Corner detection by means of contour local vectors

P. Reche, C. Urdiales, A. Bandera, C. Trazegnies and F. Sandoval

This paper proposes a corner detection algorithm for planar shapes based on a new curvature index. The calculation parameters of this new index are adapted to the shape curvature at each point of its contour. The process consists of estimating the maximum length of contour yielding no significant discontinuities on the right and left sides of each contour point in order to calculate the contour curvature index at that point. Thus, the curvature index is adaptively filtered depending on the natural scale of the contour pixels. Corners are detected by simply thresholding the curve. The proposed method grants a more precise characterisation of the contour. Detected corners are very stable against noise distortion, scaling or rotation.

Introduction: Since most corner detectors based on curvature estimation implicitly or explicitly filter the contour descriptor at a fixed cut frequency before detection, only corners not affected by such a filtering are detected [1]. In order to solve this problem, detection might be performed in an iterative way by changing the detection parameters at each iteration; in this case, corners are those relevant points that remain after several iterations [2]. A second choice is to filter contour descriptors in an adaptive way depending on the local nature of the contour [6].
In this paper, we propose a more stable adaptive corner detection method whose main novelty is that the cut frequency of the filter required to build the curvature function is adapted to the contour features and, therefore, to corner scale. Thus, corners can be unsupervisedly detected over a wide range of scales.

*Multiscale corner detection algorithm:* For each pixel \( i \) of a contour, the proposed detection algorithm consists of the following steps:

1) Calculation of the maximum length of contour presenting no discontinuities on the right and left sides of the working pixel \( i \): \( k_f[i] \) and \( k_b[i] \), respectively. \( k_f[i] \) is calculated by comparing the euclidean distance from pixel \( i \) to its \( k_f[i] \)th neighbour \( d(i,i+k_f[i]) \) to the real length of the contour between both pixels \( l(i,i+k_f[i]) \). Both distances tend to be equal in absence of corners, even if contours are noisy. Otherwise, \( d(i,i+k_f[i]) \) is quite shorter than \( l(i,i+k_f[i]) \). Thus, \( k_f[i] \) is the largest value that satisfies:

\[
\frac{l(i,i+k_f[i]) - d(i,i+k_f[i])}{l(i,i+k_f[i])} < N_A
\]

being \( N_A \) a constant value that depends on the noise level tolerated by the detector. \( k_b[i] \) is also set according to Eq. (1), but using \( i-k_b[i] \) instead of \( i+k_f[i] \).

If \( N_A \) is large, \( k_f[i] \) and \( k_b[i] \) tend to be large and some corners might be lost, but if it is low, the resulting curvature function is very noisy and false detections occur. Fortunately, it is very easy to choose a suitable
and it has been empirically proven that $N_A=0.1$ works correctly in most cases. Fig. 1 presents an example of how $k_f[i]$ and $k_b[i]$ are estimated for one random pixel of the contour.

2) Calculation of the local vectors $f_i$ and $b_i$ associated to each pixel $i$.

These new vectors present the variation in $X$ and $Y$ between pixels $i$ and $i+k_f[i]$, and between pixels $i$ and $i-k_b[i]$. If $(x_i,y_i)$ are the coordinates of the pixel $i$, the local vectors associated to $i$ are defined as $\vec{f}_i = (x_i + k_f[i] - x_i, y_i + k_f[i] - y_i) = (f_x[i], f_y[i])$ and $\vec{b}_i = (x_i - k_b[i] - x_i, y_i - k_b[i] - y_i) = (b_x[i], b_y[i])$.

3) Calculation of the angle associated to each pixel $i$ of the contour.

According to the works of Rosenfeld and Johnston [6], the angle at pixel $i$ can be easily calculated by using the equation:

$$|\kappa_b[i]| = \frac{1}{2} \cdot (1 + \cos \theta) = \frac{1}{2} \cdot \left(1 + \frac{\vec{f}_i \cdot \vec{b}_i}{|\vec{f}_i| \cdot |\vec{b}_i|}\right)$$

(2)

With the above procedure, the obtained curvature function represents in an absolute manner the curvature associated to each pixel on the contour, with no information on concavity or convexity. This information, irrelevant for corner detection, could be of paramount importance for object recognition. In order to include concavity or convexity information, a modified curvature function is constructed as follows:

$$\kappa_b[i] = \text{Sign}_i \cdot |\kappa_b[i]|$$

(3)

where $|\kappa_b[i]|$ refers to the previous curvature index defined in (2). The value of $\text{Sign}_i$ is obtained by using the local vectors $\vec{f}_i$ and $\vec{b}_i$:
If $\text{Sign}_i$ is negative, the curvature is concave, but if $\text{Sign}_i$ is positive, the curvature is convex. Fig. 2 shows an example of these two cases.

4) Cornerity weighting. To improve the definition of $\kappa_\theta[i]$, the curvature at each of the contour points is weighted by a factor which depends on the product of the lengths of the uniform sections to either side of the point (forward and backward arms). Thus, the resulting curvature index, $\kappa_\theta[i]^*$, can be obtained as follows

$$
\kappa_\theta[i]^* = \frac{\sqrt{\text{Ln}(k_t[i]) \cdot \text{Ln}(k_b[i])}}{\text{Ln}(N)} \cdot \text{Sign}_i \left(1 + \frac{\tilde{f}_i \cdot \tilde{b}_i}{|\tilde{f}_i| |\tilde{b}_i|}\right)
$$

$N$ being the length of the object contour.

5) Detection of corners over $\kappa_\theta[i]^*$. Corners are those points which: i) are local peaks of the function; and ii) are over the minimum angle required to be considered a corner instead of a spurious peak due to remaining noise.

Results: Previous works [3,6] have studied the behaviour of several corner detectors [2-6] and concluded that the circular histograms method (CHM) [3] and adaptive method (AM) [6] were the best choices concerning computational times and stability of detected corners. Therefore, the proposed method is compared to these methods to test its performance. Fig. 3 presents the contour of an object at two different scales and
orientations. Corners detected by the AM, CHM and the proposed method are superimposed on the contours. It can be appreciated in Figs. 3.b-3.d that if the contour shape is significantly affected by scale and noise, the CHM (Fig. 3.b) and AM (Fig. 3.c) lose stability. Table I shows the behaviour of different methods in terms of computational load and precision for the objects in Figs. 3.a-d according to the figure of merit (FOM). The FOM is the relationship between the compression ratio (CR) and the integral square error (ISE). CR is the ratio between the total number of contour points and the number of detected corners on it. ISE is the error of the polygonal approximation built with the detected corners when compared to the real contour. It must be noted that the main advantage of the proposed method is that its FOM is high and it is very resistant to scale and noise.

Conclusions: This paper proposes a new algorithm to estimate the curvature of a contour in an adaptive way so that corners can be detected even if they appear at different natural scales. Detection is very stable against scale and noise distortions.

Acknowledgments: This work has been partially granted by the Spanish Ministerio de Ciencia y Tecnología (MCYT), under Project No. TIC2001-1758.
References

1 ROSIN, P.L.: “Representing curves at their natural scales”, *Pattern Recognition*, 1993, 25 (11), pp. 1315-1325


Authors’ affiliations:

P. Reche, C. Urdiales, A. Bandera, C. Trazegnies and F. Sandoval

Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga, Campus de Teatinos - 29071, Málaga (Spain)

Corresponding author e-mail: bandera@dte.uma.es
Figure captions and subcaptions

**Fig. 1.** Calculation of $k_f[\mathbf{I}]$ and $k_b[\mathbf{I}]$: a) Contour of the object and point $\mathbf{P}$; b) $k_f[\mathbf{P}]$; and c) $k_b[\mathbf{P}]$.

**Fig. 2.** Calculation of $\text{Sign}_i$: a) concave angle and b) convex angle.

**Fig. 3.** a) Contour #1 and corners detected by using CHM (●), AM ( ) and the proposed method (O); b) contour #2 (x4) and corner detected by CHM (●); c) contour #2 (x4) and corner detected by AM ( ); and d) contour #2 (x4) and corner detected by the proposed method (O).

**Table I.** Performance of CHM, AM and the proposed method for Figs. 3.a-b.
Fig. 1
Fig. 2

(a) 

(b)
Fig. 3

(a)

(b)

(c)

(d)
<table>
<thead>
<tr>
<th>Method</th>
<th>Contour #1</th>
<th>Contour #2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FOM</td>
<td>Time (ms)</td>
</tr>
<tr>
<td>CHM (Figs. 3.a/3.b)</td>
<td>0.200</td>
<td>7.0</td>
</tr>
<tr>
<td>Adaptive Method (Figs. 3.a/3.c)</td>
<td>0.187</td>
<td>5.0</td>
</tr>
<tr>
<td>Proposed method (Figs. 3.a/3.d)</td>
<td>0.195</td>
<td>4.5</td>
</tr>
</tbody>
</table>