

# Computing the Eccentricity Transform of a Polygonal Shape

Walter G. Kropatsch, Adrian Ion, and Samuel Peltier\*

Pattern Recognition and Image Processing Group,  
Faculty of Informatics, Vienna University of Technology, Austria  
{krw,ion,sam}@prip.tuwien.ac.at

**Abstract.** The eccentricity transform associates to each point of a shape the distance to the point farthest away from it. The transform is defined in any dimension, for open and closed manifolds, is robust to Salt & Pepper noise, and is quasi-invariant to articulated motion. This paper presents an algorithm to efficiently compute the eccentricity transform of a polygonal shape with or without holes. In particular, based on existing and new properties, we provide an algorithm to decompose a polygon using parallel steps, and use the result to derive the eccentricity value of any point.

**keywords:** eccentricity transform, distance transform, polygon.

## 1 Introduction

To extract from a set of images the information required for a specific task, a frequently used design pattern is to repeatedly transform the input image while gradually moving from the low abstraction level of the input data to the high abstraction level of the output data. The purpose is to have a reduced amount of (important) information at the higher abstraction levels. One class of such transforms that is applied to 2D shapes, associates to each point of the shape a value that characterizes in some way its relation to the rest of the shape, e.g. the distance to some other point of the shape.

Examples of such transforms include the well known distance transform [1], which associates to each point of the shape the length of the **shortest** path to the border, the Poisson equation [2], which can be used to associate to each point the **average** time to reach the border by a random path (average length of the random paths from the point to the boundary), and the eccentricity transform [3] which associates to each point the length of the **longest** of the shortest paths to any other point of the shape. Using the transformed images one tries to come up with an abstracted representation like the skeleton [4] or shock graph [5] build on the distance transform, which could be used in e.g. shape classification or retrieval.

Minimal path computation [6] as well as distance transform [7] are used in 2D and 3D image segmentation.

---

\* Supported by the Austrian Science Fund under grants P18716-N13 and S9103-N04.

The eccentricity transform (ECC) has its origins in the graph based eccentricity [8,9]. It has been defined in the context of digital images in [3,10], where properties and robustness have been shown, and it was applied in the context of shape matching in [11]. The eccentricity transform can be defined for discrete objects of any dimension, closed (e.g. typical 2D binary image) or open sets (surface of an ellipsoid), and for continuous objects of any dimension (e.g. 3D ellipsoid or the 2D surface of the 3D ellipsoid, etc.).

For the case of discrete shapes, a naive algorithm and a more efficient one for 2D shapes without holes, have been presented in [3]. For simply connected shapes on the hexagonal and dodecagonal grid, an efficient algorithm was given in [12]. Regarding continuous shapes, a detailed study has been made for the case of an ellipse, and some preliminary properties regarding rectangles, and a class of elongated convex shapes, have been given [13]. An algorithm for finding the eccentric vertices (furthest points) for the vertices of a simple polygon was given in [14].

This paper presents an algorithm for efficiently computing the eccentricity transform of a polygonal shape. First, the shape is decomposed into patches associated to corner points then these patches are used to compute the eccentricity.

Section 2 gives a short recall of the eccentricity transform and gives the main properties relevant for this paper. Section 3 briefly recalls existing algorithms. Sections 4 and 5 present the proposed algorithm, followed by discussion in Section 6 and possible extensions in Section 7. Section 8 concludes the paper and gives an outlook of the future work.

## 2 Recall ECC

In this section basic definitions and properties of the eccentricity transform are introduced following [3,11]. Let the shape  $\mathcal{S}$  be a closed set in  $\mathbb{R}^2$  and  $\partial\mathcal{S}$  be its border<sup>1</sup>. A path  $\pi$  is the continuous mapping from the interval  $[0, 1]$  to  $\mathcal{S}$ . Let  $\Pi(\mathbf{p}_1, \mathbf{p}_2)$  be the set of all paths between two points  $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{S}$  within the set  $\mathcal{S}$ . The geodesic distance  $d(\mathbf{p}_1, \mathbf{p}_2)$  between two points  $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{S}$  is defined as the length  $\lambda$  of the shortest path  $\pi$ , such that  $\pi \in \Pi(\mathbf{p}_1, \mathbf{p}_2)$ , more formally

$$d(\mathbf{p}_1, \mathbf{p}_2) = \min\{\lambda(\pi(\mathbf{p}_1, \mathbf{p}_2)_{\pi \in \Pi})\} \text{ where } \lambda(\pi(t)) = \int_0^1 |\dot{\pi}(t)| dt \quad (1)$$

where  $\pi(t)$  is a parametrization of the path from  $\mathbf{p}_1 = \pi(0)$  to  $\mathbf{p}_2 = \pi(1)$ .

The eccentricity transform of  $\mathcal{S}$  can be defined as,  $\forall \mathbf{p} \in \mathcal{S}$

$$ECC(\mathcal{S}, \mathbf{p}) = \max\{d(\mathbf{p}, \mathbf{q}) | (\mathbf{q} \in \mathcal{S})\} = \max\{d(\mathbf{p}, \mathbf{q}) | \mathbf{q} \in \partial\mathcal{S}\} \quad (2)$$

i.e. to each point  $\mathbf{p}$  it assigns the length of the shortest geodesics to the points farther away from it. In [3] it is shown that this transformation is quasi-invariant to articulated motion and robust against salt and pepper noise (which creates holes in the shape).

---

<sup>1</sup> This definition can be generalized to higher dimensions.

### 2.1 Properties of Eccentric Points

In general, an *extremal point* is a point where a function reaches an extremum (local or global). In the case of the geodesic distance  $d$  on a shape  $\mathcal{S}$  we call an extremal point  $\mathbf{x} \in \mathcal{S}$  a point for which  $\exists \mathbf{p} \in \mathcal{S}$  s.t.  $d(\mathbf{x}, \mathbf{p})$  is a local maximum.

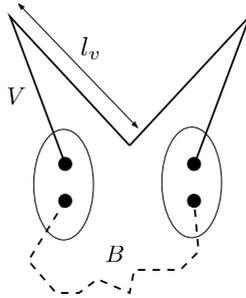
An *eccentric point* of a shape  $\mathcal{S}$  is a point  $\mathbf{e} \in \mathcal{S}$  that is farthest away in  $\mathcal{S}$  for at least one point  $\mathbf{p} \in \mathcal{S}$  i.e.  $\exists \mathbf{p} \in \mathcal{S}$  s.t.  $ECC(\mathcal{S}, \mathbf{p}) = d(\mathbf{p}, \mathbf{e})$ . For a shape  $\mathcal{S}$ ,  $E(\mathcal{S}) = \{\mathbf{e} \in \mathcal{S}\}$  denotes the set of all its eccentric points. The set of eccentric points  $E(\mathcal{S})$  is a subset of the set of extremal points  $X(\mathcal{S}) = \{\mathbf{x}\}$  i.e.  $E(\mathcal{S}) \subseteq X(\mathcal{S})$  (eccentric points are global maxima for  $d$ , while extremal points only local maxima).

Knowing  $E(\mathcal{S})$  can be used to speedup the computation of the  $ECC(\mathcal{S})$ . Instead of computing for each  $\mathbf{p} \in \mathcal{S}$  the length of the geodesics to all the other points of  $\mathcal{S}$  and taking the maximum, one can look at the inverse problem and compute the length of the geodesics from all  $\mathbf{p} \in E(\mathcal{S})$  to all the points of  $\mathbf{p} \in \mathcal{S}$ , and for each  $\mathbf{p}$  just take the maximum. This reduces the number of shortest path computation steps by  $|\mathcal{S}| - |E(\mathcal{S})|$ .

The following properties of extremal and eccentric points are relevant for this paper and concern bounded 2D shapes.

*Property 1.* All eccentric points  $E(\mathcal{S})$  of a shape  $\mathcal{S}$  lie on the border of  $\mathcal{S}$  i.e.  $E(\mathcal{S}) \subseteq \partial\mathcal{S}$ . (*Proof* due to [3]).

*Property 2.* Being an eccentric point is not a local property i.e.  $\forall B \subset \partial\mathcal{S}$  a boundary part (a 2D open and simple curve), and a point  $\mathbf{b} \in B$ , we can construct the rest of the boundary  $\mathcal{S} \setminus B$  s.t.  $\mathbf{b}$  is not an eccentric point of  $\mathcal{S}$ .



**Fig. 1.** Adding missing part  $V$  to existing one  $B$  s.t. no eccentric points lie on  $B$

*Proof.* Let  $l_B$  be the length of  $B$ . We construct  $\mathcal{S} \setminus B$  with the shape of a capital 'V' glued at its endpoints with the endpoints of  $B$ , and the length of the two branches  $l_V > 2 * l_B$  (Figure 1). The obtained shape  $\mathcal{S}$  will have two eccentric point sets clustered around the tops of the two branches of the 'V', a diameter  $\max(ECC(\mathcal{S})) \approx 2 * l_V$ , and no eccentric point will lie on  $B$ . □

*Property 3.* No eccentric points  $E(\mathcal{S})$  of a simply connected shape  $\mathcal{S}$  lie on concave or straight parts of the border of  $\mathcal{S}$  i.e.  $\nexists \mathbf{e} \in E(\mathcal{S})$  s.t.  $\partial\mathcal{S}$  is concave or straight at  $\mathbf{e}$ .

*Proof.* All points at the same distance to a point  $\mathbf{p}$  lie on a circle  $C(\mathbf{p}, r)$ . If the circle is contained in the shape  $\mathcal{S}$  then there are points further away to  $\mathbf{p}$ . A circle  $C$  through a point  $\mathbf{x} \in C$  is partly inside the shape  $\mathcal{S}$  if  $\mathbf{x}$  is on a straight or concave part of the boundary. Thus there exists a point  $\mathbf{q} \in \mathcal{S}$  with  $\mathbf{q} \notin C$  s.t.  $d(\mathbf{p}, \mathbf{q}) > r$ .

A hole can be bypassed in two ways. It partitions the points behind it in two groups: those for which the shortest path passes on one side and, those for which the shortest path passes on the other side of the hole. Shortest paths from both sides meet at the separation curve. Points on the separation curve have on both sides neighbours with smaller distances. Extremal paths never cross the separation curve. Thus, if  $\mathbf{x}$  is on the separation curve then it is an extremal point no matter the curvature of  $\partial\mathcal{S}$  at  $\mathbf{x}$ .  $\square$

*Property 4.* (from Property 3) All eccentric points  $E(\mathcal{S})$  of a simply connected shape lie on convex parts of the border of  $\mathcal{S}$  i.e.  $\forall \mathbf{e} \in E(\mathcal{S}) \Rightarrow \partial\mathcal{S}$  is convex at  $\mathbf{e}$ .

Properties 1, 3 and 4 also apply to extremal points.

*Property 5.* For any shape  $\mathcal{S}$ , all boundary points in convex regions of  $\partial\mathcal{S}$  are external points. (Proof similar to Property 3).

From Properties 1-4 we see that for the case of simple polygons all corner points with angles less than  $180^\circ$  makeup the extremal points, but whether such a corner point is actually an eccentric point or not, can be known only after computing  $ECC(\mathcal{S})$  for the polygon.

For the case of multiply connected 2D shapes, depending on the number and size of the holes, Properties 3 and 4 do not always hold (for a square with a maximum size square hole, all boundary points are eccentric points). But, due to Property 5, the corner points mentioned above are extremal points and thus still eccentric point candidates (see Section 6 for a discussion).

### 3 Previous Work - Algorithms

The naive algorithm to compute  $ECC(\mathcal{S})$  for a given discrete shape  $\mathcal{S}$  has a complexity of  $O(|\mathcal{S}|^3)$  in the number of pixels  $|\mathcal{S}|$ . Eccentric points lie only on the boundary of  $\mathcal{S}$ . If we assume that the average number of border points  $|\partial\mathcal{S}|$  is much smaller than  $|\mathcal{S}|$ , and we use Dijkstra's algorithm [15] to compute the shortest paths from one point to all other points (runs in  $O(|E| + |\mathcal{S}| \log |\mathcal{S}|)$ , where  $|E|$  is the number of edges i.e. the number of adjacent pixel pairs), we get a complexity of  $O(|\partial\mathcal{S}|(|E| + |\mathcal{S}| \log |\mathcal{S}|))$  for a more efficient algorithm. One can also use fast marching [16] for the computation of geodesic length ( $O(|\mathcal{S}| \log |\mathcal{S}|)$ ), a case in which the complexity decreases to  $O(|\partial\mathcal{S}||\mathcal{S}| \log |\mathcal{S}|)$ .

For continuous shapes, a detailed study of the ellipse and some properties regarding rectangles and a class of elongated shapes are given in [13]. Ellipses can be divided along the short diameter and to each point of the short diameter one eccentric point in each half ellipse can be associated. The eccentricity of any point of the ellipse can be computed in linear time by finding its corresponding point on the short diameter [13]. Also, in the case of the ellipse, the set of eccentric points can be analytically characterized. For the eccentricity of the  $n$  corner points of simple polygons, an  $O(n \log n)$  algorithm was given in [14].

### 4 Distances Inside a Polygonal Shape

Given a point  $\mathbf{o}$  of the input polygon  $\mathcal{S}$  we want to calculate the shortest distance between the point  $\mathbf{o} \in \mathcal{S}$  and an arbitrary point  $\mathbf{p}$  inside the shape. In a simply connected convex shape it is the length of the straight line  $d(\mathbf{o}, \mathbf{p})$  connecting  $\mathbf{o}$  and  $\mathbf{p}$ . However concave portions and holes may not allow straight connections in all cases.

We cover the inside of a polygonal shape by patches  $\mathcal{P}_i$  within which the distance of a point  $\mathbf{p} \in \mathcal{P}_i$  is the distance to a reference point  $\mathbf{r}_i \in \mathcal{P}_i$  plus a handicap  $h_i$ :

$$d(\mathbf{p}, \mathbf{o}) = d(\mathbf{p}, \mathbf{r}_i) + h_i \tag{3}$$

The handicap  $h_i$  corresponds to the length of the shortest path inside  $\mathcal{S}$  between  $\mathbf{o}$  and the reference point  $\mathbf{r}_i$ .

---

Create Patch( $\mathbf{r}, h, \mathcal{P}$ )

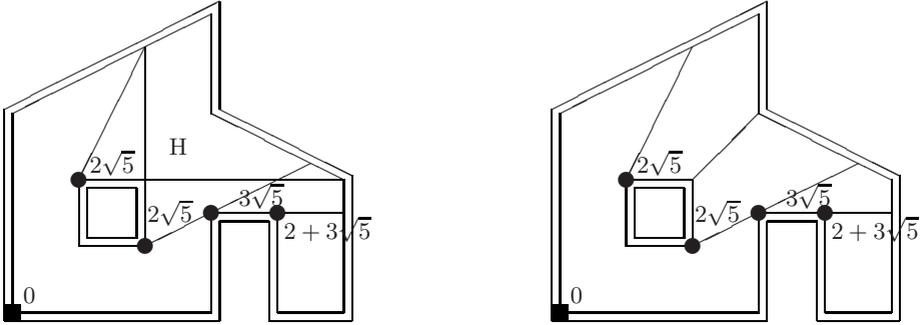
1. determine the visibility polygon  $\mathcal{Q} \subset \mathcal{P}$  delineating the region inside  $\mathcal{P}$  that can be reached from reference point  $\mathbf{r}$  along a straight line.
  2. for all occluding points  $\mathbf{t} \in \mathcal{Q}$  do  
     Create Patch( $\mathbf{t}, d(\mathbf{r}, \mathbf{t}) + h, \mathcal{P} \setminus \mathcal{Q}$ )
  3. return patch  $\mathcal{Q}$
- 

We initialize the computation backwards from potential eccentric points by creating patches for all corner points  $\mathbf{o} \in \mathcal{S}$ : Create Patch( $\mathbf{o}, 0.00, \mathcal{S}$ ).

Fig. 2 illustrates the first steps. It shows that patches overlap in the shadow of holes. Every hole of the shape can be bypassed by the shortest path on either sides of the hole. Hence all the points in the shadow of a hole can be reached along two alternate paths. Two overlapping patches can be cut along the *separation curve*, which is the curve where the two paths have equal length from  $\mathbf{o}$ :

$$d(\mathbf{p}, \mathbf{r}_1) + h_1 = d(\mathbf{p}, \mathbf{r}_2) + h_2. \tag{4}$$

In general the separation curve is a hyperbolic arc. It degenerates into a straight line as in area  $H$  of Fig. 2 if the two handicaps are equal.



**Fig. 2.** First patches created by the reference point (bottom-left corner), the obtained handicaps, and the shadow 'H' created by the hole

Assume that the two reference points on the hole are  $\mathbf{r}_1$  and  $\mathbf{r}_2$  with handicaps  $h_1$  and  $h_2$  respectively. After translating the midpoint  $\frac{\mathbf{r}_1 + \mathbf{r}_2}{2}$  to the origin and rotating, the two reference points have coordinates  $\mathbf{r}_1 = (-f, 0)$  and  $\mathbf{r}_2 = (f, 0)$ . Points  $\mathbf{p} = (x, y)$  that the same distance to the original point must satisfy

$$d(\mathbf{p}, \mathbf{r}_1) + h_1 = d(\mathbf{p}, \mathbf{r}_2) + h_2 \tag{5}$$

Without restricting generality we assume that  $h_2 > h_1$  and set  $l = h_2 - h_1$ . Then we have  $\sqrt{(x+f)^2 + y^2} = \sqrt{(x-f)^2 + y^2} + l$ ,  $(x+f)^2 + y^2 = (x-f)^2 + y^2 + 2l\sqrt{(x-f)^2 + y^2} + l^2$ ,  $4xf - l^2 = 2l\sqrt{(x-f)^2 + y^2}$ ,  $16x^2f^2 - 8xfl^2 + l^4 = 4l^2(x^2 - 2xf + f^2 + y^2)$ ,  $\frac{16f^2 - 4l^2}{l^2(4f^2 - l^2)}x^2 - \frac{4}{4f^2 - l^2}y^2 = 1$ , or the hyperbola  $\frac{x^2}{(l/2)^2} - \frac{y^2}{f^2 - (l/2)^2} = 1$ .

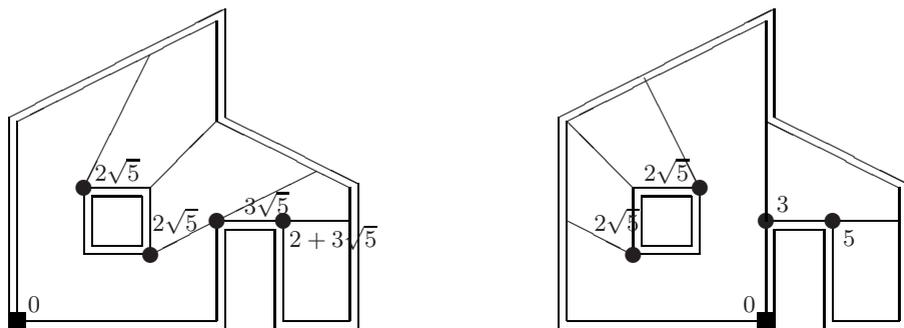
Note that the computation is independent for each starting point  $\mathbf{o}$  and hence can be done in parallel.

## 5 Combining Distance Patches into the ECC-Patches

Every original starting point creates a separate set of patches in which the distance to the original point can be computed locally. Let us call these original patches the *distance patches*. In the following we combine the different sets of patches into a new partition of the ECC. The new patches will be called *ECC patches*.

### 5.1 The Smallest Common Partition

After creating a partition into patches for all starting points of the given polygon the produced partitions will not coincide with each other. We therefore overlay them to create a finer partition such that every patch of the finer partition fits into any of the distance patches. Fig. 3 shows the first two sets of patches.



**Fig. 3.** The first two sets of patches

The resulting finer patches may have as many reference points as there were original points on the polygon since any extremal point on the polygon can be reached from any patch.

## 5.2 Example with One Concave Part

Fig. 4 shows a simple example with 8 corner points and 11 patches A, B, C, D, E, F, G, H, I, J, K. The table in Fig. 4 lists all the patches together with the reference points leading to all the 8 corner points together with the handicaps.

## 5.3 Non-maxima Suppression

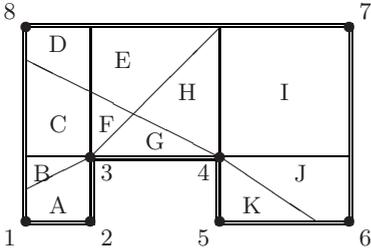
The ECC computes the longest distance between any point  $\mathbf{p}$  inside the shape to the farthest extremal point  $\mathbf{x}$  inside the shape along the shortest path inside the shape. Every patch of the refined partition of the shape contains  $n \geq 1$  reference points allowing the computation of the length of the shortest paths to any of the  $n$  original points.

First a reference point  $\mathbf{r}$  may appear several times with different handicaps  $h$  depending on the number of paths from original points that go across  $\mathbf{r}$ . Only the largest handicap must be kept (dropped handicaps are indicated by  $(h)$  in Fig. 4).

For the ECC patch we need to keep further only the reference points  $\mathbf{r} \in \mathcal{P}$  leading to the farthest extremal point:  $\operatorname{argmax}\{d(\mathbf{p}, \mathbf{r}) + h \mid \mathbf{p} \in \mathcal{P}\}$ . In some cases there may be more than one reference point leading to maximal distances in one patch. The patches that have more than one reference point after non-maxima suppression need to be further subdivided along curves separating influence areas of extremal points.

## 5.4 Subdividing Center Patches

Any remaining patch has two or more reference points  $\mathbf{r}_i$ . The subdivision can proceed similar to the generation of a weighted Voronoi diagram. Any pair of



patch	1	2	3	4	5	6	7	8
A <b>r</b>	1	2	3	3	3	3	<b>3</b>	8
<i>h</i>	0.0	0.0	(0.0)	(2.0)	(3.0)	4.2	<b>4.5</b>	0.0
<i>d<sub>max</sub></i>	1.4	1.1	1.4	3.4	4.4	5.6	<b>5.9</b>	3.1
B <b>r</b>	1	2	3	3	3	3	<b>7</b>	8
<i>h</i>	0.0	0.0	(0.0)	(2.0)	(3.0)	4.2	<b>0.0</b>	0.0
<i>d<sub>max</sub></i>	1.4	1.4	1.1	3.1	4.1	5.3	<b>5.6</b>	2.5
C <b>r</b>	1	2	3	4	4	<b>4</b>	7	8
<i>h</i>	0.0	0.0	0.0	(0.0)	(1.0)	<b>2.2</b>	0.0	0.0
<i>d<sub>max</sub></i>	2.5	2.7	1.8	3.6	4.6	<b>5.8</b>	5.4	2.2
D <b>r</b>	1	2	3	4	4	<b>6</b>	7	8
<i>h</i>	0.0	0.0	0.0	(0.0)	1.0	<b>0.0</b>	0.0	0.0
<i>d<sub>max</sub></i>	3.2	3.2	2.2	3.6	4.6	<b>5.8</b>	5.0	1.4
E <b>r</b>	1	3	3	4	4	<b>6</b>	7	8
<i>h</i>	0.0	1.0	(0.0)	(0.0)	1.0	<b>0.0</b>	0.0	0.0
<i>d<sub>max</sub></i>	4.2	3.8	2.8	2.8	3.8	<b>5.0</b>	4.1	3.0
F <b>r</b>	1	3	3	4	4	4	<b>7</b>	8
<i>h</i>	0.0	1.0	(0.0)	(0.0)	(1.0)	2.2	<b>0.0</b>	0.0
<i>d<sub>max</sub></i>	2.4	2.0	1.0	2.2	3.2	4.4	<b>4.5</b>	2.2
G <b>r</b>	3	3	3	4	4	4	<b>7</b>	8
<i>h</i>	1.4	(1.0)	(0.0)	(0.0)	(1.0)	2.2	<b>0.0</b>	0.0
<i>d<sub>max</sub></i>	3.4	3.0	2.0	2.0	3.0	4.2	<b>4.5</b>	3.6
H <b>r</b>	<b>3</b>	3	3	4	4	6	7	8
<i>h</i>	<b>1.4</b>	(1.0)	(0.0)	(0.0)	1.0	0.0	0.0	0.0
<i>d<sub>max</sub></i>	<b>4.2</b>	3.8	2.8	2.0	3.0	4.1	3.9	3.6
I <b>r</b>	<b>3</b>	3	3	4	5	6	7	8
<i>h</i>	<b>1.4</b>	(1.0)	(0.0)	0.0	0.0	0.0	0.0	0.0
<i>d<sub>max</sub></i>	<b>5.9</b>	5.5	4.5	2.8	3.6	3.6	2.8	5.4
J <b>r</b>	4	4	4	4	5	6	7	<b>8</b>
<i>h</i>	3.4	(3.0)	(2.0)	(0.0)	0.0	0.0	0.0	<b>0.0</b>
<i>d<sub>max</sub></i>	5.6	5.2	4.2	2.2	2.2	2.2	3.0	<b>5.8</b>
K <b>r</b>	4	4	4	4	5	6	7	<b>4</b>
<i>h</i>	3.4	(3.0)	(2.0)	(0.0)	0.0	0.0	0.0	<b>3.6</b>
<i>d<sub>max</sub></i>	5.2	4.8	3.8	1.8	1.5	2.2	3.6	<b>5.4</b>

**Fig. 4.** Polygon with ECC patches and table of reference points (**r**), handicaps (*h*) and distances (*d<sub>max</sub>*) of ECC patches (A-K) to corner points (1-8, table header)

reference points  $\mathbf{r}_1, \mathbf{r}_2$  subdivides the patch into two half spaces along a second order curve through  $N = \frac{\mathbf{r}_1 + \mathbf{r}_2}{2} + (h_1 - h_2) \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|}$ . The remaining reference points are split among the two new patches and the subdivision repeated for all patches having more than one reference point until all patches have only one reference point left.

### 5.5 Merging Oversegmented Patches

The successive subdivision of patches may have introduced patches that have the same reference point. These patches can be merged in the final ECC-partitions

which subdivides the original shape into patches having exactly one reference point. The ECC value of any point inside the patch can be computed using the reference point  $\mathbf{r}$  and its handicap  $h$ :

$$ECC(\mathcal{S}, \mathbf{p}) = d(\mathbf{p}, \mathbf{r}) + h \tag{6}$$

## 6 Discussion

Sections 4 and 5 produce a partition of shape  $\mathcal{S}$  into patches  $\mathcal{D}(\mathcal{S})$  s.t. the eccentricity transform is analytically defined in each patch. Given the decomposition  $\mathcal{D}(\mathcal{S})$ , the computation of the eccentricity of a point  $\mathbf{p} \in \mathcal{S}$  is reduced to:

---

Compute  $ECC(\mathcal{S}, \mathbf{p}, \mathcal{D})$ :

1. find patch  $\mathcal{P} = (\mathbf{r}, h) \in \mathcal{D}$  s.t.  $\mathbf{p} \in \mathcal{P}$
  2.  $ECC(\mathcal{S}, \mathbf{p}) = d(\mathbf{p}, \mathbf{r}) + h$
- 

In a hierarchical structure (e.g. binary trees, quadtrees, irregular pyramids), step 1 runs in logarithmic time in the number of patches. Step 2 executes in a fixed amount of time. The number of patches depends on the number of corner points, the number of holes, and the number of points for which the computation in Section 4 was initialized.

For a simple polygon, the correct eccentricity is computed. In the worst case all corner points are also eccentric points and define at least one patch.

A hole makes the points in the shadow less accessible, which can make border points on the separation curve in the shadow further away than any corner point, and thus eccentric. The assumption that only corner points can be eccentric might not hold, and the number of eccentric points can be infinite. In this case the presented algorithm gives an approximation of the eccentricity transform, less or equal to the correct value. An upper bound for the error is half the length of the longest polygon side. To reduce this upper bound, one can initialize the computation in Section 4 with additional boundary points.

## 7 Extensions

**Circular Arc:** Reference point and handicap can also be used to correctly describe shapes with circular arcs. In this case the center of the circle serves as reference point and the radius as handicap. For concave circular arcs the radius carries a negative sign.

**3D triangulated surface:** In a triangulated surface two adjacent triangles share an edge. The corresponding 3D straight line serves as rotational axis to place the two adjacent triangles into a common plane such that shortest paths become straight lines in this new plane. Therefore the length of the shortest path from a corner to a point in an adjacent triangle can be

computed as the Euclidean distance to the corner after rotating it into the common plane. This can serve as reference point for all straight lines crossing the common edge segment.

## 8 Conclusion

In this paper an algorithm for efficiently computing the eccentricity transform of a continuous polygonal shape is presented. Corner points are candidates for eccentric points. Parallel steps are used to decompose the shape based on the length and the topology of the shortest paths to each corner point. The resulting decompositions are merged and used to derive the eccentricity transform of the polygon. An algorithm for discrete shapes will be derived from the one presented here. Future work includes a general algorithm for 2D continuous shapes.

## References

1. Rosenfeld, A.: A note on 'geometric transforms' of digital sets. *Pattern Recognition Letters* 1(4), 223–225 (1983)
2. Gorelick, L., Galun, M., Sharon, E., Basri, R., Brandt, A.: Shape representation and classification using the poisson equation. In: *CVPR* (2), pp. 61–67 (2004)
3. Kropatsch, W.G., Ion, A., Haxhimusa, Y., Flanitzer, T.: The eccentricity transform (of a digital shape). In: *13th International Conference on Discrete Geometry for Computer Imagery*, Szeged, Hungary, October 25-27, pp. 437–448. Springer, Heidelberg (2006)
4. Ogniewicz, R.L., Kübler, O.: Hierarchic Voronoi Skeletons. *Pattern Recognition* 28(3), 343–359 (1995)
5. Siddiqi, K., Shokoufandeh, A., Dickinson, S., Zucker, S.W.: Shock graphs and shape matching. *International Journal of Computer Vision* 30, 1–24 (1999)
6. Paragios, N., Chen, Y., Faugeras, O.: 6. In: *Handbook of Mathematical Models in Computer Vision*, pp. 97–111. Springer, Heidelberg (2006)
7. Soille, P.: *Morphological Image Analysis*. Springer, Heidelberg (1994)
8. Harary, F.: *Graph Theory*. Addison-Wesley, Reading (1969)
9. Diestel, R.: *Graph Theory*. Springer, New York (1997)
10. Klette, R., Rosenfeld, A.: *Digital Geometry*. Morgan Kaufmann, San Francisco (2004)
11. Ion, A., Peyré, G., Haxhimusa, Y., Peltier, S., Kropatsch, W.G., Cohen, L.: Shape matching using the geodesic eccentricity transform - a study. In: *31st OAGM/AAPR, Schloss Krumbach, Austria, OCG* (May 2007)
12. Maisonneuve, F., Schmitt, M.: An efficient algorithm to compute the hexagonal and dodecagonal propagation function. *Acta Stereologica* 8(2), 515–520 (1989)
13. Ion, A., Peltier, S., Haxhimusa, Y., Kropatsch, W.G.: Decomposition for efficient eccentricity transform of convex shapes. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) *CAIP 2007*, Springer, Heidelberg (2007)
14. Suri, S.: The all-geodesic-furthest neighbor problem for simple polygons. In: *Symposium on Computational Geometry*, pp. 64–75 (1987)
15. Khuller, S., Raghavachari, B.: *Basic Graph Algorithms*. CRC Press (1998)
16. Sethian, J.: *Level Sets Methods and Fast Marching Methods*, 2nd edn. Cambridge Univ. Press, Cambridge (1999)