Contour Compression Using Centroid Method

ANDRZEJ DZIECH [1] REMIGIUSZ BARAN [2] ALI UKASHA [3]
[1] Department of Telecommunications, AGH University of Science and Technology, POLAND
[2] Department of Electronics and Intelligent Systems, Kielce University of Technology, POLAND,

[3] University of Computer Technology and Telecommunications in Kielce, POLAND,

Abstract: - A new approach for contour data compression is presented in the paper – Cartesian co-ordinates of an input contour are processed in such a way that final contour is represented by one-dimensional set of distances with constant, regular angle between them. The selection of vertices and the final algorithm for contour compression are described. Comparison of proposed method with the Ramer algorithm was also performed. For comparison the mean square error and SNR criteria were used. Finally, results of the experiments and advantages and drawbacks of proposed method are discussed.

Key-Words: - contour representation, contour compression, polygonal approximation, the Ramer algorithm.

1 Introduction

Contours can be treated as important image structures for both coding and recognition. Contour processing, including contour compression is widely used in such common applications as topographic or weather maps preparation, character recognition, processing of medical images, image compression etc. Contour processing is also required in computer vision e.g. robot guidance or non-contact visual inspection.

One of the main approaches to the problem of contour compression is the time domain approach. The time domain methods are mainly based on the polygonal approximation scheme [5, 6, 7]. One of the most appreciated examples of such scheme is the Ramer algorithm [2], which uses the maximum distance of the curve from the approximating polygon as the fit criterion.

Most of the contour approximation methods use Cartesian representation. However, there are also schemes and applications where polar or Freeman's (also generalised) [1, 3] chain coding representations are required.

2 The algorithm (in general)

The input contour for the proposed method is obtained from 256 x 256 grey-scale images by using the SSPCE contour extraction procedure [4].

The input contour is represented by x and y vectors of Cartesian co-ordinates. The algorithm starts with finding the center of the input contour mass called the reference point $O = (x_m, y_m)$. Co-ordinates x_m, y_m are defined as follows

$$x_m = \frac{1}{N}\sum_{i=0}^{N-1} x_i \qquad y_m = \frac{1}{N}\sum_{i=0}^{N-1} y_i \tag{1}$$

where:

N - number of contour vertices;
x_m- mean value of the x vector;
y_m- mean value of the y vector.

The input contour is then shifted by x_m and y_m in both axes directions

$$x_i = x_i - x_m,\ y_i = y_i - y_m \tag{2}$$

Centroid of the shifted contour is placed at (0,0) point, and therefore further computations are vastly simplified. Next, distances r between (0,0) point and shifted contour line are calculated. For that purpose, straight lines are passed through the (0,0) point. Slopes of these lines depend on earlier assignment of an angle between them. The angle between the contour lines is represented by the input parameter Φ. The Φ value is assigned with respect to another input parameter I - accuracy of the procedure. The relation between parameters I and Φ is as follows

$$\Phi = \frac{\pi}{2^{I+1}} \tag{3}$$

When accuracy $I = 0$ it means that $\Phi = \pi / 2$ and two perpendicular straight lines are passed. In case of $I = 7$ the $\Phi = \pi / 256$ and 256 straight lines are passed. This is illustrated in Fig. 1 for $I = 0$ and $I = 1$.

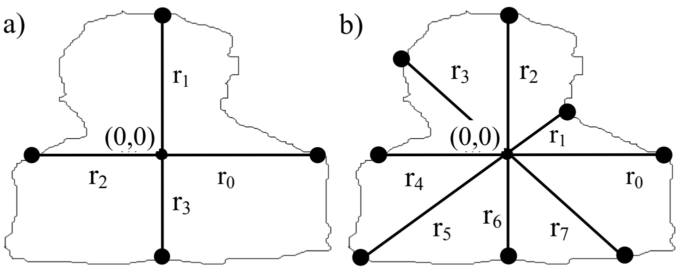

Fig. 1. Selection of output vertices a) $I = 0$ and b) $I = 1$.

Selected vertices are fully determined by the accuracy of the procedure and sequence of r distances. Therefore, representation of the approximated contour can be one-dimensional. This is the advantage of the presented method.

Distances r are given by the following equation

$$r_i = \sqrt{x_i^2 + y_i^2} \tag{4}$$

where:

x_i, y_i - co-ordinates of selected vertices.

Flowchart of the algorithm is depicted in Fig 2.

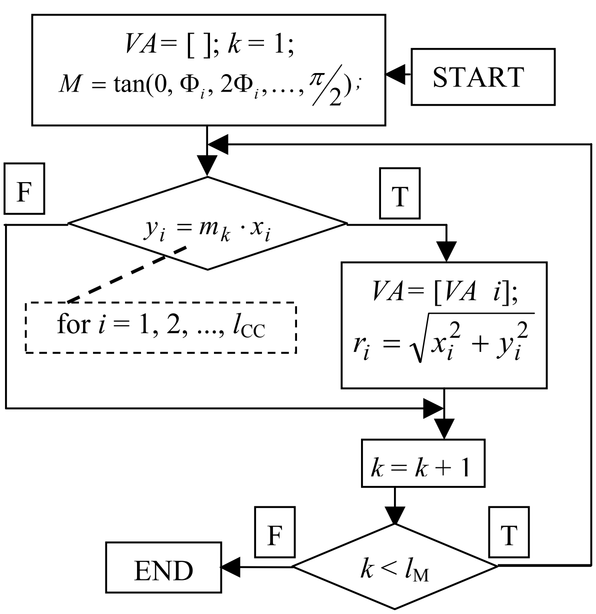

Fig. 2. Flowchart of the algorithm.

where:

VA - sequence of the output vertices;
M - vector of slopes m;
l_M - length of vector M;
x_i, y_i - co-ordinates of the input contour points;
l_{CC} - length of the input contour;
k - counter.

3 Applied measures

Presented method is related to the data compression problem. To evaluate its compression ability, the following compression ratio was introduced

$$CR = \frac{(B_{CC} - B_{AC})}{B_{CC}} \cdot 100\% \tag{5}$$

where:

B_{CC} - total number of bits required for the input contour;
B_{AC} - total number of bits required for the output contour.

The B_{CC} value depends on length of the input contour and the maximum values of x and y. The value of B_{CC} for contours extracted from 256 x 256 images can be calculated as follows

$$B_{CC} = 2 \cdot l_{cc} \cdot 8 \text{ bits}$$

The full information required for contour reconstruction consists in collecting the following parameters: Φ , r and co-ordinates of the reference point O.

Such output contour representation is sufficient only for regular contour shapes, as in Fig. 1. More complicated contours require additional information. Let's consider contour presented in Fig. 3.

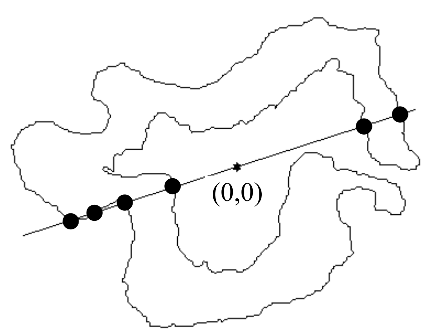

Fig. 3. Selection of output vertices in case of generalized contour shapes.

As it is seen, one straight line can mark more then two vertices. The position of each marked vertex in reference to the (0,0) point and the order of vertices are now necessary.

The total number of bits B_{AC} of the output contour representation contains:

- 3 bits for the accuracy of the procedure,
- 16 bits for co-ordinates of the centroid,
- 9 bits for distances r.

In addition for generalised shapes we need:

- 1 bit to indicate the position of each vertex,
- b bits to indicate the order of vertices.

The number of bits b is calculated in the following way

$$b = nb_{\text{maximum\_coded\_value}} \cdot nv$$

where:

$nb_{\text{maximum\_coded\_value}}$ - number of bits required for maximum coded value;

nv - number of selected vertices.

The mean square error (MSE) and signal-to-noise ratio (SNR) criteria were used as measures of quality of approximation. The MSE is defined by the following equation

$$MSE = \frac{1}{l_{CC}} \sum_{i=1}^{l_{CC}} d_i^2 \qquad (6)$$

where:

d_i - distance between line of the input contour and vertex i.

The SNR is defined by the following formula

$$SNR = -10 * \log_{10}\left(\frac{MSE}{VAR}\right) \qquad (7)$$

where:

VAR - variance of the input sequence.

From the practical point of view the values of MSE and SNR can not exceed the 4,0 and 30 dB, respectively. Otherwise, the details of contours are eliminated and level of introduced distortion can not be accepted.

4 Results of experiments

To visualise the experimental results a set of two test contours (one generalised and one regular) was selected. Selected contours are depicted in Fig. 4.

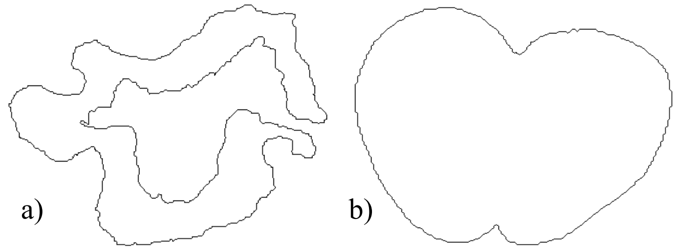

Fig. 4. Test contours a) Serpent, b) Apple.

Selected results of the compression of the test contours are illustrated in Fig. 5 and Fig. 6. Fig. 5 shows the results of compression performed for the test contour Serpent. Fig. 5a presents the reconstruction obtained when the additional information is not used.

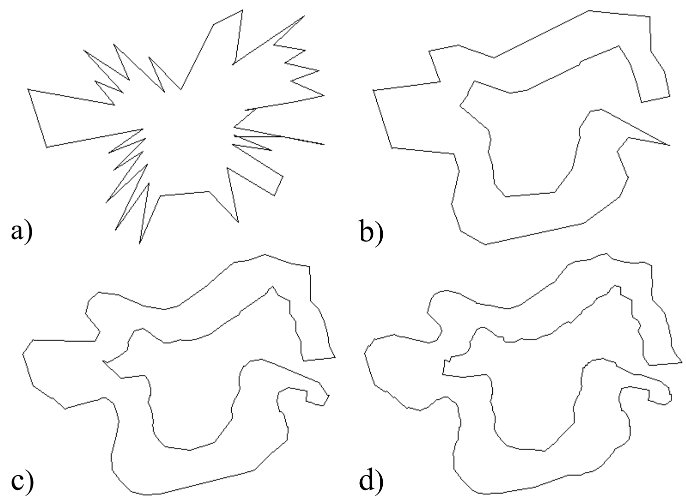

Fig. 5. Results of compression of the test contour Serpent.

	AI	I	MSE	SNR	CR [%]
a)	not used	3	-	-	97,61
b)	used	3	32,39	22,09	96,45
c)	used	4	5,33	29,93	90,48
d)	used	5	1,14	36,63	79,30

AI – additional information

As it is seen, the maximum value of compression ratio, which can be obtained when contours of general shapes are processed, is less than 90%. It is also seen that maximum useful accuracy I is equal to 4. This however means that complexity of such compression process is very small and it can be done very fast.

The centroid method gives much better results, in sense of compression ratio and level of introduced distortion, for contours of regular shape.

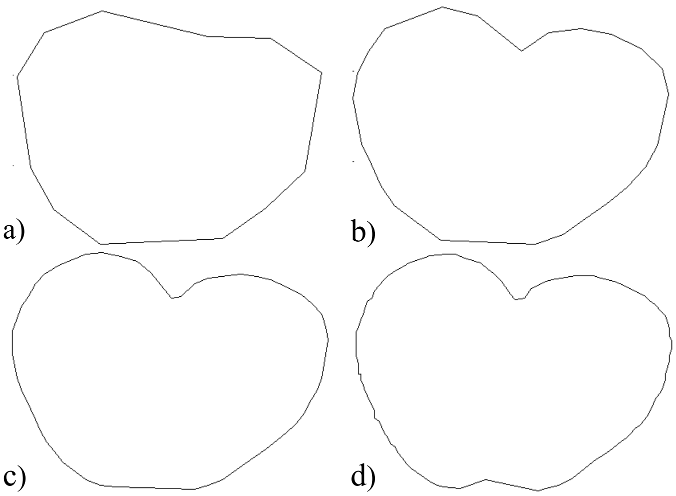

Fig. 6. Results of compression of the test contour Apple.

	AI	*I*	*MSE*	*SNR*	*CR* [%]
a)	not used	2	12,93	22,18	97,89
b)	not used	3	2,61	29,14	95,86
c)	not used	4	2,11	30,05	91,67
d)	not used	5	0,59	35,61	83,74

AI – additional information

When the accuracy of the procedure is assigned to 4, the compression ratio is much greater than 91% and values of *MSE* and *SNR* can be fully accepted.

Comparison between compression abilities of the proposed method and the Ramer algorithm can be done from Figs. 7 and 8.

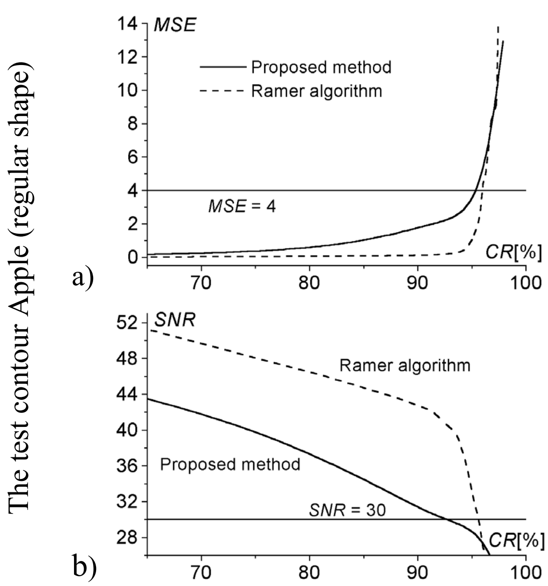

Fig. 7. The cenroid method versus the Ramer algorithm a) *MSE* versus *CR*, b) *SNR* versus *CR*.

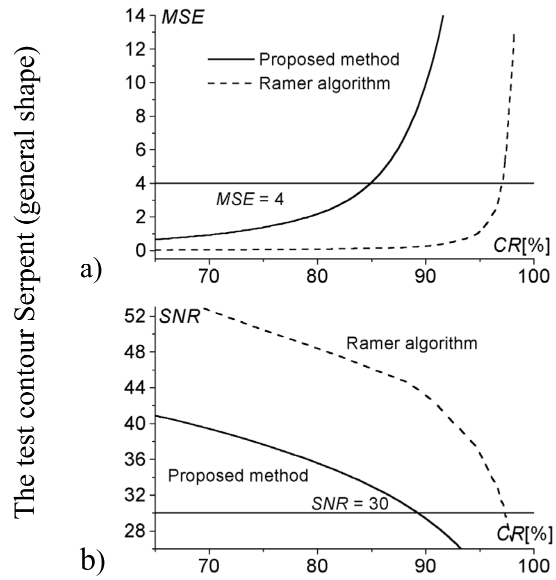

Fig. 8. The centroid method versus the Ramer algorithm a) *MSE* versus *CR*, b) *SNR* versus *CR*.

Presented plots confirm that the compression abilities of the centroid method are much better in case of general shapes. Fig. 7 shows that the compression ratio for this type of contours can be even greater than 96%. Also 96% of *CR* can be obtained when the Ramer algorithm is chosen, but the complexity of the proposed method is much less.

Fig. 8 shows however that the Ramer algorithm can be more useful for contours of general shapes. The maximum acceptable compression ratio for this algorithm is about 10% greater then for the proposed method.

5 Conclusions

A new method for contour compression is presented in the paper. The main advantage of this method is "one-dimensional" representation of the final contour. It was shown that proposed method, especially for the contours of regular shapes, has very good compression abilities. Compression ratio for the regular shapes can exceed 96% - similarly as in case of the Ramer algorithm. However, the complexity of the centroid method is much less than that of Ramer. Therefore, the proposed method seems to be also very useful for contours of general shapes. Although the Ramer compression scheme ensures greater values of compression ratio, the new proposed method is much faster.

Acknowledgement

This work was supported by the Polish State Committee for Scientific Research (KBN) under the Grant No. 4T11D00525 (period of support: 2003-2005).

Reference:

[1] H. Freeman, Techniques for The Digital Computer Analysis of Chain Encoded Arbitrary Plane Curves, *Proc. of the National Electrician Conference*, 1961, pp. 421-432.

[2] U. Ramer, An Iterative Procedure for the Polygonal Approximation of Plane Curves, *Computer Graphics and Image Proc., Academic Press*, 1972, pp.244-256.

[3] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall International, 1989.

[4] A. Dziech, W. S. Besbas, Fast Algorithm for Closed Contour Extraction, *Proc. of the Int. Workshop on Systems, Signals and Image Processing*, Poznań, Poland, 1997, pp. 203-206.

[5] B. Tibken, A. Dziech, R. Baran, Spatial Methods of Contour Approximation, *Int. Conf. on Computational Intelligence for Modeling, Control and Automation*, Vienna, Austria, 2003, pp. 354-362.

[6] R. Baran, W. Dziech, Tangent Method of Contour Approximation, *Int. Workshop on Multimedia Communication and Services*, Kielce, Poland, 2003, pp. 11-16.

[7] A. Dziech, R. Baran, The Tangent Method Algorithms for Contour Approximation, *Int. Workshop on Multimedia Communication and Services*, Kraków, Poland, 2004, pp. 11-16.