



Polar contour shape descriptors in the template matching approach to object recognition

Dariusz Frejlichowski*

*Faculty of Computer Science and Information Systems,
West Pomeranian University of Technology, Szczecin
49 Żołnierska St., 71-210, Szczecin, Poland*

Abstract

The paper provides a review of contour polar shape descriptors used in recognition of objects based on their silhouettes. The process of recognition in the template matching approach has to be based on so called descriptors, assigned to object features, e.g. shape, texture, color, luminance, context of the information and movement. Amongst them very special attention is paid to the shape, because in many applications it is the most relevant and the less changeable feature that can be used.

The shape in the digital image processing has usually a form of binary object. One of the representations uses the boundary, contour of a silhouette. The most important advantage of such approach is a small number of pixels to consider.

Amongst several dozen shape descriptors special properties can be found in the polar ones, which use the transformation from the Cartesian to the polar coordinates. The most important is invariance to translation of the object points. The rotation becomes a circular shift what can be easily solved in further processing. Owing to the normalization the descriptors can be also invariant to scaling. Some of the methods are also robust to some level of noise and occlusion.

*E-mail address: dfrejlichowski@wi.ps.pl

1. Introduction

The template matching is one of the approaches to objects identification or recognition. It is based on matching a particular representation of an object with the elements stored in the data base, represented in the same way. The description of an object has to utilize some features, e.g. colour, texture, shape, luminance, context of the information, movement, etc. One of them, commonly applied, is the shape. This representation has many advantages, but the most important is the possibility of its application to various objects. Sometimes it is the only possible feature, especially when an image is distorted due to the changes of weather or light conditions.

There are two main ways of representing a shape in recognition. It can be treated as its boundary only or as the whole shape with its interior. In this paper only the first representation is considered. Its main advantage is a smaller number of points to process.

A shape descriptor has to be robust to as many shape deformations as possible. Those deformations can be divided into four groups. The first one is related to special problems occurring only in the contour representation. Those are: selection of the starting point and the direction of tracing the contour during processing. The second group covers affine transforms of the shape according to the original unaffected one. Those are: shifting (translation) of an object in the image plane, change of size (scaling) and rotation. The third group includes problems of various positions of single points as influenced by noise or discontinuities in the boundary. The last group is related to deformations of the shape as a whole. The first problem here is the varying number of points and the second one is the occlusion, where part of an object is missing or some parts are added, e.g. as a result of shadows. The most difficult and challenging are the problems belonging to the last two groups.

Amongst algorithms used for representation of shapes very interesting properties can be found in the polar ones. Those methods are based on transformation of points in the contour from the Cartesian to the polar coordinates. Those descriptors are easy and fast to obtain. They are invariant to translation (thanks to the calculation of new coordinates according to a particular point, usually inside the shape) and scaling (if the normalization is applied). The rotation becomes a circular shift after the transform, which can be easily solved using another step in the algorithm (e.g. Fourier transform or histogram).

The goal of this paper is to provide description of the most popular and efficient polar contour shape representations. They will be presented in the consecutive sections.

2. Centroidal Distance

The first method described here is the oldest one as well. Because the idea of transforming contour points to polar coordinates is rather obvious, it is difficult to point out the first usage of this solution. For decades it has been found in many algorithms and under various names, e.g. the signatures or Centroidal Distance ([1]). In fact, the approach is so simple that it even does not exploit the whole transformation. In fact, only the vector of distance from the centroid (centre of gravity, calculated as the mean of Cartesian coordinates for all points) is used. However, the order of achieved points is remembered according to the increasing value of the angle.

The Centroidal Distance is invariant to translation, owing to the usage of centroid and scaling as well as the normalization.

The algorithm can be easily formulated. Assuming that the Cartesian coordinates of n points in the contour are stored in vectors X and Y :

$$X = [x_1, x_2, \dots, x_n], \quad Y = [y_1, y_2, \dots, y_n],$$

the centroid can be calculated as:

$$O = (O_x, O_y) = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i \right). \quad (1)$$

The vector D of distances from O can be calculated as:

$$d_i = \sqrt{(x_i - O_x)^2 + (y_i - O_y)^2},$$

where $i = 1, 2, \dots, n$.

The vector D is the achieved description of the object.

3. UNL descriptor

The UNL (*Universidade Nova de Lisboa*) shape descriptor is based on the UNL transform ([2]). It uses complex representation for the Cartesian coordinates of points and parametric curves, but described in the discrete manner ([2]):

$$z(t) = (x_1 + t(x_2 - x_1)) + j(y_1 + t(y_2 - y_1)), \quad t \in (0, 1), \quad (2)$$

where $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$ (complex numbers) and z_i means the point with coordinates x_i, y_i .

The centroid is calculated (denoted as O) and the maximal Euclidean distance between the points and the centroid is found ([2]):

$$M = \max_i \{ \|z_i(t) - O\| \}, \quad \forall i = 1, \dots, n, \quad t \in (0, 1).$$

The transformation of coordinates is formulated as follows ([2]):

$$U(z(t)) = R(t) + j \times \theta(t) = \frac{\|z(t) - O\|}{M} + j \times \arctan \left(\frac{y(t) - O_y}{x(t) - O_x} \right).$$

The discrete version can be also used ([2]):

$$U(Z(t)) = \frac{\|(x_1 + t(x_2 - x_1) - O_x) + j(y_1 + t(y_2 - y_1) - O_y)\|}{M} + j \times \arctan \left(\frac{y_1 + t(y_2 - y_1) - O_y}{x_1 + t(x_2 - x_1) - O_x} \right). \quad (3)$$

The parameter i is discretized in the interval $[0,1]$ with significantly small steps ([2]).

The UNL descriptor is a very efficient algorithm. It is robust to the problem of discontinuities and a varying number of points OWING to the usage of parametric curves. It is invariant to translation as a result of the usage of a particular origin of the transform (centroid) and to scaling owing to the normalisation. The rotation becomes a vertical circular shift after the transform and has to be solved later (e.g. using the Fourier transform or histogram).

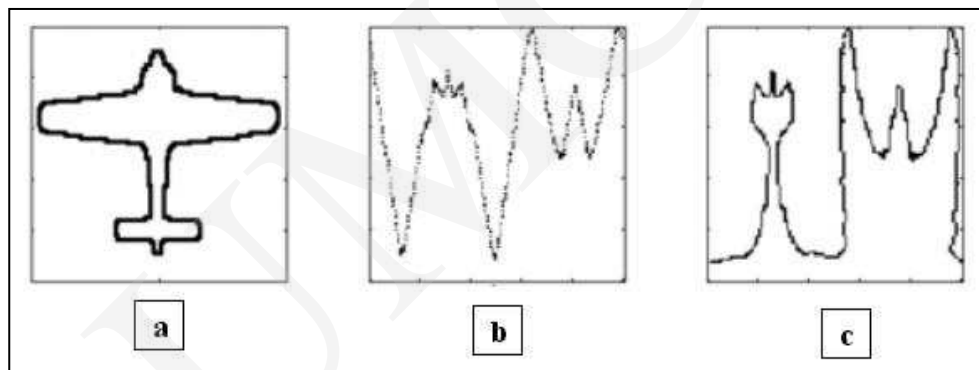


Fig. 1. Illustration of the difference in appearance between the two descriptors: a) an object, b) Centroidal Distance, c) UNL ([3])

The most important advantage of the UNL is the fact that it is more distinguishable for various shapes owing to the specific method of writing the resultant coordinates – they are put into matrix, in which row is corresponding to the distance from the centroid, and column – to the angle (see Fig. 1). The obtained matrix is 128×128 pixels size.

4. Log–Pol descriptor

The Log–Pol descriptor is based on the addition of the logarithmic transform to the polar one. As in the case of polar transform this approach is invariant to

translation. After the logarithmic–polar transform we receive the image, where the angular coordinates are placed on the vertical axis and the logarithms of the radius coordinates are placed on the horizontal one. Because of the periodic nature of the angular component, rotations in the original image become vertical cyclic shifts in the transformed one. The changes of size in the original image become horizontal shifts in the transformed image, according to the autosimilarity property of the exponential function ([4]).

The method starts with the calculation of centroid O (see eq. (1)). After that the transform can be formulated as follows ([4]):

$$\rho_i = \log \left(\sqrt{(x_i - O_x)^2 + (y_i - O_y)^2} \right), \quad 0 \leq \rho_i \leq \rho_{\max} \quad (4)$$

$$\omega_i = \arctan \left(\frac{y_i - O_y}{x_i - O_x} \right), \quad 0 \leq \omega_i \leq 2\pi. \quad (5)$$

Equation (4) allows to receive the logarithmic-radius coordinates, and the equation (5) the angular ones.

5. The Log–Pol–F and the UNL–F descriptors

For the case of Log–Pol–F ([4]) and UNL–F ([5]) descriptors the idea is to add a two-dimensional Fourier transform as a next stage to the original algorithms. It is a very interesting solution, because this transform is invariant to circular shift, which gives the achieved representation additional robustness to rotation and scaling. Moreover, the Fourier transform has a well known property of robustness to noise. The above advantages of described approaches make them ones of the most effective methods in shape representation.

6. mUNL descriptor

As it was mentioned in the last section the application of the Fourier transform gives as a result a very efficient approach. In fact, the UNL–F transform is invariant to rotation, scaling, translation, noise, discontinuities in the contour and selection of the starting point during tracing the boundary. It is not robust only to heterogeneous noise and occlusion. The second problem occurs when part of an object is not visible or new parts appear in the object (e.g. as an effect of the weather conditions or problems during image acquisition). It is in fact the most difficult problem nowadays.

In [6] a solution to this problem was proposed. It was noticed that another method of deriving the origin of the polar transform makes it possible to achieve robustness to occlusion. Such replacement is useful, because when the contour is affected by occlusion, the centroid changes its location in comparison to

the unaffected object. As a new method the IPMinD (*Iterative Polar Minima Derivation*) was used. This method tries to find a pre-assumed number of points lying the most inside the contour and uses them to calculate the new origin. However, this method starts with the calculation of traditional centroid as in equation (1). This time, because of the iterative matter of the latter calculations the superscript (j) is added. The vector of distances is now calculated ([6]):

$$\rho_i = \sqrt{(x_i - O_x^{(j)})^2 + (y_i - O_y^{(j)})^2}, \quad i = 1, \dots, n.$$

The d (the number of points for later derivation of the origin) smallest elements in the vector of distances are selected and denoted as ([6]):

$$\rho_1^{(j+1)}, \rho_2^{(j+1)}, \dots, \rho_d^{(j+1)}.$$

The Cartesian coordinates corresponding to them are denoted as ([6]):

$$(x_1^{(j+1)}, y_1^{(j+1)}), (x_2^{(j+1)}, y_2^{(j+1)}), \dots, (x_d^{(j+1)}, y_d^{(j+1)})$$

The origin of the polar transform at the $j + 1$ iteration will be ([6]):

$$O^{(j+1)} = (O_x^{(j+1)}, O_y^{(j+1)}) = \left(\frac{1}{d} \sum_{i=1}^d x_i^{(j+1)}, \frac{1}{d} \sum_{i=1}^d y_i^{(j+1)} \right).$$

The above process is reiterated until the difference between the derived point and that derived at the former iteration will be lower than a given threshold. Then, the last derived values are taken as the origin of the polar transform ([6]):

$$O = (O_x, O_y) = O^{(j+1)}.$$

Instead of the centroid the achieved point is used as an origin of the polar transform in the UNL transform (eq. (2)–(3)).

As it was mentioned the reason for modifying the UNL descriptor was the necessity of solving the problems of occlusion and heterogeneous noise. Unfortunately, in order to match the achieved representation, rather on a local than global level, the special algorithm had to be proposed. The PPMA (*Partial Point Matching Algorithm*) allows to determine the similarity on the very low level - single points in mUNL representations ([6]).

The combination of the two above algorithms gives the approach robustness to all shape deformations. Normalization results in invariance to scaling. Derivation of the transform according to a particular point as an origin gives the invariance to translation. The parametric curves (or interpolation) solve the problem of discontinuities. The usage of the new method for finding the origin in combination with the mentioned matching algorithm solves the problems of rotation, noise, starting point selection and occlusion.

7. PDH descriptor

The PDH (*Point Distance Histogram*) was for the first time presented in [7] and is based on combination of polar coordinates and the histogram. It was developed to indicate small differences between shapes (e.g. in blood cells). As in the case of the mUNL descriptor the choice of the method of calculating the centre of the object O is unconstrained.

Firstly, the polar coordinates are derived and put into two vectors: Θ^i for angles and P^i for radii ([7]):

$$\rho_i = \sqrt{(x_i - O_x)^2 + (y_i - O_y)^2},$$

$$\theta_i = \arctan\left(\frac{y_i - O_y}{x_i - O_x}\right).$$

The obtained values are converted into the nearest integers ([7]):

$$\theta_i = \begin{cases} \lfloor \theta_i \rfloor, & \text{if } \theta_i - \lfloor \theta_i \rfloor < 0.5 \\ \lceil \theta_i \rceil, & \text{if } \theta_i - \lfloor \theta_i \rfloor \geq 0.5 \end{cases}.$$

The elements in Θ^i and P^i are now rearranged, according to the increasing values in Θ^i , and denoted as Θ^j , P^j . If some elements in Θ^j are equal, only the one with the highest corresponding value P^j is selected. That gives a vector with at most 360 elements, one for each integer angle. For further work only the vector of radii is needed. We denote it as P^k , where $k = 1, 2, \dots, m$ and m is the number of elements in P^k (m is less or equal to 360). Now, the normalization of elements in vector P^k is performed ([7]):

$$M = \max_k \{\rho_k\}, \quad \rho_k = \frac{\rho_k}{M}.$$

The elements in P^k are assigned to bins in histogram (ρ_k to l_k) ([7]):

$$l_k = \begin{cases} r, & \text{if } \rho_k = 1 \\ \lfloor r\rho_k \rfloor, & \text{if } \rho_k \neq 1 \end{cases}.$$

The possible bin index ranges from 1 to r , where r is a predetermined number of bins. The next step is the normalization of the values in bins, according to the highest one ([7]):

$$S = \max_k \{l_k\}, \quad l_k = \frac{l_k}{S}.$$

The final histogram representing a shape can be formally rewritten as a function $h(l_k)$ ([7]):

$$h(l_k) = \sum_{k=1}^m b(k, l_k),$$

where ([7]):

$$b(k, l_k) = \begin{cases} 1, & \text{if } k = l_k \\ 0, & \text{if } k \neq l_k \end{cases}.$$

The PDH descriptor is invariant to rotation, scaling, translation and to the problem of selection of the starting point. Its most important advantage is the ability of indicating small differences between similar objects.

8. Conclusions

The paper presented a few popular and efficient polar contour shape descriptors, from the simplest to more sophisticated ones. The first described and also the oldest one – Centroidal Distance – is invariant only to translation and scaling as well as the Log–Pol (logarithmic–polar) transform. The UNL descriptor is additionally robust to discontinuities in the boundary and to some level of noise. It is also more distinguishable for various shapes. Better results can be achieved after application of the Fourier transform to the polar one. The Log–Pol–F (logarithmic–polar–Fourier) as well as the UNL–Fourier transforms are invariant to affine planar transforms, noise, discontinuities in the contour and selection of the starting point. The mUNL descriptor is also robust to some level of occlusion (when used in combination with PPMA for matching). The PDH descriptor is invariant to the mentioned affine transforms and robust to the problem of the starting point. However, it is very useful when we are trying to indicate small differences between shapes.

References

- [1] Jaggi S., Karl W. C., Mallat S. G., Willsky A. S., *Silhouette recognition using high resolution pursuit*, Pattern Recognition 32(5) (1999).
- [2] Rauber T.W., Steiger-Garcia A.S., *2-D form descriptors based on a normalized parametric polar transform (UNL transform)*. Proc MVA'92 IAPR Workshop on Machine Vision Applications (1992).
- [3] Frejlichowski D., *Problem braku części sylwetki w rozpoznawaniu obrazów konturowych z użyciem przekształcenia do układu biegunowego*, Materiały VIII Sesji Naukowej Informatyki (2003).
- [4] Luengo-Oroz M., Angulo J., Flandrin G., Klossa J., *Mathematical morphology in polar-logarithmic coordinates. Application to erythrocyte shape analysis*, Lecture Notes in Computer Science 3523 (2005).
- [5] Rauber T.W., *Two-dimensional shape description*, Technical Report: GR UNINOVA-RT-10-94, Universidade Nova de Lisboa (1994).
- [6] Frejlichowski D., *An algorithm for binary contour objects representation and recognition*, Lecture Notes in Computer Science 5112 (2008).
- [7] Frejlichowski D., *Shape representation using point distance histogram*, Polish Journal of Environmental Studies 16(4A) (2007).