Computation of normals at the boundaries of Two Dimensional Objects from their Medial Axis Transforms

Jayanta Mukherjee*, Aswatha Kumar M $,P.P.Das * and B.N Chatterji#
* Department of Comp. Science and Engineering, I.I.T, Kharagpur-721 302.
$ Dept. of Electronics and Communication Engg, BDT College of Engg. ,
Dawangere, Karnataka.
Electronics and ECE Department, I.I.T, Kharagpur-721 302.

Abstract

Though medial axis transform (MAT) is well known for object representation, its use in different kinds of computations remains unexplored till date. One of the main reasons is the popularity of other spatial representations such as quad-tree in 2D and octree in 3D for better data compression and ordered storage and retrieval facility etc. In this paper an algorithm has been proposed for computation of normals at the boundaries of 2D objects based on their medial axis transforms. In this technique, there is no requirement of linking boundary points during the computation compared to other existing techniques. The added advantage in the computation is that the computation can be restricted purely in the integer domain.

I. Introduction

Medial axis transform (MAT) for 2-D and 3-D binary images data have been introduced by Rosenfeld and Pfaltz [1]. As the maximal blocks of MAT are also termed as 'circles' in 2D and 'spheres' in 3D the representation is also called as medial circle representation (MCR) in 2D and as medial sphere representation (MSR) in 3D. It has been shown that using generalized octagonal distances[2] good MCR (as approximations to Euclidean Circles) can be obtained for binary images. The representation is found to be useful for the computation of linear geometric transformations like translation, rotation and scaling[3] and computation of cross-sections for three-dimensional objects [4]. Interestingly, such computations are difficult to perform using hierarchical schemes like quad-tree or octree which are other popular alternatives for representing binary data. Medial axis transforms have also been used in discrete shading techniques [5] for rendering 3D objects and it is found that the quality of the rendered images are better than those obtained from other spatial representations such as voxel and octree.

In this paper, the computation of normals at boundary points of a 2D binary object from MCR has been discussed. Similar methodology can be extended in 3D also. But, as this method is computationally intensive and it gives only a rough estimate of the normals, the present work is restricted to 2D only. In 3D, instead of computing normals, similar concept has been used for discrete shading which has been reported elsewhere [5].

In [6] we have discussed about Medial Circle Representation using Octagonal Distances. We follow the same conventions for representing an octagonal distance (by a neighborhood sequence of type 1 (4-neighborhood) or type 2 (8-neighborhood)). For example an octagonal distance denoted by {1,2} implies the metric defined by the neighborhood sequence of type 1 and type 2 (alternately). The computation of distance transform with an octagonal distance and the computation of the vertices of the medial circles (which are found to be convex polygons) defined by the respective distance function are also discussed in that paper [6].

II. Computation of Normals

The principle behind the computation of normals at the boundaries of 2D objects can be explained with the help of Figure 1. Let P be a point on the contour of a 2D object. Let M be the medial circle touching the contour at P and let the center of M be O_c. Then $\overline{PO_c}$ forms the inward normal at point P. The same concept could easily be extended in 3D where inward surface normal at any boundary point is expressed by the vector drawn from that point to the center of touching medial sphere. However in the present work the experimentations have been carried out in 2D only. The proposed technique requires the list of boundary points of 2D object and the list of medial circles to represent the object.

There are distinct characteristics of this technique in comparison with the existing analytical techniques for computing normal at a boundary point (by regression method [7] around its neighboring point etc.). These are as follows:

1. In an analytical technique, contour tracing is per-

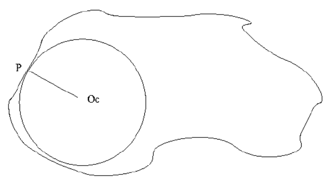


Fig. 1. Schematic diagram for normal computation

formed for determining the adjacent neighboring boundary points. This is also required to determine whether the normal is inward or outward. The proposed technique does not require any such contour following.

2. In the proposed technique the whole computation can be performed in integer domain.

It may be noted that there are also techniques for computing tangents (an equivalent of normal computation) of the digital curves in the discrete domain. In the seventies (of the last century) these techniques were proposed by several researchers, more notably by Rosenfeld and Johnston [8] and Bennet and MacDonald [9]. However all these techniques require the contour tracing prior to computing the tangents (as well as normals) at the points in the digital curve. The techniques also heavily suffer from the computation errors due to local perturbation in the neighboring points. The proposed technique captures the global shape information in a better way in the form of the touching medial circles totally contained in the pattern.

In the following section, the algorithm for computing normal at contour points of a 2D binary object is presented. Experiments are carried out to judge the quality of normal computation which has been discussed in the subsequent section.

A. *Algorithm for Normal Computation*

Let C be the set of contour points of a 2D object and M be the set of medial circles representing the object. The algorithm for computing normals at each boundary point is presented below:

Algorithm : Normal-Computation

Input : C, M

Output : Normals at the boundary points.

Begin

For each boundary point $p \in C$, perform the following operations.

Step 1: Find the set of medial circles S close to p such that

$$\forall s \in S, \mid d(p, o_s) - r_s \mid < E_\theta$$

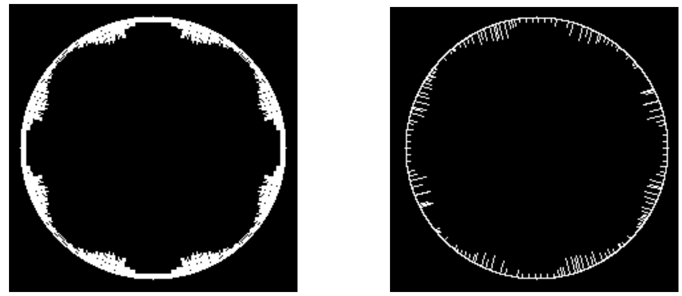
where E_θ is the threshold for declaring s close to p and o_s and r_s are the center and radius of the medial circle s. $d(p, o_s)$ is the distance function used in MCR.

Step 2: Normal at P is expressed by the unit vector along $\vec{N}_p$, where

$$\vec{N}_p = \sum_{\forall s \in S}(\overline{PO}_s)$$

.

End (Normal-Computation)



(a) (b)

Fig. 2. Normal computation at the contour points for the image circle for B={1,1,2} (a) Dense normal map, (b) Sparse normal map

The above algorithm works in two stages. First a set of medial circles close to the boundary point is formed. It is known that for each boundary point there exists at least one medial circle which touches that point. In the next stage, the resultant normal vector is computed from the set of nearer medial circles.

B. *Experimental Results*

Experimentations have been carried out to judge the quality of the resulting normal vector obtained by the proposed technique against the true analytical values of the normals at the respective boundary points. For this purpose three regular 2D objects have been considered - a circle, a rectangle, and a square. Figure 2(a) shows the normals computed at the contour points of the circle by using digital distances for B = {1,1,2}. Note that normals at all the contour points can be computed by this technique. To make the visual presentation better, the sparse normal map on the contour (at some selective points) is presented in the Figure 2(b). The value of E_θ is kept 1 in this case.

To show the effect of different MCRs (with different digital octagonal distance) Figure 3 typically presents (a sparse normal map) the normal computation of circle for all the digital distances whose length of neighborhood sequences B is less than or equal to 4.

Next aim of the experiment has been to provide a quantitative measure for judging the quality of the normal computation. In Figure 4 analytical normal maps have been shown for circle, square and rectangle respectively. Let $\vec{n}_p$ be the computed value of the unit normal vector at a point $p \in C$ using the proposed technique. Let $\vec{m}_p$ be the analytical value of unit normal vector at a point $p \in C$ (refer

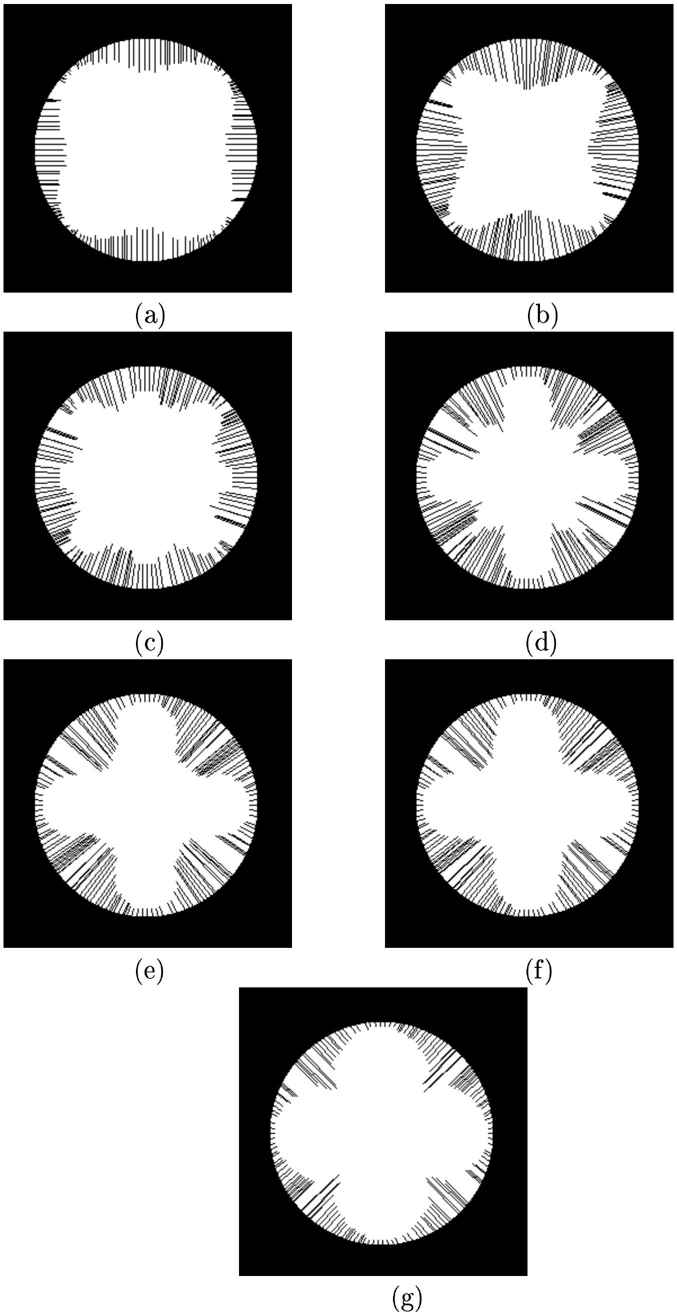


Fig. 3. Normal computation for an image CIRCLE using digital circles for different distance functions
(a) for B={1}, (b) for B={1,1,1,2}, (c) for B={1,1,2}, (d) for B={1,2}, (e) for B={1,2,2}, (f) for B={1,2,2,2}, (g) for B={2}

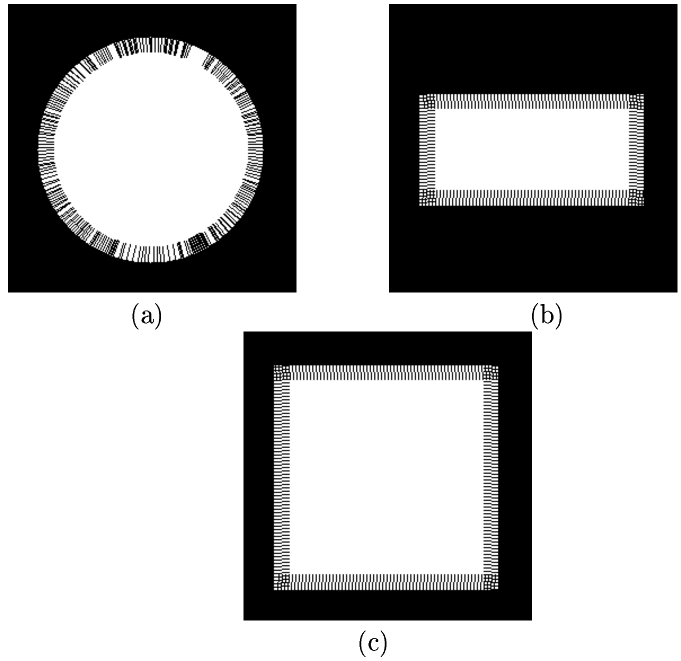


Fig. 4. Analytical normal maps: (a) for circle, (b) for rectangle, (c) for square

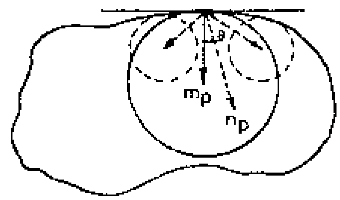


Fig. 5. Computation of errors between analytical and computed normals

Figure 5). For example, for a circle of radius R with center at (x_c, y_c), $\vec{m}_p$ at a point $p(x, y)$ is expressed as

$$(\frac{(x_c - x)}{R}, \frac{(y_c - y)}{R}) / \parallel (\frac{x_c - x}{R}, \frac{y_c - y}{R}) \parallel$$

where $\parallel . \parallel$ denote the magnitude of the corresponding vector. Then an error measure E_n has been defined here as

$$E_n = \sum_{\forall p \in C} \mid \vec{n}_p \cdot \vec{m}_p \mid / \mid C \mid$$

If the proposed technique is unable to compute $\vec{n}_p$ at any point $P \in C$, then the contribution at that point towards the error measure (E_n) is not considered and subsequently the point p is excluded from the set C.

In the Table I the errors are computed for normal computation using MCR with digital distances. It can be observed that B={1,1,2}, B={1,2} and B={1,1,1,2} perform well in comparison with other distances.

The technique has been used for normal computation of different binary images. The normal maps are shown in Figure 6 using digital octagonal distance for B = {1,1,2}.

TABLE I

Percentage Normal computational errors using MCR with digital metric (for E_θ=1)

B	Circle	Rectangle	Square
{1}	4.28	0.73	0.48
{1,1,1,2}	0.99	1.70	1.27
{1,1,2}	0.38	2.59	2.04
{1,2}	0.23	4.97	4.38
{1,2,2}	1.03	7.69	6.74
{1,2,2,2}	1.65	8.85	7.75
{2}	3.33	1,2.18	11.89

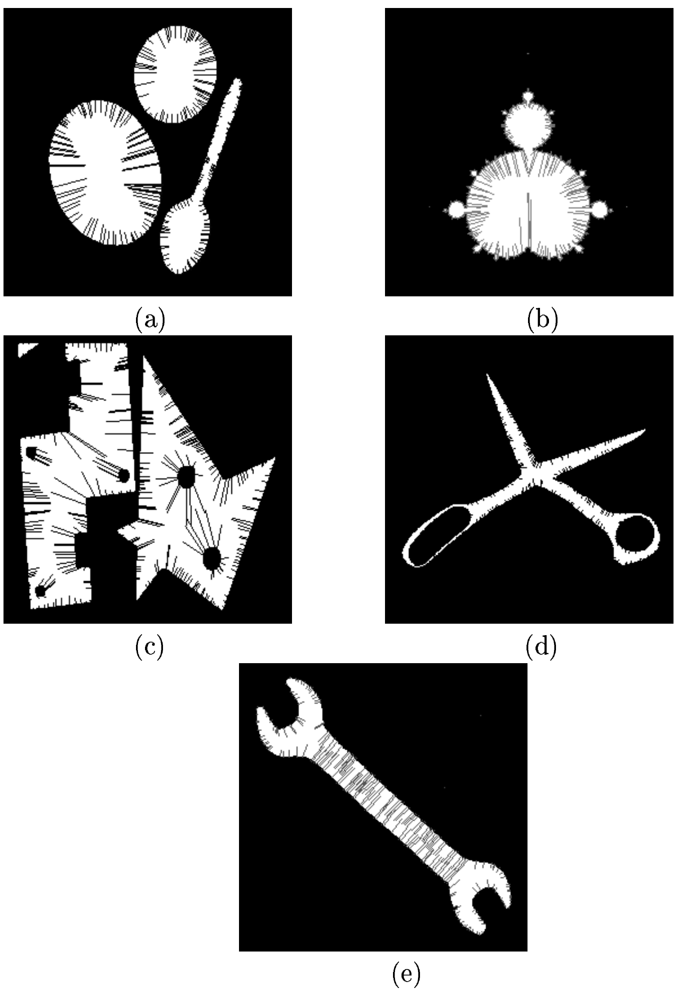


Fig. 6. Normal maps of different images using digital octagonal distance B = {1,1,2}

III. Conclusion

In this paper we have shown how medial axis transforms (MATs) of binary images can be effectively used for computing the normals along the boundary points of 2D objects. This is another interesting example of the application of MAT for computing the image features. It may be noted, earlier we have also reported other applications such as the linear transformations of binary objects [3], discrete shading [5] and computation of cross-sections [4] of 3D objects. We have further illustrated that the theoretical performance of the digital distance as shown in [2] [6] do match in the practical experiments. The method could be used easily in detecting corners and segmenting the boundary curves of the 2D objects.

References

[1] A. Rosenfeld and J. L. Pfaltz. Distance functions in digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[2] P. P. Das and B. N. Chatterji. Octagonal distances for digital pictures. *Information Sciences*, 50:123–150, 1990.

[3] M. Aswatha Kumar, Jayanta Mukherjee, B. N. Chatterji, and P. P. Das. Representation of 2D and 3D binary images using medial circles and spheres. *Intl. Jrnl. Pattern Recognition and Artificial Intelligence*, 10:365–387, 1996.

[4] Jayanta Mukherjee, M. Aswatha Kumar, B. N. Chatterji, and P. P. Das. Fast computation of cross-sections of three-dimensional objects. *Pattern Recognition Letters*, 21:605–613, 2000.

[5] Jayanta Mukherjee, M. Aswatha Kumar, B. N. Chatterji, and P. P. Das. Discrete shading of three dimensional objects from medial axis transform. *Pattern Recognition Letters*, 20:1533–1544, 1999.

[6] Jayanta Mukherjee, M. Aswatha Kumar, B. N. Chatterji, and P. P. Das. On approximating euclidean metrics by digital distances in 2D and 3D. *Pattern Recognition Letters*, 21:573–582, 2000.

[7] D.M. Tsai and M.F. Chen. Curve fitting approach for tangent angle and curvature measurements. *Pattern Recognition*, 27(5):699–711, 1994.

[8] A. Rosenfeld and R. Johnston. Angle detection on digital curve. *IEEE trans. of Computers*, 22:805–878, 1973.

[9] J.R. Bennet and J.S. MacDonald. On the measurement of curvature in a quantized environment. *IEEE trans. of Computers*, 24:803–820, 1975.