G. Lu, M. Mursheed, An affine resilient curvature scale-space corner detector, in: ICASSP 2007, vol. 1, pp. I-1233–I-1236, April 2007.
- [12] X. Zhang, M. Lei, D. Yang, Y. Wang, L. Ma, Multi-scale curvature product for robust image corner detection in curvature scale space, *Pattern Recognit. Lett.* 28 (5) (2007) 545–554.
- [13] J. Han, T. Poston, Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature, *Pattern Recognit. Lett.* 22 (10) (2001) 1133–1144.
- [14] M. Awrangjeb, G. Lu, Robust image corner detection based on the chord-to-point distance accumulation technique, *IEEE Trans. Multimedia* 10 (6) (2008) 1059–1072.
- [15] N. Ansari, K. wei Huang, Non-parametric dominant point detection, *Pattern Recognit.* 24 (9) (1991) 849–862 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/003120391900040>.
- [16] C.-H. Teh, R. Chin, On the detection of dominant points on digital curves, *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (August (8)) (1989) 859–872.
- [17] R.M.-C.A. Carmona-Poyato, N.L. Fernandez-Garcia, F. Madrid-Cuevas, Dominant point detection: a new proposal, *Image Vis. Comput.* 23 (13) (2005) 1226–1236 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0262885605001332>.
- [18] X. Zhang, H. Wang, M. Hong, L. Xu, D. Yang, B.C. Lovell, Robust image corner detection based on scale evolution difference of planar curves, *Pattern Recognit. Lett.* 30 (March) (2009) 449–455.
- [19] J.S. Lee, Y. Sun, C. Chen, Multiscale corner detection by using wavelet transform, *IEEE Trans. Image Process.* 4 (January (1)) (1995) 100–104.
- [20] A. Quddus, M. Gabouj, Wavelet-based corner detection technique using optimal scale, *Pattern Recognit. Lett.* 23 (January) (2002) 215–220.
- [21] X. Gao, F. Sattar, A. Quddus, R. Venkateswarlu, Multiscale contour corner detection based on local natural scale and wavelet transform, *Image Vis. Comput.* 25 (6) (2007) 890–898.
- [22] D.M. Tsai, H.T. Hou, H.J. Su, Boundary-based corner detection using eigenvalues of covariance matrices, *Pattern Recognit. Lett.* 20 (1999) 31–40.
- [23] X. Zhang, H. Wang, A.W.B. Smith, X. Ling, B.C. Lovell, D. Yang, Corner detection based on gradient correlation matrices of planar curves, *Pattern Recognit.* 43 (April) (2010) 1207–1223.
- [24] M. Awrangjeb, G. Lu, C.S. Fraser, Performance comparisons of contour-based corner detectors, *IEEE Trans. Image Process.* 21 (September (9)) (2012) 4167–4179.
- [25] J. David, C. Munson, A note on lena, *IEEE Trans. Image Process.* 5 (January (1)) (1996).
- [26] M. Awrangjeb, G. Lu, C.S. Fraser, M. Ravanbakhsh, A fast corner detector based on the chord-to-point distance accumulation technique, in: Proceedings of the 2009 Digital Image Computing: Techniques and Applications, DICTA '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 519–525.
- [27] F. Mohanna, F. Mokhtarian, Performance evaluation of corner detection algorithms under similarity and affine transforms, in: British Machine Vision Conference, 2001, pp. 353–362.
- [28] M. Awrangjeb, G. Lu, An improved curvature scale-space corner detector and a robust corner matching approach for transformed image identification, *IEEE Trans. Image Process.* 17 (December (12)) (2008) 2425–2441.
- [29] D. Chetverikov, A simple and efficient algorithm for detection of high curvature points in planar curves, in: N. Petkov, M. Westenberg (Eds.), *Computer Analysis of Images and Patterns, Lecture Notes in Computer Science*, vol. 2756, Springer, Berlin, Heidelberg, 2003, pp. 746–753.

Shyh Wei Teng received his Ph.D. degree from Monash University. He is currently a Senior Lecturer at Faculty of Science and Technology, Federation University Australia and his research interests include data mining, machine learning techniques, content-based image retrieval and image registration.

Rafi Md. Najmus Sadat obtained his Ph.D. degree from Monash University, Australia. His research topics are image registration and image retrieval.

Guojun Lu is currently a Professor at Faculty of Science and Technology, Federation University Australia. He has held positions at Loughborough University, National University of Singapore, Deakin University and Monash University, after he obtained his Ph.D., in 1990, from Loughborough University and B.Eng., in 1984, from Nanjing Institute of Technology (now South East University). His main research interests are in multimedia communications and multimedia information indexing and retrieval. He has published over 140 refereed journal and conference papers and two books in these areas.