HAL
open science

# An improved method for high-curvature detection with applications to automated inspection

José-Cruz Pineda, Radu Horaud

## To cite this version:

José-Cruz Pineda, Radu Horaud. An improved method for high-curvature detection with applications to automated inspection. Signal Processing, 1983, 5 (2), pp.117-125. 10.1016/0165-1684(83)90018-X . hal-01119236

**HAL Id: hal-01119236**

**https://inria.hal.science/hal-01119236v1**

Submitted on 25 Feb 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Signal Processing 5 (1983) 117–125
North-Holland Publishing Company



# AN IMPROVED METHOD FOR HIGH-CURVATURE DETECTION WITH APPLICATIONS TO AUTOMATED INSPECTION

José-Cruz PINEDA and Patrice HORAUD
*Laboratoire d'Automatique de Grenoble, B.P. 46, 38402 Saint-Martin-d'Hères, France*

Received 16 November 1981
Revised 24 May 1982

**Abstract.** Digital curve acquisition with a vision system introduces a large amount of degradation. At the same time 'corners' convey important visual information. Unfortunately they are found in the same spectral domain as the quantization noise. Low-pass filtering of the curvature function is compared with more sophisticated methods that keep the angles 'sharp'. An automated inspection is successfully performed, demonstrating the efficiency of a new method.

**Zuzammenfassung.** Die Erfassung von Kurvenzügen mit einem digitalen Bildverarbeitungssystem führt zu einer erheblichen Verschlechterung der Qualität der Bildwiedergabe. 'Ecken' im Kurvenverlauf enthalten wichtige visuelle Information, werden aber bei spektraler Verarbeitung in den gleichen Spektralbereich abgebildet wie das Quantisierungsrauschen. Im vorliegenden Beitrag wird die Methode der Tiefpaßfilterung des Kurvenzugs mit anspruchsvolleren Verfahren verglichen, die die Ecken 'scharf' lassen. Ein neues Verfahren wird vorgeschlagen, das den Kurvenzug automatisch auf Ecken absucht, und seine Wirkungsweise wird anhand eines Beispiels beschrieben.

**Résumé.** La discrétisation d'une courbe à l'aide d'un système de vision, introduit une dégradation dû au bruit de quantification. Or les 'angles' contiennent une information visuelle importante. Malheureusement ils sont placés dans le même domaine spectral que le bruit de discrétisation. Le filtrage passe-bas de la fonction de courbure est comparé avec d'autres méthodes plus sophistiqués, pouvant détecter les angles 'aigus'. Une expérience en inspection automatique a été effectuée avec succès, démontrant l'efficacité de la méthode proposée.

**Keywords.** Angle detection, automated inspection, contour description.

## 1. Introduction

The information conveyed by a two dimensional shape is concentrated on its boundary [1]. The digital processing of this data depends on how the pattern has been degraded during acquisition. The most important sources of noise are:

(1) the optical system and the lighting,
(2) the electronic device and
(3) the image quantization.

One possible representation of a plane curve is the curvature $\mathscr{C}$ as a function of the arc length $(S)$ [2]. But in practice we measure $\mathscr{C}_N(S)$ which is an altered version of the curvature. In order to extract noise-free features of a contour for recognition tasks it is necessary to perform filtering:

$$\mathscr{C}(S) = H(S) * \mathscr{C}_N(S) \tag{0}$$

The impulse response of this filter $H(S)$ must be chosen differently for every contour tracing algorithm, since this process introduces noise specific to each method.

In this paper, we will first present a low-pass filter. Similar results are obtained using an algorithm derived from [4] and [5]. This last method is then compared with the filtering process.

0165-1684/83/0000–0000/$03.00 © 1983 North-Holland

The shape can be recognized from the curvature maxima.

To demonstrate the efficiency of the method, accurate automated inspection is then performed on a class of flat mechanical parts. Reference [6] is an example of the utility of the curvature to discriminate three dimensional shapes.

## 2. Curvature function

### 2.1. The continuous case and the discrete case

Let $(C)$ be a closed oriented curve of the plane. The position of a point $M$ on this curve is expressed by the length $S$ of the arc $M_0M$, where $M_0$ is an origin point, $S$ is commonly called curvilinear abcissa (Fig. 1). Such a curve can be represented by a complex function $Z(S)$:

$$Z(S) = X(S) + \mathrm{i}\, Y(S). \tag{1}$$

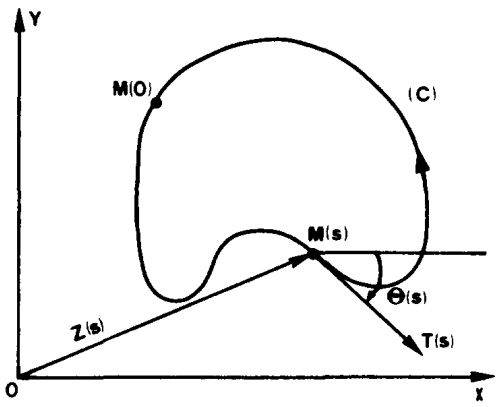


Fig. 1. Curvature computation of a closed contour.

The tangent function is given by the derivative of $Z$:

$$T(S) = \dot{X}(S) + \mathrm{i}\, \dot{Y}(S). \tag{2}$$

The tangent angle with the $x$-axis is a real function:

$$\theta(S) = \mathrm{Arctg}(\dot{Y}(S)/\dot{X}(S)). \tag{3}$$

Finally the curvature is the variation of $\theta(S)$ and is a real function as well:

$$\mathscr{C}(S) = \dot{\theta}(S). \tag{4}$$

In the discrete case $(C)$ is determined by a set of points $M_0, M_1, \ldots, M_n, M_{n+1}$ $(M_0 = M_{n+1})$ of arc lengths $S_0\,(=0), \ldots, S_k, \ldots S_{n+1}$. The equivalent equations for curvature extraction are (same expressions as in [2]):

$$Z(S_k) = X(S_k) + \mathrm{i}\, Y(S_k), \tag{5}$$

$$T(S_k) = [X(S_k) - X(S_{k-1})] + \mathrm{i}[Y(S_k) - Y(S_{k-1})], \tag{6}$$

$$\theta(S_k) = \mathrm{Arctg}\left[\frac{Y(S_k) - Y(S_{k-1})}{X(S_k) - X(S_{k-1})}\right], \tag{7}$$

$$\mathscr{C}(S_k) = \theta(S_k) - \theta(S_{k-1}). \tag{8}$$

### 2.2. Practical computation of the curvature

The real curves are obtained from a binary image using a border following algorithm. For a rectangular grid every point has eight neighbors in eight uniformly distributed directions coded $0, 1, 2, \ldots, 7$. This allows direct estimation of $\theta(S_k)$ taking discrete values which are multiples of 45°. The curvature function will be affected by the sampling noise; actually we compute $\mathscr{C}_N(S_k)$.

Our aim e.g., is to extract meaningful information about the nature of the shape (regions of maximum curvature are significant) but this noisy curve does not allow it without further processing. One possibility is to smooth it. More practical solutions from the computational point of view will be discussed in the next section.

### 2.3. Low-pass filtering

A simple method for smoothing the step function $\mathscr{C}_N(S_k)$ is to pass it through a low-pass filter of a specified bandwidth. Such a process will eliminate undesirable high frequencies. Although this looks very attractive, there are some problems. It has been shown in [2] that the non-linear process of curvature determination amplifies the high-frequency quantization noise and distributes this amplified noise over the entire spectral range of the resulting one dimensional function. Unfortunately sharp corners are also in the high-frequency domain. Smoothing them may distort very useful

information about the shape, since one of the goals is to reconstruct the shape ony from high curvature regions.

Let us consider the contour in Fig. 2(a). The curvature function that was computed using the chain code (Fig. 2(b)) has been low-pass filtered. In order to do that we worked in the spectral domain using a FFT algorithm. With a cut-off frequency of 0.2 cycles per length unity we produced the function displayed in Fig. 2(c). It is an easy task now to count the number of curvature peaks or to measure the spacing between those peaks, performed in the 'world space' instead of the 'camera space' using a calibration program. This avoids gross errors when the object is rotated. Indeed, for a given shape, the number of its boundary pixels can vary as much as 50%. This variation is a function of the orientation of the shape with respect to the direction of the camera scanning. The contour in Fig. 2(d) has been reconstituted after filtering. While the straight-lines have been nicely smoothed, the corners have lost their sharpness which is an undesirable effect.

The filtered curvature $\mathscr{C}(S_k)$ has been produced here by low-pass filtering, the cutoff frequency being determined by spectral analysis. This result is not satisfactory enough for accurate recognition of the angles as mentioned above. Meanwhile it can be useful to evaluate other methods for extracting curvature information without altering the contour shape.

## 3. Angle detection algorithms

### 3.1. The basic method

We found out that one of the angle detection schemes the most similar to a smoothing operation was proposed in [4]. The object border is described by a chain-code but instead of recording the direction changes of these small elements, a moving line segment scans the chain, connecting the end points of a sequence of $l$ links. The angular differences between successive segment positions are used as a measure of curvature.

Let a contour be described by the chain:

$$A^n = a_1, a_2, \ldots, a_i, \ldots, a_n;$$

$a_i$ links points $M_i$ and $M_{i+1}$. Every code can take a value between 0 and 7. $L_j^l$ is the line segment connecting $l$ elements and pointing on $a_j$. The projections of $a_j$ on the $x$- and $y$-axis are $a_{jx}$ and $a_{jy}$.

$$a_{jx}, a_{jy} \in \{-1, 0, 1\}.$$

The $x$ and $y$ components of $L_j^l$ will be:

$$X_j^l = \sum_{i=j-l+1}^{j} a_{ix}, \qquad Y_j^l = \sum_{i=j-l+1}^{j} a_{iy}. \tag{9}$$

The length and the slope of $L_j^l$ viewed as a vector will be:

$$d_j^l = [(X_j^l)^2 + (Y_j^l)^2]^{1/2}, \tag{10}$$

$$\begin{aligned} \theta_j^l &= \operatorname{arctg}(Y_j^l/X_j^l) \quad \text{if } |X_j^l| \geq |Y_j^l|, \\ &= \operatorname{arccot}(X_j^l/Y_j^l) \quad \text{if } |X_j^l| < |Y_j^l|. \end{aligned} \tag{11}$$

The variations of $\theta_j^l$ express the curvature. One possibility to calculate it is to consider the following expression, the same as in [4]. (see Fig. 3(a)):

$$\delta_j^l = \theta_{j+1}^l - \theta_{j-1}^l. \tag{12}$$

Eq. (12) is a smooth measure of the curvature compared with (8). The larger $l$ is the more important the smoothing will be. The choice of $l$ will depend on the spectral distribution of the curvature-noise. Note that $\phi_j^1 = \mathscr{C}_N(S_j)$.

Let's look now at the behaviour of this new curvature function at corner locations. Figs. 3(b and c) show two possible situations. In the first case $\delta_j^l$ is composed of three different regions. For $j < B$, $\delta_j^l = 0$. Then $\delta_j^l \neq 0$ for $B \leq j \leq B + l$. For $j > B + l$, $\delta_j^l = 0$. In the second case there is no sharp corner. For the relatively smooth region $BB'$ of $m$ links $\delta_j^l$ is not zero over $m + l$ points. Hence the width of the curvature peaks includes important information.

In practice the curvature is not zero for the straight lines $AB$ and $B'C$ because of the inherent noise. This method is reliable if we want to detect angles different from those produced by the

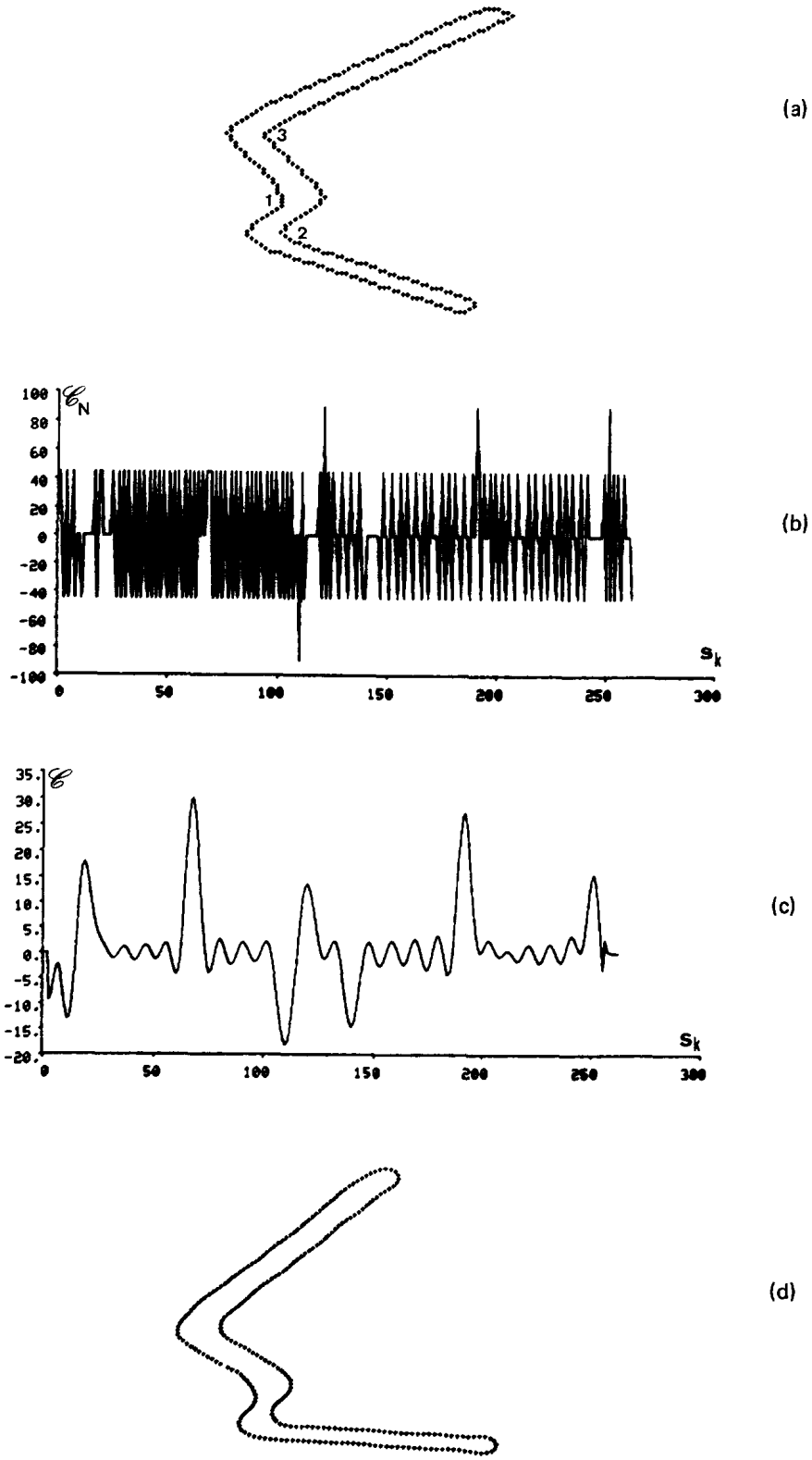


Fig. 2. Low-pass filtering: (a) initial contour, (b) noisy curvature, (c) filtered curvature, (d) smoothed contour.

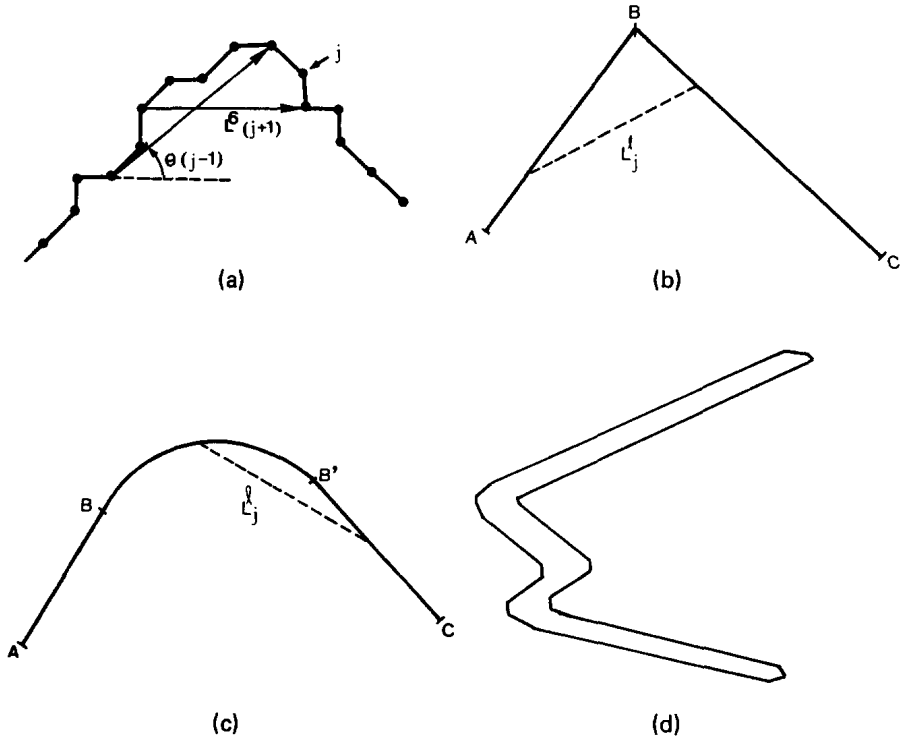


Fig. 3. Angle and curve detection: (a) computation of $\delta^l$, (b) a corner, (c) an arc, (d) reconstituted contour.

quantization effect. We can threshold $\delta^l_j$ and produce a measure of the conerity. The contour in Fig. 2(a) has been reconstituted from its eight high-curvature regions detected by the algorithm described above (Fig. 3(d)). Smooth corners were approximated by two line segments. The low-curvature regions are smoothed while the abrupt variations remain unchanged.

The number of points of a curve varies with its position in the image. This means that the contour is sampled at different rates. Algorithm's parameter $l$ depends strongly upon this rate. One concludes then that $l$ has to be related somehow to a cutoff frequency and be chosen by considering the spectral distribution of the sampling noise.

### 3.2. An improved method

A different possibility for angle detection is offered in [5]. The idea of this parallel algorithm is that noise is perceived as a local phenomenon while the corner notion is strongly attached to the context. Let $M_i = (x_i, y_i)$ be a contour point. For every $i$ two vectors are defined:

$$\begin{aligned} \vec{a}_{ik} &= (x_i - x_{i+k}, y_i - y_{i+k}), \\ \vec{b}_{ik} &= (x_i - x_{i-k}, y_i - y_{i-k}). \end{aligned} \tag{13}$$

A possible measure of the curvature at $M_i$ is the cosine of the angle between $a_{ik}$ and $b_{ik}$:

$$C_{ik} = (\vec{a}_{ik} \cdot \vec{b}_{ik})/|\vec{a}_{ik}| \cdot |\vec{b}_{ik}|. \tag{14}$$

$C_{ik}$ is large (1) when the curve is turning rapidly and small (−1) for a straight line.

The disadvantage of this method is the time consuming procedure to calculate $k$ iteratively for every curve point. If $k$ varies from $m$ to 1, there exists $h$, $1 < h \leq m$ for which:

$$C_{im} < C_{im-1} < \cdots < C_{ih} \geq C_{ih-1}.$$

For every $M_i$, $C_{ih}$ is computed. An angle is a local maximum if $C_i \geq C_j$ for all $j$ such that $|i-j| \leq \frac{1}{2}h$ which means that two corners will be confused if they are within a distance of $\frac{1}{2}h$.

### 3.3. A new approach

For this method the parameter $h$ has a similar meaning as $l$ for the previous one. This comparison suggests a new curvature function, which is a modification of eq. (12):

$$\lambda_j^l = \theta_{j+l}^l - \theta_j^l \tag{15}$$

where $\theta_j^l$ is defined in eq. (11). $\lambda_j^l$ makes more use of the context than $\delta_j^l$ (see Fig. 4(a)). This is made without computing $l$ at every point as in [5]. We noticed that $\lambda^l$ was less sensitive to noise effects than $\delta^l$, especially when the object was rotated. Fig. 4(b) shows $\lambda^5$ for the same contour. The minimum distance between two angles detected without ambiguity is $l$.

### 3.4. Optimal evaluation of 'l'

One last question must be answered to complete the method: how could one automatically estimate the parameter '$l$'? Instead of doing it 'on line' in the same way as '$h$' is computed for every point in Section 3.2. We suggest an 'off line' evaluation. We have stated in Section 3.1 that the curvature extraction performed with (12) or (15) is a smoothing function very similar to the one obtained by low-pass filtering. In addition (12) or (15) preserve the angles sharpness which is highly desired for our purposes. We know how to determine the parameters of $\mathscr{C}(S_k)$, i.e., the cutoff frequency. For every contour tracing algorithm the quantization noise has a different spectral distribution. We want to determing '$l$' in each case such that this noise is properly removed. Consider the problem of determining a value of $l$ that would be equivalent to the cutoff frequency of the filter. To compute $l$ we measure the error $E$ between $\mathscr{C}(k)$ and $\lambda_k^l$ by the following formula:

$$E = \operatorname*{Min}_j \frac{\sum_{k=1}^{n} [(\lambda_{k+j}^l - \bar{\lambda}^l) - (\mathscr{C}(k) - \bar{\mathscr{C}})]^2}{\sum_{k=1}^{n} [(\lambda_k^l - \bar{\lambda}^l)^2 + (\mathscr{C}(k) - \bar{\mathscr{C}})^2]} \tag{16}$$

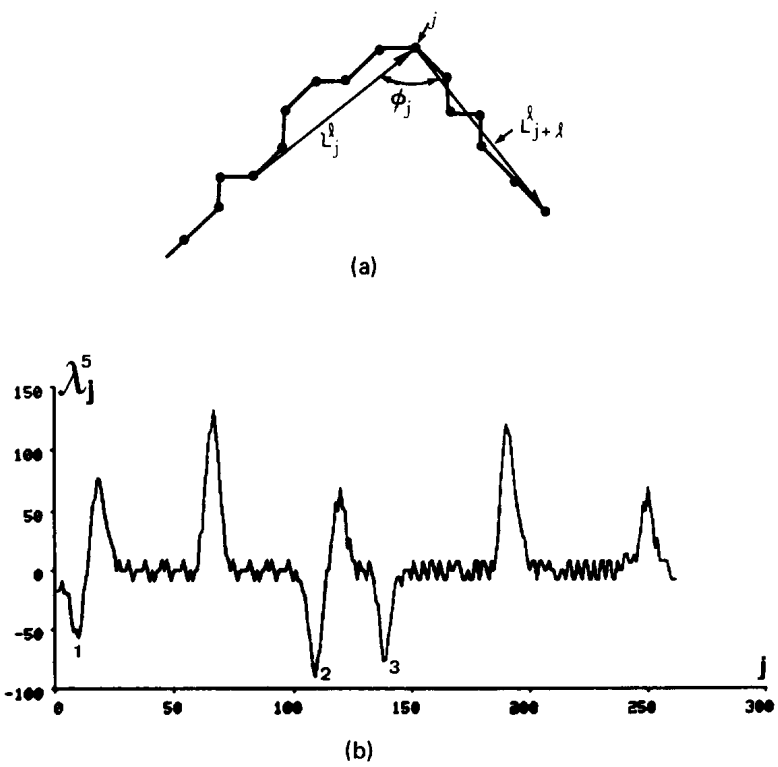


Fig. 4. Improved angle detection: (a) computation of $\lambda^l$, (b) $\lambda^l$ for the contour in Fig. 3(a).

where $j$ is a delay parameter that is varied to produce a minimum, $\bar{\mathscr{C}}$ and $\bar{\lambda}^l$ are the average values of $\mathscr{C}(k)$ and $\lambda_k^l$ over the $n$ curve points:

$$\bar{\mathscr{C}} = \frac{1}{n} \sum_{k=1}^{n} \mathscr{C}(k), \qquad \lambda^l = \frac{1}{n} \sum_{k=1}^{n} \lambda_k^l. \tag{17}$$

The mean square error $E$ is shown plotted as a function of $l$ (Fig. 5). $E$ is minimum for $l = 3$.

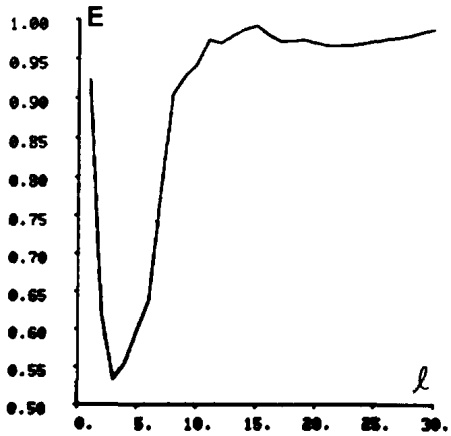


Fig. 5. Mean-square error between $\mathscr{C}$ and $\lambda^l$.

Hence $\lambda^3$ is very similar to $\mathscr{C}(S)$. For $l < 3$ the curvature will be more like trains of impulses (Fig. 2(b)). As '$l$' increases more noise is removed. In Fig. 4(b), $\lambda^5$ is plotted. Compared with $\mathscr{C}$ (in Fig. 2(c)), $\lambda$ is smoother on the straight parts and sharper at corner locations. Hence $l = 5$ seems to be a judicious choice (at least in this example). This value is very close to the actual minimum. $\mathscr{C}$ is not completely noise-free and therefore one should instead use a more reliable reference to optimally estimate $l$. One approach would be to use a shape that can be described analytically, derive its curvature and compare it with an observed curve. This may not be realistic since we have analytic models only for very simple curves and the model itself will suffer for quantization effects as we work in a discrete space.

## 4. Applications

### 4.1. Using curvature for shape description

The simplest further processing of the curvature function is to threshold it in order to eliminate small variations. What remains is a set of positive and negative peaks connected by straight lines. We produce a piecewise contour approximation:

- straight line segments;
- corners (for narrow curvature peaks);
- arc curves (for wide curvature peaks).

If the rapidly turning pieces of curve are considered as angles we have a piecewise polygonal approximation, which is a common way to describe a shape [1].

Let $k$ be the curvilinear abcissa of a corner (or $S_k$). Its measure $\phi_k$ can be approximated by the angle between line segments $L_{k+l}^l$ and $L_k^l$ (Fig. 4(a)):

$$\phi_k = \lambda_k^l. \tag{18}$$

Another type of useful information about a curve is the local concavity (or convexity). Convexity is intimately related to the curvature. The sign of the angular variation $\mathrm{d}\theta/\mathrm{d}s$ (or $\theta_{k+1} - \theta_k$ for the discrete case) describes this function. For a convex closed contour $\mathrm{d}\theta/\mathrm{d}s$ never changes its sign. Therefore convexities correspond to positive peaks while concavities are represented by the negative ones. In Fig. 4(b) the curvature minima show the three concavities (1, 2 and 3) of the initial shape (Fig. 2(a)).

Finally one can measure the spacing between consecutive curvature peaks, which gives the lengths of the line segments.

### 4.2. Flaw detection

Automated inspection of mechanical parts is one of the tasks that can be performed by vision. Fig. 6 shows two items belonging to the same class, a good one (a) and a defective one (b). Area, perimeter, moments or other such parameters do not discriminate them, while angle measurement, if it is performed accurately, allows detection of manufacturing failures.

If we apply our scheme to this family of objects, we produce a description in terms of the cosines or the corners. Neglecting the two extrema we are interested in the angles that determine the global

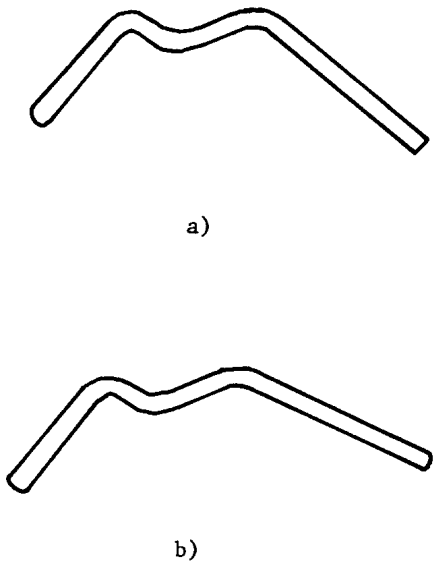


Fig. 6. Automated inspection: (a) good item, (b) defective item.

shape: their measures are $C_1, C_2, \ldots, C_6$. The error between the reference shape $(C_i^0)$ and a defective one is estimated by the formula:

$$e = \sum_{i=1}^{6} |C_i^0 - C_i|. \tag{19}$$

Instead of considering the pattern as a vector in a high dimensional feature space, (i.e. the space defined by the six features $C_1 \ldots C_6$), eq. (19) gives a very good global disimilarity estimation.

We have considered a lot of 190 good items and 11 defective ones, which roughly represents the actual industrial situation where 0.5% of the items were distorted. A reasonable goal is to reduce by a factor of ten the presence of defective shapes.

The parameters for the set of good items (the left column of Table 1) have been obtained by averaging the experimental measurements obtained for all these objects in various conditions (random position and orientation in the TV camera field of view). The standard deviation of the lot for these particular angular measurements is 3%, which gives the following classification strategy:

- If $e$ is less than or equal to 0.11, then the pattern is accepted.
- If $e$ is greater than 0.11, then the pattern is rejected.

The right column of Table 1 gives the parameters of one of the defective items. In this particular case $e = 1.05$.

Table 1

| | Good item (Fig. 6(a)) | Bad item (Fig. 6(b)) |
|---|---|---|
| $C_1$ | −0.62 | −0.60 |
| $C_2$ | −0.82 | −0.72 |
| $C_3$ | −0.34 | −0.87 |
| $C_4$ | −0.50 | −0.83 |
| $C_5$ | −0.80 | −0.75 |
| $C_6$ | −0.62 | −0.60 |

## 5. Conclusion

The aim of this paper was to suggest a corner detection algorithm on a digital curve with the following characteristics:

- to reduce the quantization noise;
- to keep angle sharpness unmodified;
- to be optimal compared to classic filtering;
- to be fast and easy to implement an a minicomputer;
- to be insensitive if the object is rotated.

The algorithm can distinguish between corners and rapidly turning arc curves. Its ability to recognize very flat angles has to be improved.

Its accuracy has been proved in a typical industrial-vision experiment.

## References

[1] T. Pavlidis, "Algorithms for shape analysis of contours and waveforms", *I.E.E.E. Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, July 1980, pp. 301–312.

[2] J.R. Bennett and J.S. MacDonald, "On the measurement of curvature in a quantized environment", *I.E.E.E. Trans. Computers*, Vol. C-24, No. 8, August 1975, pp. 803–820.

[3] J.D. Dessimoz, "Curve smoothing for improved feature extraction from digitized pictures", *Signal Processing*, Vol. 1, No. 3, July 1979, pp. 205–210.

[4] H. Freeman and L. Davis, "A corner finding algorithm for chain coded curves", *I.E.E.E. Trans. Computers*, Vol. C-26, No. 3, March 1977, pp. 297–303.

[5] A. Rosenfeld and E. Johnson, "Angle detection on digital curves", *I.E.E.E. Trans. Computers*, Vol. C-22, No. 9, September 1973, pp. 875–878.

[6] T.P. Wallace, O.R. Mitchell and K. Fukunaga: "Three-dimensional shape analysis using local shape descriptors", *I.E.E.E. Trans. Pattern Ananlysis and Machine Intelligence*, Vol. PAMI-3, No. 3, May 1981, pp. 310–323.