VISUAL IDENTIFICATION AND LOCATION IN A MULTI-OBJECT ENVIRONMENT BY CONTOUR TRACKING AND CURVATURE DESCRIPTION

Jean-Daniel DESSIMOZ

Laboratoire de traitement de signaux
Ecole Polytechnique Fédérale de Lausanne
ch. de Bellerive
Switzerland

Reprinted from the

Proceedings

8th International Symposium On Industrial Robots

10. IPA ARBEITSTAGUNG

4th International Conference On Industrial Robot Technology

Stuttgart, West Germany
30.5.1978 — 1.6.1978

VISUAL IDENTIFICATION AND LOCATION IN A MULTI-OBJECT ENVIRONMENT BY CONTOUR TRACKING AND CURVATURE DESCRIPTION

Jean-Daniel DESSIMOZ

Laboratoire de traitement de signaux
Ecole Polytechnique Fédérale de Lausanne
ch. de Bellerive
Switzerland

SUMMARY

The method presented here has been used satisfactorily for many months in our laboratory, with an experimental set up including a TV vidicon-camera and a four degree of freedom manipulator in order to sort and specifically position objects.

In a multi-object environment pictures must first be segmented in order to isolate objects. Two contour-tracking algorithms are presented, adapted to various types of contours. They allow picture segmentation.

Several global features, such as contour length, silhouette area and moments of inertia, are extracted directly from the isolated contour.

Local shape descriptors such as teeth, notches, etc. are also considered. Features of this type are necessary if an object is not entirely seen, overflowing the picture field or hidden by another object. As local shape descriptor, the curvature K(s) of a contour - closely related to its bending energy - provides highly significant information. Measured in the neighbourhood of every point, the curvature is used both as a function of the position along the contour, and globally through its first statistical moments: mean value m_K and standard deviation σ_K.

A new procedure is proposed, to improve by adequate filtering of parametrized contours the accuracy of contour curvature and length.

Finally σ_K, L, the area A and some groupings of them such as $L \cdot \sigma_K$ and $L^2/2\pi A$ $2\pi A$ will be discussed as features for the identification of entirely seen objects. Furthermore, K(s) will be shown extremely useful for location purpose.

1. INTRODUCTION

Our purpose is to program one of the main tasks that an industrial robot should perform: Visual identification and localization of objects in a multi-object environment. A planar or quasi-planar object can generally be recognized by its shape. It can be described conveniently by its outline. Therefore, our laboratory has developed a contour extrator (Ref. 1).

In a picture where objects are given by their outlines, only the picture elements (pixels) of the contours should be processed. Contour tracking tends to do so: by adequate book-keeping, one can detect closed contours, overcome possible gaps of the outline and anticipate ends (Fig. 1). Moreover, when the tracking of a closed contour is completed, the area covered by the corresponding object is separated from the global image, and thus, picture segmentation is achieved.

From this isolated region, several global features, such as perimeter, silhouette area and moments of inertia are directly extracted in order to identify and locate the corresponding object.

Local descriptors, such as teeth or notches should also be considered: features of this type are necessary if an object is not entirely seen, overflowing the picture field or hidden by another object. We have chosen, as local shape descriptor, the curvature of contours (closely related to their bending energy). Measured in the neighbourhood of every point, the curvature is used both as a function of the position along the contour K(s) and globally, through its first statistical moments: mean value m_K and standard deviation σ_K. While m_K and σ_K are used for the identification of completely seen objects, segments of K(s) can be used for the recognition of partly seen objects, and for orientation purpose by locating local properties of contours. For example, the respective position of two notches on an object gives the object orientation.

A new procedure is proposed to improve the accuracy of contour curvature and length by adequate filtering of contours, thus allows better identification and location of objects.

Let us summarize the general context of our method development:

Assumptions:
- good contrast between background and objects
- vidicon T.V. camera with 16 levels of digitization
- 256 x 256 samples per frame
- several objects per picture (in a first approach, they are not assumed to overlap).
- possible implementation in specialized hardware.

Aims :
- 1 object recognized and localised per second in average
- accuracy better than 1% in position and in orientation detection
- possibility of lowering the speed to increase accuracy and reciprocally.

Several methods process already one-object images for recognition and localisation (Ref. 2,3,4). Under these restrictive assumptions, the method presented below leads to results similar to theirs. Moreover, it can cope with several objects invading the scene at the same time. Other works have been reported, closer in some respect of what is shown here. In particular K.J. Turner (Ref. 8), J.E. Bowie (Ref. 5), J.W. Mac Kee and Aggrawal (Ref. 6) have used tangent or curvature functions along contours.

2. CONTOUR TRACKING

When a contour is tracked, it appears as a sequence of pixels. Being on a pixel, one moves to the one of its neighbours that also belongs to the contour and so on. The neighbourhood of a pixel (let us choose P_0) can be defined more or less strictly : either it shrinks to be the eight or even the four pixels that touch P_0, and this leads to Freeman's chain code (Ref. 7), or a larger neighbourhood radius r is considered (5,7,10... grid units), which leads to about 2 πr neighbours (30,45, 60 pixels !). Two algorithms have been programmed ; one is parametrized (r can be optimized and the adresses of the neighbours at r distance are in a look-up table), the other encodes contours with Freeman's chain code. This latter method appears to be faster and more accurate in the case of very thin outline (Fig. 2). These two algorithms lead to different frequency content for the resulting curvature and abscissa.

3. FEATURE EXTRACTION

In order to identify and locate an isolated contour, the first moment of inertia, the curvature in every contour element, the area, the perimeter length and some statistical moments of the curvature are extracted.

3.1 Curvilinear abscissa

Let "s" be the curvilinear abscissa defined along a contour as the arc length starting in O, an arbitrarily chosen origin. With respect to the coordinates of the picture x & y, the abscissa can be expressed as follows :

For a continuous contour :

$$s_k = \int_{t_o}^{t_k} ds = \int_{t_o}^{t_k} \sqrt{\{\phi'(\tau)\}^2 + (\psi'(\tau))^2}\, d\tau \qquad (1)$$

with the parametric definition of the outline :

$x = \phi(t)$; $y = \psi(t)$

When the parameter is sampled with such a space between samples (it does not have to be constant) that every pixel of the contour is taken into account, then "s" becomes in any point P_k of the outline :

$$s_k = \sum_{i=1}^{k} \Delta s_i = \sum_{i=1}^{k} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \qquad (2)$$

3.2 Center of mass

The center of mass of an outline C_m is its first moment of inertia.

$$x_{Cm} = \frac{\sum_{i=1}^{N} m_i x_i}{\sum_{i=1}^{N} m_i} \quad ; \quad y_{Cm} = \frac{\sum_{i=1}^{N} m_i y_i}{\sum_{i=1}^{N} m_i} \qquad (3)$$

where m_i is the mass of the contour element in P_i; as the contour is sampled, m_i is defined as follows :

$$m_i = k \left(\frac{s_{i+1} - s_{i-1}}{2}\right)$$

With L, the perimeter length, it becomes :

$$x_{Cm} = \frac{\sum_{i=1}^{N} (s_{i+1} - s_{i-1})x_i}{\sum_{i=1}^{N} (s_{i+1} - s_{i-1})} = \frac{\sum_{i=1}^{N} (s_{i+1} - s_{i-1})x_i}{2L} \quad ; \quad y_{Cm} = \frac{\sum_{i=1}^{N} (s_{i+1} - s_{i-1})y_i}{2L} \qquad (4)$$

3.3 *Curvature*

The curvature K(s) is defined for continuous outlines by :

$K(s) = \frac{d\theta}{ds}$ where θ is given by the tangent to the contour at P(s)

$\theta = \text{Arc tan}\,\frac{dy/ds}{dx/ds}$

For a sampled outline, we have :

$$K(s_k) = \frac{\Delta\theta_k}{\Delta s_k} = \frac{\theta_k - \theta_{k-1}}{\frac{(s_{k+1} - s_{k-1})}{2}} \quad \text{where } \theta_k = \text{Arc tan}\,\frac{y_{k+1} - y_k}{x_{k+1} - x_k} \qquad (5)$$

3.4 *Area*

The silhouette is often chosen as a feature for the recognition of planar objects. A pseudo-area can be directly derived from the outline. However, to estimate the true visible surface of an object, one should still deal with the possible holes. In this latter case, the followed outline is not enough but may be used as a limit for a subsequent analysis : reading of the contours inside the isolated region, or even analysis of the primary picture which still contains surfaces rather than contour lines.

3.5 *Perimeter*

Perimeter is an immediate feature of an object when picked out by edge-tracking :

$L = \sum_{i=1}^{N} \Delta s_i$ where N is the number of elements of the sampled contour.

3.6 *Statistical moments of the curvature*

K(s) is estimated in N points regularly spaced along a contour. We can then estimate very easily its average m_K and its standard deviation σ_K.
For a closed contour :

$$m_K = \frac{1}{N}\sum_{i=1}^{N} K(S_i) = \frac{1}{N}\sum_{i=1}^{N}\frac{\Delta\theta_i}{\Delta s_i} = \frac{\pm 2\pi}{N\Delta s_i} = \frac{\pm 2\pi}{L} \qquad (6)$$

$$\sigma^2_K = \frac{1}{N}\sum_{i=1}^{N} K^2(s_i) - m_K^2 \qquad (7)$$

Relation (6) gives again the information of perimeter already extracted. On the contrary, the standard deviation σ_K is very meaningful, showing how much the curvature changes along the contour. Therefore, it has also been chosen as a basic feature of objects.

3.7 *Remarks*

Notice that the quantization noise on the x and y coordinates of contours is another way to consider the spatial sampling of two-level pictures. When considered parametrically $\{x = \phi(t), y = \psi(t)\}$, a contour is not sampled but quantized.
In the preceeding paragraphs, various features have been extracted from contours They all are functions of the x and y coordinates. Now, x and y are quantized (in our case to 256 levels). How does this affect the features ? While center of mass and especially area are not too distorted, the tangents θ_i, the curvatures K_i and the contour element lengths are strongly affected (respectively 8, 8 and 2 possible values !).
Quantization noise and original x and y functions have different spatial frequency content. Therefore, in order to reduce the noise influence, a filter has been designed, with respect to a spatial variable : the curvilinear abscissa s. However, only a noisy estimation $\hat{s}$ of s is available. But provided that this estimation is not too bad, the noisy estimates $\hat{x}$ and $\hat{y}$ of x and y can be improved. Then $\hat{s}$ is better evaluated when computed from these improved $\hat{x}$ and $\hat{y}$ functions. As iteration goes on, $\hat{s}$, $\hat{x}$ and $\hat{y}$ tend respectively to s, x and y.

Let $\hat{v}^{(o)}$ be an observed variable and $\hat{v}^{(n)}$ the same variable after n iterations.

$$\hat{x}^{(n)} = h(\hat{s}^{(n-1)}) * \hat{x}^{(o)}(\hat{s}^{(n-1)}) \qquad (8)$$

$$\hat{y}^{(n)} = h(\hat{s}^{(n-1)}) * \hat{y}^{(o)}(\hat{s}^{(n-1)}) \qquad (9)$$

and $\hat{s}^{(n)} = f(\hat{x}^{(n)}, \hat{y}^{(n)})$ cf (3.1)

In practice, two iterations are generally sufficient; the error on $\hat{s}$ leads to a jitter of the filter impulse response that processes the coordinates.

4. IDENTIFICATION

In paragraph 3, several features have been introduced. They fall into very different categories : global features, which result from properties of an object considered as a whole (perimeter length, area, statistical moments of the curvature); and local features, which describe only small part of an object, without any effect from the other parts of the same object : the K(s) function consists of such features.
Under the condition of completely seen objects, the global features can be used. For our applications, they are sufficient to allow the recognition of objects (however, they do not distinguish between the observe and reverse visible sides of objects). A nearest neighbour rule is applied to the feature space. An object is recognized when the "distance" T is minimal between its features and the corresponding model, memorized in a learning phase.

Criterion :

$$T_1 = \alpha\left|\frac{\hat{L}-L_1}{L_1}\right| + \beta\left|\frac{\hat{S}-S_1}{S_1}\right| + \gamma\left|\frac{\hat{\sigma}_K-\sigma_{K1}}{\sigma_{K1}}\right| \qquad (10)$$

α, β and γ balance the effect of every feature, as a function of its dispersion.

$l = 1,2\dots N$ for N models.

If $T_i < T_j$ for all $j \neq i$ and $T_i < T_{minimum}$

the object is recognized of the same type as the i^{th} model.

Remark : let d be the distance between an object and the camera. In first approximation :

$L \propto 1/d$
$S \propto 1/d^2$
$m_K, \sigma_K \propto d$

Then, these features can be merged to produce new ones, size invariant.

$$I_1 = \frac{S}{4\pi L^2} \qquad (11) \qquad\qquad I_2 = \frac{\sigma_K}{m_K} = L\,\frac{\sigma_K}{2\pi} \qquad (12)$$

Altough the well known I_1 is more difficult to estimate in our experimental conditions (cf 3.4), I_2 can be implemented easily since L is given by the contour-tracking algorithm and σ_K is computed on K(s) necessary for location (cf § 6). However, these invariants have the drawback that they do not balance the various features according to their noise. Therefore, they are often too dependent on the most noisy features.

5. OBJECT LOCATION

The position of an object can be defined in one point only, independently from any rotation. We have chosen to consider the rotation of objects with respect to C_m (x_{Cm}, y_{Cm}), the center mass of the outline. The position of C_m is given by the (4) formulae. If one wants the position of an other point (for example $P_0(x_{P_0}, y_{P_0})$), a correction which varies with rotation should be taken into account.

$$x_{P_0} = x_{Cm} + r_{CP}\cos(\theta_r + \theta_{CP}) \qquad (13)$$

$$y_{P_0} = y_{Cm} + r_{CP}\sin(\theta_r + \theta_{CP})$$

$$\text{with } r_{CP} = \sqrt{(x_{Cm} - x_{P_0})^2 + (y_{Cm} - y_{P_0})^2} \quad \text{and } \theta_{CP} = \text{Arctan}\frac{y_{P_0} - y_{Cm}}{x_{P_0} - x_{Cm}} \qquad (14)$$

θ_r is the rotation angle of the object with respect to its model (cf § 6).

6. OBJECT ORIENTATION

Consider an outline with a curvilinear abscissa s and a vector $\vec{v}$ defined by an origin and an end on the outline. This vector is called orientation vector :

$\vec{v} = \overrightarrow{P_0P_1}$ or, in a parametrical form, $\vec{v} = \overrightarrow{P(s_0)P(s_1)}$

In a learning phase, every type of object is given such a vector. Consider the same outline after translation and rotation. The corresponding variables are s', $\vec{v}'$, P' (cf Fig. 4).

First property : if the outline turns with a θ_r rotation, its orientation vector rotates with the same angle. Reciprocally, the difference between the arguments of $\vec{v}$ and $\vec{v}'$ shows the rotation of the object. The problem remains,to find the orientation vector of an object encoded by its outline.

Second property : every point P_0 of a contour is characterized by a curvature $K(s_1)$ in this point. This curvature does not vary with translation and rotation.
Suppose $K(s)$ known for the first contour. When the second outline is analysed, an arbitrary origin is assigned to the curvilinear abscissa; this leads to a new function $K'(s')$, different from $K(s)$ only by an offset (cf second property and Fig. 5).
This offset is detected by cross-correlation.

For a closed contour $K(s)$ is periodical :

$$R_{KK'} = \frac{1}{L} \int_0^L K(\eta)\, K'(\eta+\tau)d\eta \tag{15}$$

and for an open contour :

$$R_{KK'} = \lim_{L\to\infty} \int_{-L/2}^{L/2} K(\eta) \,.\, K'(\eta + \tau)d\eta, \tag{16}$$

assuming a zero value for K and K' where they are not defined.

$R_{KK'}$ shows a maximum value for such a τ that $K(s) = K'(s'-\tau)$.
When the size does not change or is normalized, an other function can be used, which avoids multiplications : the "mean absolute difference" (MAD).
For a closed contour :

$$MAD(\tau) = \frac{1}{L} \int_0^L \left|K(\eta) - K'(\eta+\tau)\right| d\eta \tag{17}$$

This function has a minimum value when τ leads to $K(s) = K'(s'-\tau)$.

However, the crosscorrelation can reveal false maxima when the functions are noisy. Therefore, two quality criteria are derived from the MAD using mean value, standard deviation, first and second minima of the function :

$$C_1 = \frac{m_{MAD} - min_{MAD}}{\sigma_{MAD}} \quad (18) \qquad C_2 = \frac{min_{MAD} - min_{2MAD}}{m_{MAD}} \quad (19)$$

Experimentally, C_1 and C_2 have not been found very significant. Their two probability distributions (in case of success and in case of failure) overlap widely.
The problem can be solved in another way. Let two T/2 wide windows be applied on K(s), without overlapping. When they are mutually dependent, they lead in some sense to a better result than when they are applied independently. The former case means a window of equivalent width T and a better estimation of the complete function (the spectrum has a better definition). In the latter case, however, a criterion is available : if both windows, independently lead to the detection of the same offset, the probability of success is high.
Let us see better the second case; when each of the segments selected by a window on K(s) gives a maximum of the crosscorrelation function for the same value τ, the noise has probably no influence; for the noise that disturbs the crosscorrelation does not depend on the real contour and has no relation through the windows. If, on the contrary, the windows lead to extrema at different τ values, that is a failure : the windows must be broadened (the computation time will increase) in order to reduce the relative influence of noise and thereby to find the true offset.

Remark :
A good quality criterion allows a better trade-off between good results and computation time : the crosscorrelation is generally estimated with a rather narrow window, at high speed; however in presence of failure, detected by the criterion, the computation starts again, with more complexity for this particularly difficult case.
Instead, without criterion, the worst case conditions would force the choice of a very wide window for every estimation.

7. RESULTS

7.1 *Filtering and identification*

For the mechanical parts of the Photo 1, a recognition rate of 100 % has been achieved using as features, the perimeter length and the standard aviation of the curvature along the outline only. As noticed in (3.7), both the perimeter length and the curvature are computed with the filtered x and y functions.
A simple average of x and y over about 10 grid units has lead to a satisfactory accuracy of the perimeter length; after resampling, a further average over 2 samples (equivalent to about 10 grid units) has been found good enough for the curvature estimation (Fig. 6).
It has been experimented that when the filtering is performed on more than 10 samples, the error on perimeter length does no longer decrease. This comes from the fact that quantization noise disappears with respect to a very low spatial frequency distortion due to camera (Fig. 6c).

7.2 *Location and orientation detection*

The accuracy of center of mass may seem to be very insensitive to quantization noise, since it results from averaging. But experiments show important variations of its position, as a function of the orientation, when it is computed directly from the coordinates (sometimes the error amounts to more than 10 quantization steps). However, after filtering of the coordinates and weighting of each contour element proportionally to its length (cf 3.2 and 3.7), the errors of position can be reduced to less than 1 quantization step (1 quantization step = 1 grid unit).

The accuracy in orientation detection depends at a first level on the quälity of the orientation vector origin and end. At the higher level, it depends on the quality of the offset detection by crosscorrelation as presented in §6. While the maximum error involved at the first level can easily be estimated, the error at the next level depends on the particular location of the orientation vector, and on the particular shape of the object. In the learning phase, one will choose a good position of the orientation vector with respect to the outline, so as to minimize the sensitivity of its orientation to little shifts of its ends along the contour.
In our experiments, the orientation was detected with a rather good accuracy (cf fig. 7). If the crosscorrelation detects the good offset between the model function K(s) and the new function K'(s') being processed, the accuracy of the detected orientation falls in a range of ±3 degrees. However, the true offset may be missed and this leads to unpredictable results, in case of symmetries in the outline of objects (cf photo 2).

8. CONCLUSIONS

A method has been shown which alows the identification and the location of multiple objects in a TV camera field. The tracking procedure that partitions the image is followed by a quantization noise filter and a regularly spaced resampling. This first processing takes half of the total computation time on PDP11/40.
Then, object recognition and localization are performed. In our experiments, results are satisfactory : the accuracy is better than one grid unit for translation, 3^0 for rotation and pratically 100 % for the recognition.
Our main limitations are as shown on photo 2 symetries of objects (subperiods in the curvature function), losses of information about texture and details inside boundary (such as holes...), and the computation time. Although the complete computation time with programs written in Fortran, on a 24 k words of memory PDP 11/40 is about 15 seconds long presently this should be reduced to less than one second by specialised hardware.

The method presented here has been used satisfactorily for many months in our laboratory with an experimental set up including a four degrees of freedom manipulator.

9. ACKNOWLEDGEMENT

The author is deeply grateful to Prof. F. de Coulon and to Dr M. Kunt for many helpful discussions, to Prof. C.W. Burckhardt for his contribution in the early development of this work and to J.M. Zürcher for his support in hardware design. This work was supported by the Swiss Commission d'Encouragement pour les Recherches scientifiques (project 880).

REFERENCES

1. Zürcher, J.-M. "Video Picture Digitizer" (Numériseur d'images vidéo), Report LTS 76.07, Laboratoire de Traitement de Signaux, Ecole Polytechnique Fédérale de Lausanne, (1976).

2. de Coulon F. et Kammenos P. "Polar Coding of Planar Objects in Industrial Robot Vision", Neue Technik, No. 10, (1977).

3. Agui, T. and Nakajima, M. "Picture processing by Shape Contour Functions", Tokyo Institute of Technology, Private communication.

4. Dudani, S.A., Breeding, K.J. and Mc Ghee "Aircraft Identification by Moment Invariants", IEEE Irans on computers, (January 1977).

5. Bowie, J.E. "Syntactic Descriptions of Biological Shape", Ph.D. Thesis, Massachussets Institute of Technology, (1973).

6. Mc Kee, J.W. and Aggarwal, J.K. "Computer Recognition of Partial views of Curved Objects", IEEE Irans on Computers, C-26, No. 8, (August 1977).

7. Freeman, H. "On the digital computer classification of geometric line patterns", Proc. Nath. Elect. Conf. 18, 312-324 (1962).

8. Turner, K.J. "Computer Perception of Curved Objects using a Television Camera", Ph.D. Thesis, University of Edinburgh, (1974).

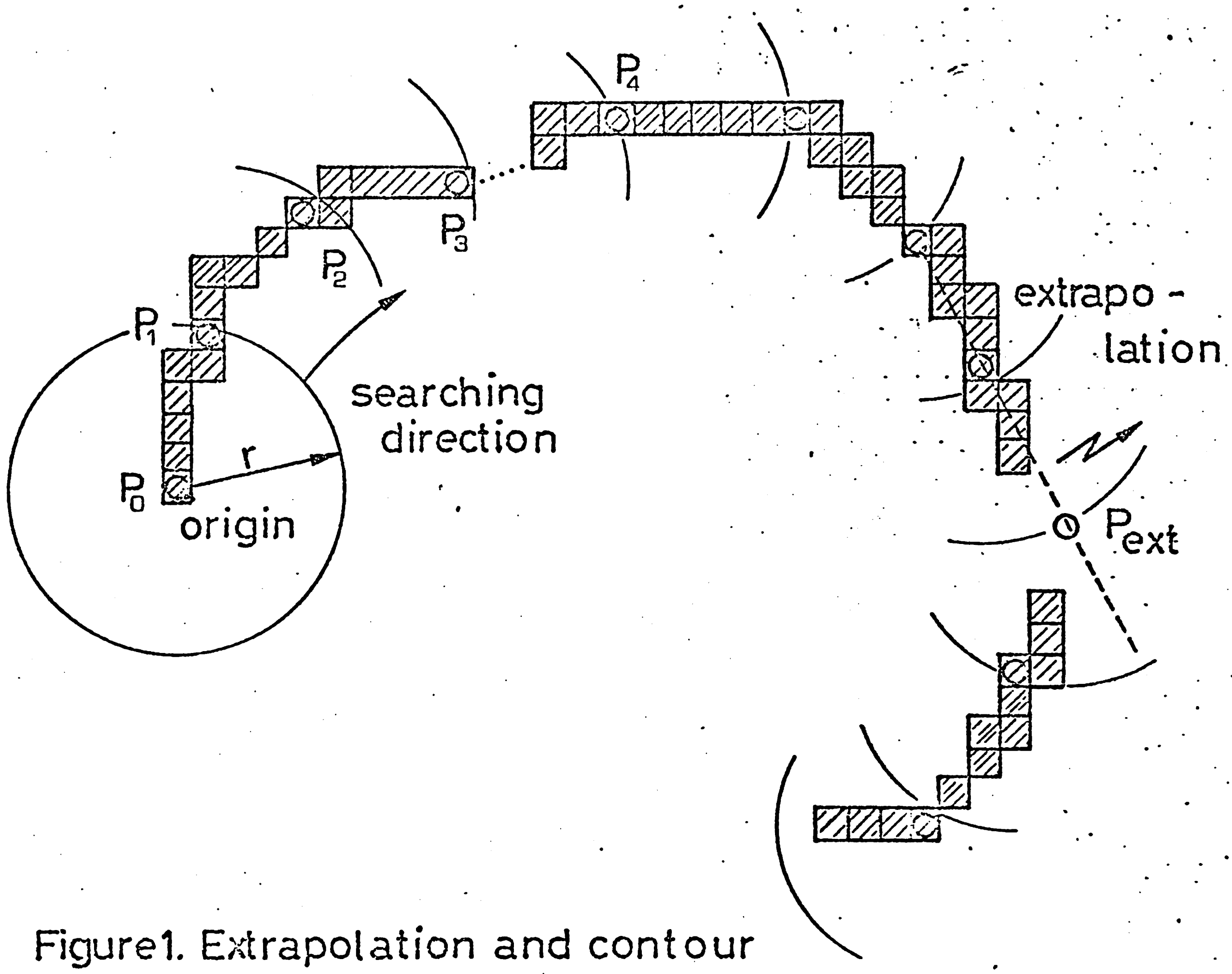

Figure1. Extrapolation and contour end.

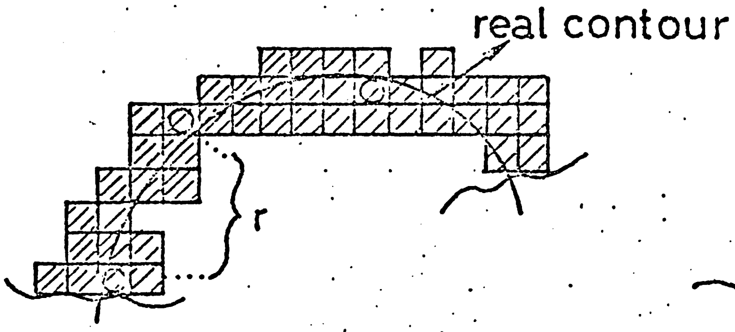

Figure2a.Thick contour

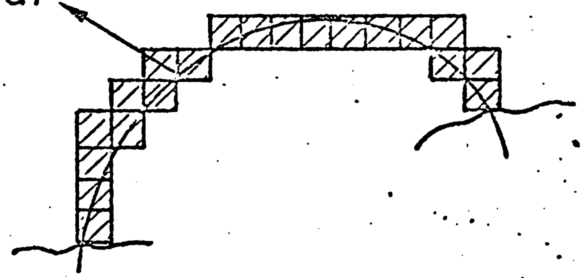

Figure2b.Thin contour

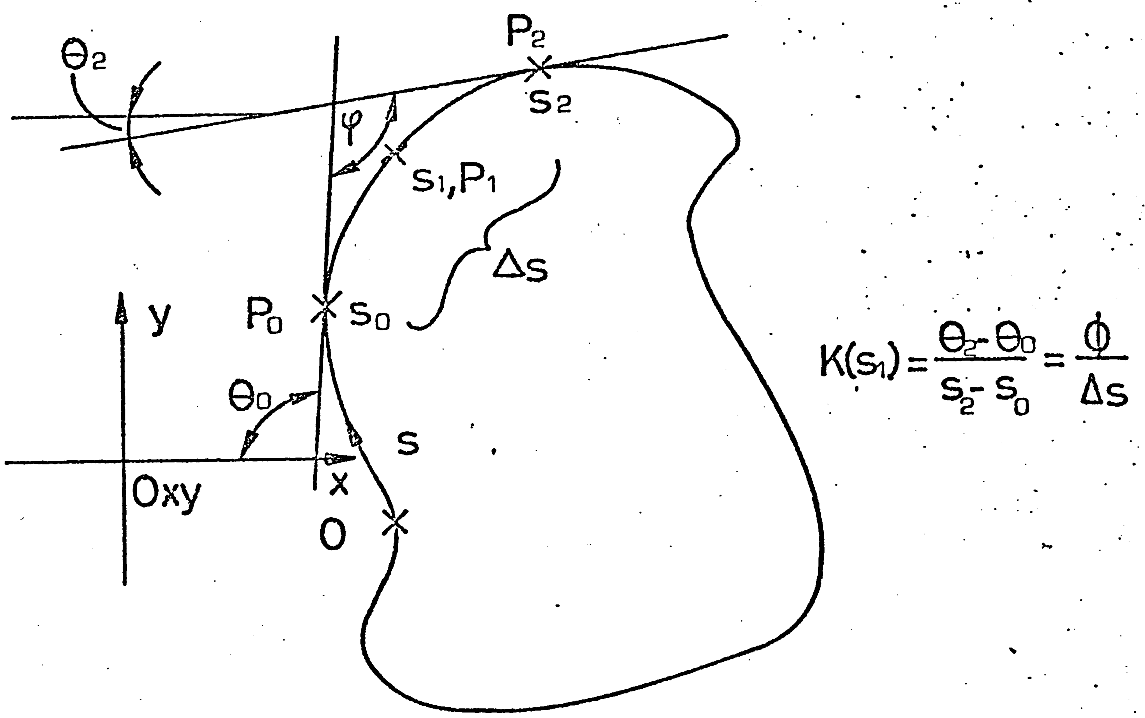

Figure 3. Curvilinear abscissa s and curvature K.

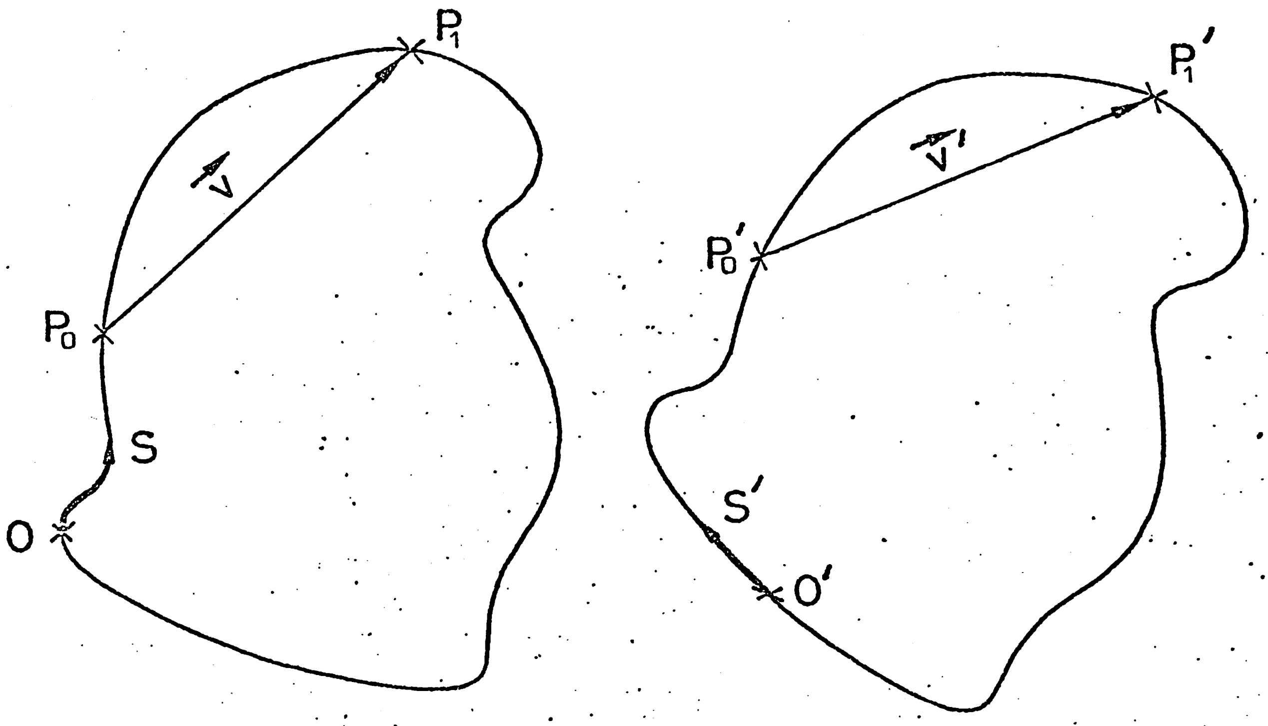

Figure 4. Orientation vector and rotation of an outline.

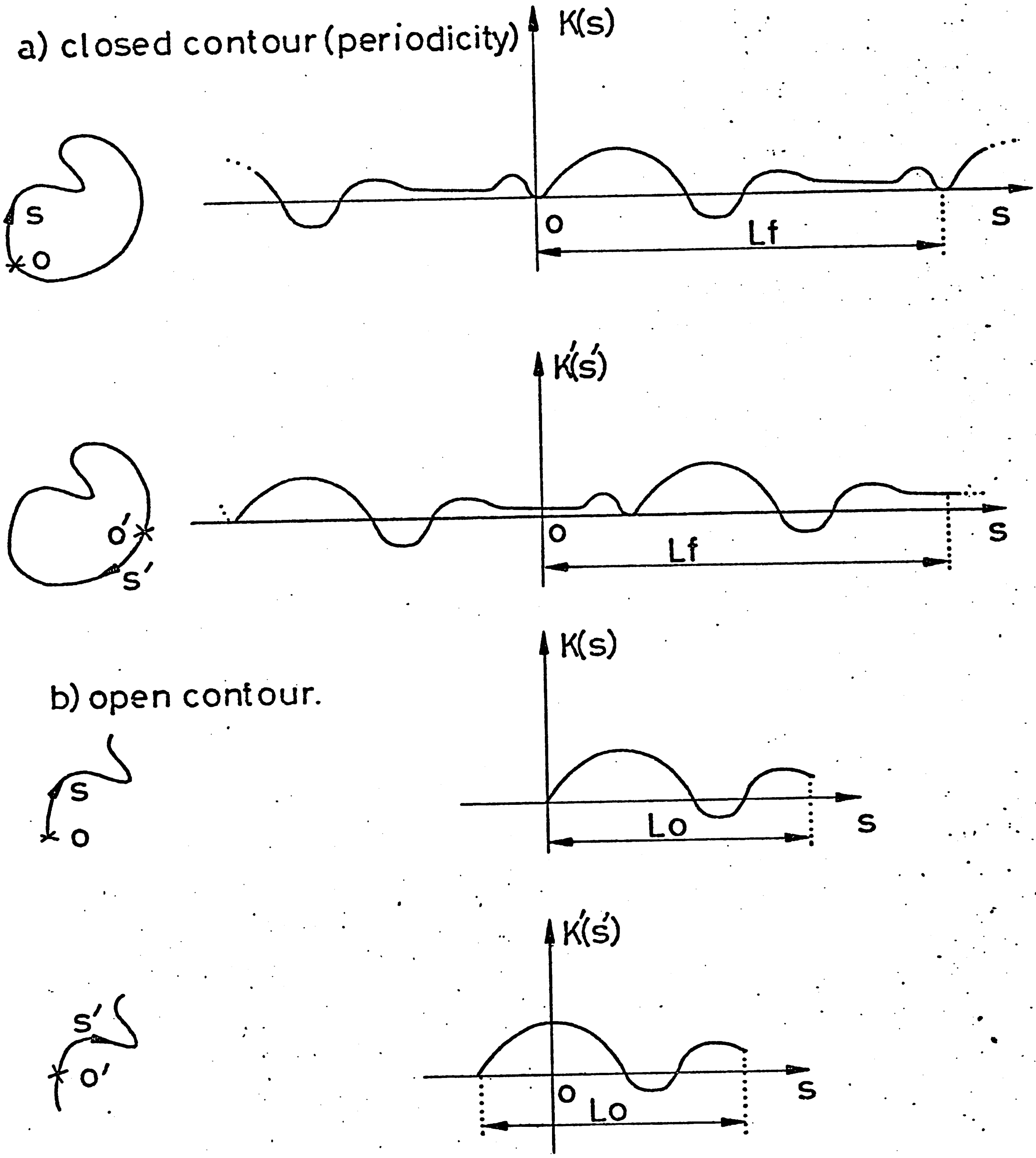

Figure 5. The offset results from a different choice of the origin, and does not depend on the orientation.

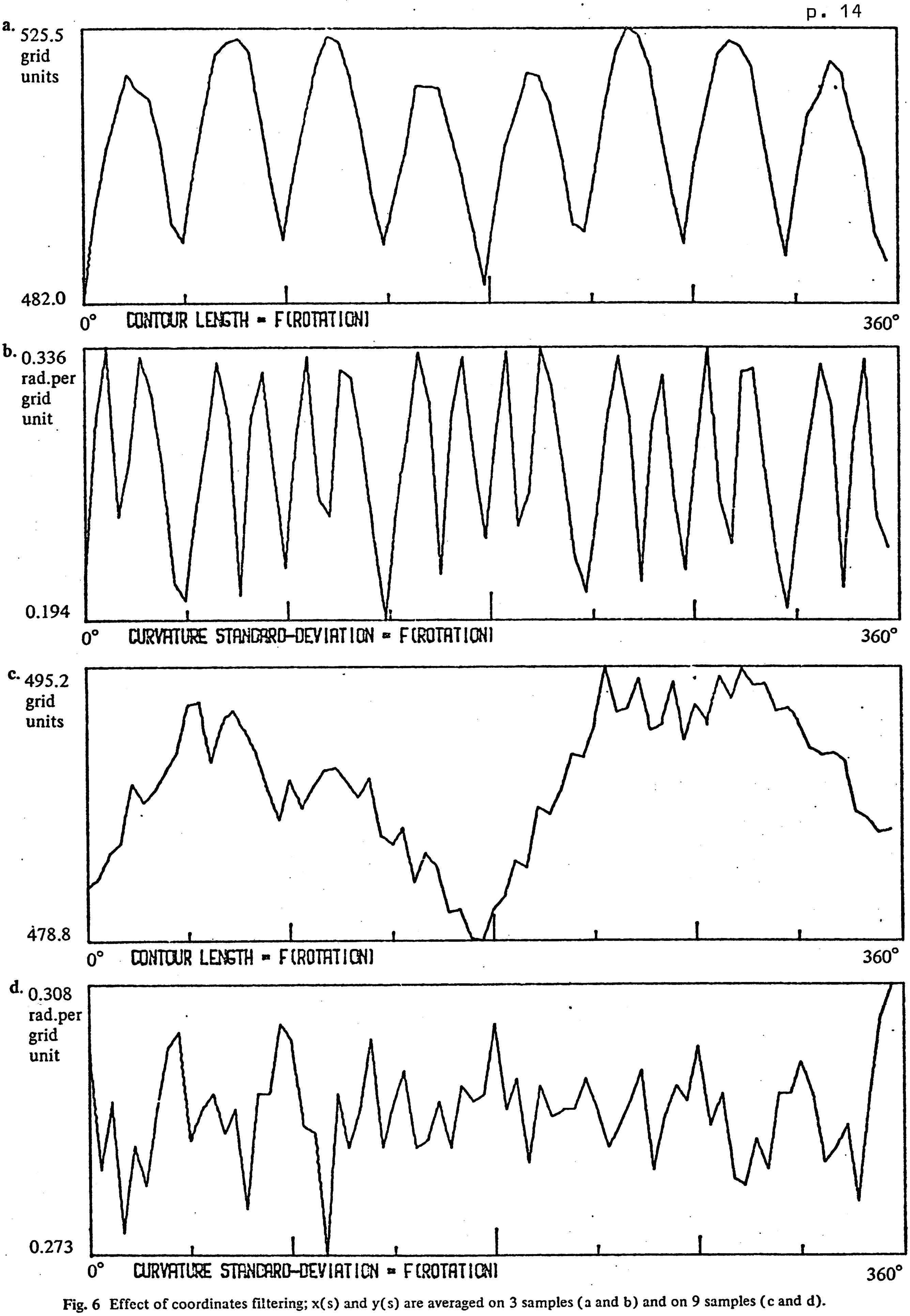

Fig. 6 Effect of coordinates filtering; x(s) and y(s) are averaged on 3 samples (a and b) and on 9 samples (c and d).

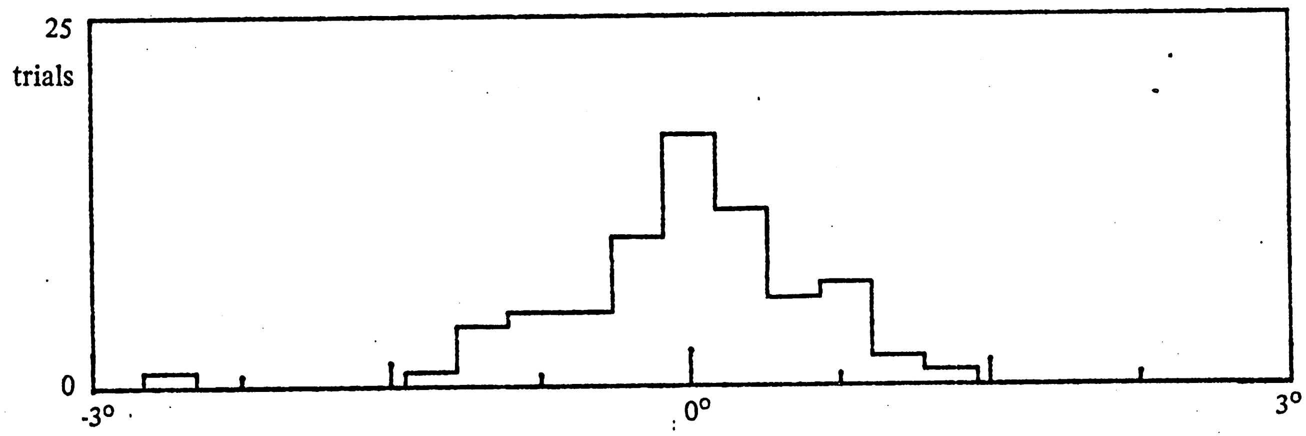

Fig. 7 Histogram of the error in orientation detection of objects, for 72 orientations uniformly distributed between 0° and 360°.

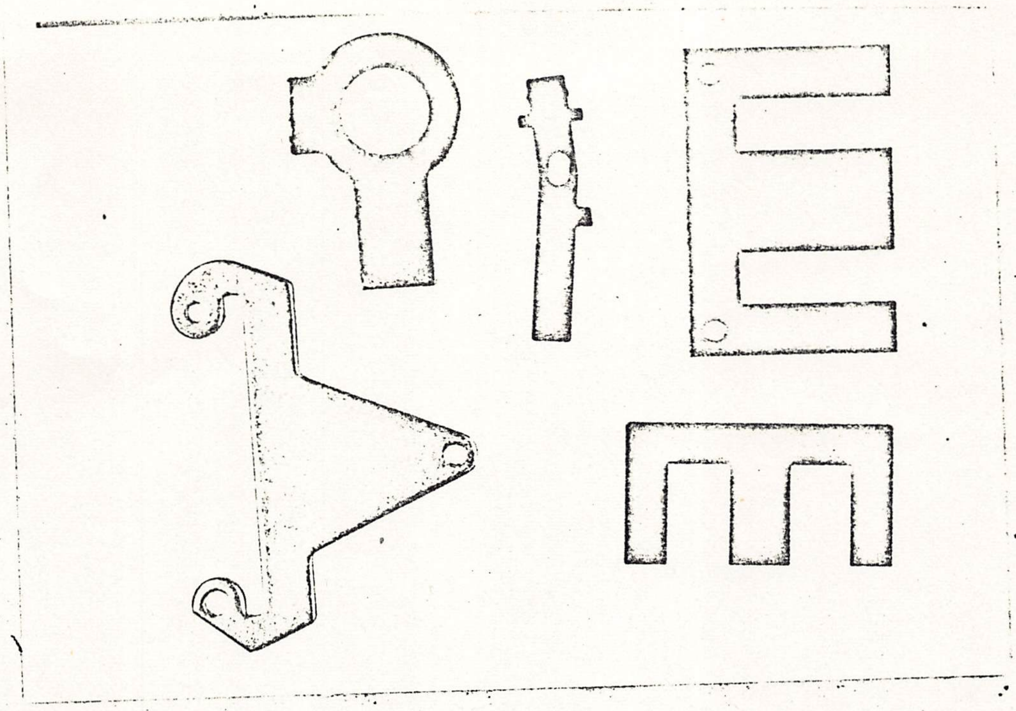

Photo. 1   Typical industrial parts used in our experiments.

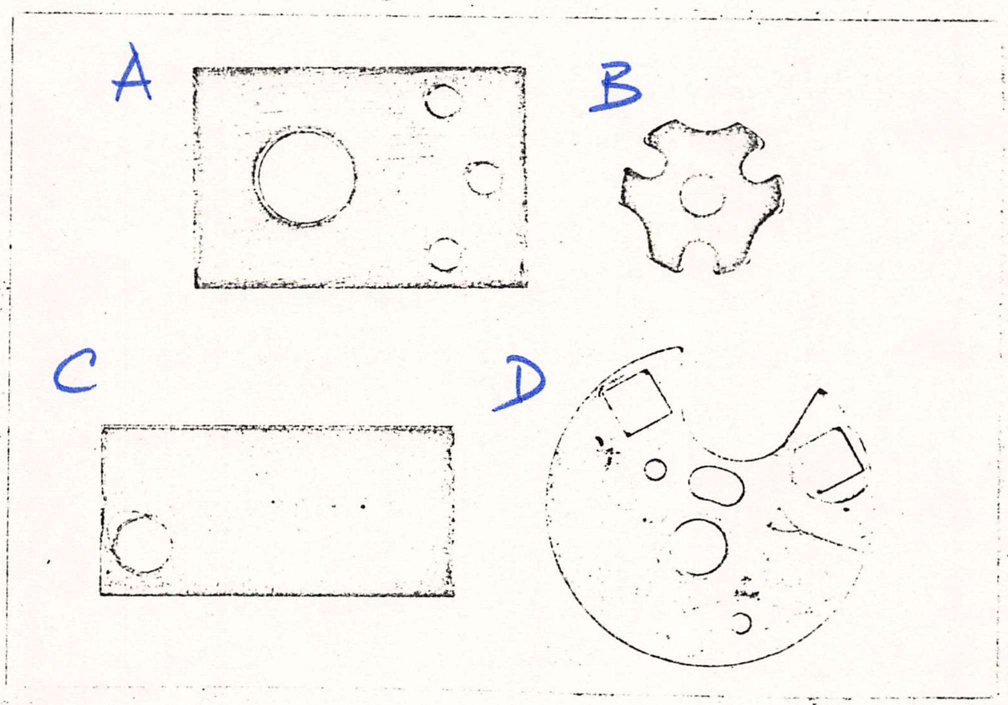

Photo. 2.   A further analysis of object properties within the outline should be done in order to detect the orientation of objects A,B,C (subperiods of contour curvature) and to discriminate the obverse and reverse faces of object D (symmetry).