

The Eccentricity Transform (of a Digital Shape)^{*}

Walter G. Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer

Pattern Recognition and Image Processing Group
Vienna University of Technology, Favoritenstr. 9/1832, A-1040 Vienna, Austria
`{kwr, ion, yll, flanitzt}@prip.tuwien.ac.at`

Abstract. Eccentricity measures the shortest length of the paths from a given vertex v to reach any other vertex w of a connected graph. Computed for every vertex v it transforms the connectivity structure of the graph into a set of values. For a connected region of a digital image it is defined through its neighbourhood graph and the given metric. This transform assigns to each element of a region a value that depends on its location inside the region and the region's shape. The definition and several properties are given. Presented experimental results verify its robustness against noise, and its increased stability compared to the distance transform. Future work will include using it for shape decomposition, representation, and matching.

1 Introduction

Recognition, manipulation and *representation* of visual objects can be simplified significantly by “abstraction”. Abstraction extracts essential features and properties while it neglects unnecessary details. Shape is one such form of visual abstraction, which describes distinctive features of the object’s appearance i.e. its projection on the surface of a 2D sensor (in our case the retina). If shape matching is done invariant with respect to certain deformation classes (e.g. articulated motion), shape based object recognition can be used for generic object recognition, a much desired ability of humans.

Different approaches that use shape for recognition exist [1, 2, 3, 4], with many of them using the distance transform [5] derived skeletons [6, 7] as a basis for shape description. Skeletons have proved themselves to be the basis of powerful shape descriptors [2] with the main advantages including their ‘cue’ for a natural decomposition of shapes into parts (e.g. usually the parts of the skeleton of a human decompose its shape into body, limbs, and head) and their invariance to certain types of movement including the very important articulated motion. On the other hand, one of their weak points come from their apparent locality and the fact that they are derived from the distance transform which is known to be unstable with respect to small perturbation of the shape (e.g. spurious branches can appear in the skeleton if a few pixels are added at the border of the region).

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

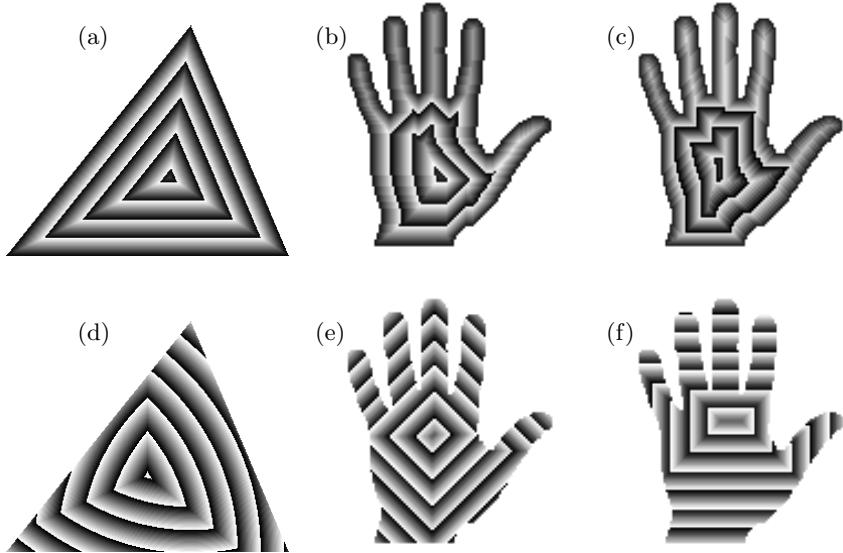


Fig. 1. Isoheight lines for distance (a-c) and eccentricity (d-f) transform of 2 images, using the euclidean (a,d), 4- (b,e), and 8- (c,f) neighbourhood (where continuous, lighter means higher value)

The distance transform associates to each point of the shape, the minimum distance from it to the border of the shape (see Fig. 1a: gray values are independent between the two images, and where continuously changing: lighter means higher value), which makes it very unstable with respect to Salt and Pepper noise and certain kind of segmentation errors. Approaches like removing regions below a certain size or pruning spurious branches of the obtained skeleton have been used to cope with these kinds of problems, but this has been shown not to be the optimal way and should be avoided mainly because the size of a region does not tell anything about its importance [8].

Instead of minimum distance, other measures have also been used, e.g. the mean time for a randomly moving particle to hit the border [9].

Inspired from graph theory, we present a new transform which associates to each point the longest distance (geodesic) from it to the points on the border of the shape (see Fig. 1d). We show that it is robust against the types of noise mentioned above, and comment about it's applicability to shape description and matching.

We recall the distance transform in Sec. 2, including a formulation for the distance transform of a graph (Sec. 2.2). The eccentricity transform is presented in Sec. 3, beginning with a recall of the graph theory based definition for eccentricity (Sec. 3.1). The properties of the transform are discussed in Sec. 4,

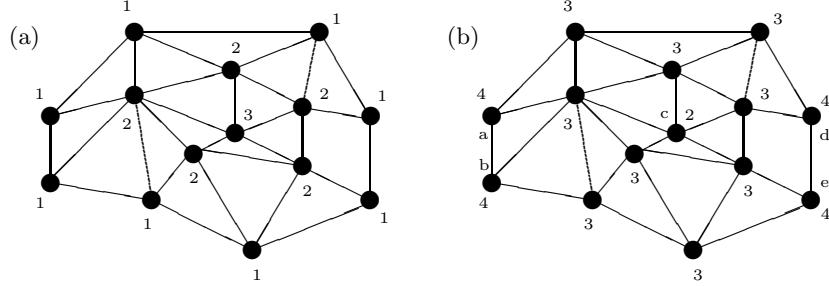


Fig. 2. Distance transform (a) and eccentricity transform (b) of a graph

followed by computation strategies in Sec. 5. Experimental results in Sec. 6 will complete the presentation, summed up in Sec. 7 with conclusions and outlook.

2 Distance Transform

The distance transform assigns to each point in the binary image a value of a distance to the closest point on its border (obstacles). Let $I = B \cup \overline{B}$ be a binary image and let a point $p \in B$. We adapt the definition of the general distance function [10] for the rest of the section. A neighbourhood N_i is a pair of (P_i, d_i) , where P_i is a finite subset of \mathbb{Z}^K and d_i is a function $d_i : P \rightarrow \mathbb{R}^+$, for $i = 0, \dots, T-1$ and $T, K \in \mathbb{Z}^+$. We say that p_i is adjacent to p_{i+1} iff $p_{i+1} = p_i + r$ for some $r \in P_i$. Let α be a finite sequence of neighbourhoods $N_0 N_1 \dots N_{T-1}$, where T is called the period of the sequence. The distance transform dt_α of I associates to every point $p \in B$ the minimal distance from p to \overline{B} , formally we write:

$$dt_\alpha(p) = \min\{\lambda(\pi_\alpha(p, q)) \mid q \in B \wedge q \in N_\alpha(\overline{q}), \overline{q} \in \overline{B}\}, \quad (1)$$

where N_α is an α -adjacency, $\pi_\alpha(p, q)$ is the set of all α -paths from point p to q , and $\lambda(\pi_\alpha(p, q))$ is the length of one of the paths $\pi_\alpha(p, q)$. The α -path is a sequence of points (p_0, p_1, \dots, p_n) , such that end points are $p_0 = p$, $p_n = q$, and p_{i+1} is α -adjacent to p_i ($0 \leq i \leq n-1$), then the length of this path is the sum of $d_i(p_{i+1} - p_i)$ for all $i = 1, \dots, n$. If $d_i(r) = 1$ then the length of the path is n , the number of points. To define the chessboard distance (dt_8) or the square distance (dt_4) one takes the sequence of neighbourhood with $T = 1$ ($\alpha = N_0$) and defines the neighbourhood P_0 as in [10, page 239]. Note that there may be many shortest paths. If there is no other shorter α -path between the same end points, then this path is called α -geodesic [11]. The border point $q \in B$ is α -neighbour of a point not being in B . In the Euclidean space there is always a unique path between two points, which is the straight line between the points. This straight line does not exist in digital images and thus the distance transform computed is dependent on the way the neighbourhood is defined, i.e. how the Euclidean distance is approximated. In the section below we use the definitions above as the basis of defining the distance transform of digital images and graphs.

2.1 Distance Transform of a Digital Image

The transformation of the continuous space \mathbb{R}^n into a discrete space \mathbb{Z}^n is done by sampling \mathbb{R}^n . A particular sampling scheme can be used to digitise the continuous space. If there is no a priori knowledge about the local variation, the usual scheme is the square or hexagonal grid. For the sake of the presentation we will constrain the discussion only on digital images on 2D square grids with the 4-neighbourhood (city block metric), and the 8-neighbourhood (chessboard metric). Using Eq. 1 one can define the distance transform dt_4 and dt_8 , respectively. These distance transforms are easy to compute by scanning the image twice [12, 11], although they are not a good approximation of the Euclidean distance. In Fig. 1b,c) distance transforms dt_4 and dt_8 of a binary image are shown. Better approximations can be found by chamfer distances [13].

2.2 Distance Transform of a Graph

If the sampling grid is not regular, one could use graph representation for the sampling points. Let $G = (V, E, a, w)$ be the undirected weighted graph with vertices $v \in V$ representing sampling points, edge set $e = (v, w) \in E$ representing the connection between vertices; and $a : V \rightarrow \mathbb{Z}^+$ and $w : E \rightarrow \mathbb{Z}^+$ are attributes on vertices and edges respectively. Let the weights on edges represent the cost of going from one vertex to the other. In order to define the distance transform one should define the boundary vertices [14]. Any bounded region has a boundary that separates it from the background. The background can be considered as the complement of the region with respect to the embedding space. Border faces are faces of the dual graph that are surrounded by both vertices of $G \subset G' = (V', E')$. The boundary of a subgraph $G = (V, E) \subset G' = (V', E')$ collects all the vertices $C \subset V$ which bound border faces.

A path $\pi_G(v, w)$ is a sequence of vertices (v_0, v_1, \dots, v_n) in G such that the end vertices are $v = v_0$, $w = v_n$, all vertices are distinct and $\exists e = (v_i, v_{i+1}) \in E, i = 0, 1, \dots, n - 1$. The length $\lambda(\pi_G)$ of path λ_G is the sum of the edge weight in the sequence:

$$dt_G(v) = \min\{\lambda(\pi_G(v, w)), v \in G \setminus C \wedge w \in C\}. \quad (2)$$

Usually, the border vertices are set to 1. If vertices v and w are not connected, we say that the $\lambda(\pi_G(v, w))$ is infinite. If the graph G is connected then this distance is a graph metric [11]. A simple example of the distance transform on a graph is given in Fig. 2a, where the edge cost is set to 1. Note that square grid can be easily represented by graphs. In this case the weight on edges could be set to 1 (but not necessarily). Similar to the square grid, we can define the 4-, 8-neighbourhood of vertices.

3 Eccentricity Transform

The eccentricity transform assigns to each point in the binary image the shortest distance to the point farthest away from it. Analogously, to the notation presented in Sec. 2 we define the eccentricity transform $ecc_\alpha(p)$ of $I = B \cup \overline{B}$ such

that it associates to every point $p \in B$ the longest of the distances to any other point $q \in B$, formally we write:

$$ecc_\alpha(p) = \max\{\lambda(\pi_\alpha(p, q)) \mid \forall q \in B\}, \quad (3)$$

where $\pi_\alpha(p, q)$ is the shortest α -path from point p to q , and $\lambda(\pi_\alpha(p, q))$ is the length of the path $\pi_\alpha(p, q)$. In the section below we use the Eq. 3 as the basis in defining the eccentricity transform of graphs and digital images.

3.1 Eccentricity Transform of a Graph

Let $G = (V, E, w)$ be an attributed undirected and connected graph with vertex set V , edge set E and with edge weights $w : E \rightarrow \mathbb{Z}^+$ as the cost of going from one vertex to the other. Let v be a vertex in V . The eccentricity $ecc_G(v)$ of v is the distance to a vertex farthest from v and it is defined as [15, Page 31]:

$$ecc_G(v) = \max\{\lambda(\pi_G(v, w)) \mid \forall w \in V\}, \quad (4)$$

where $\lambda(\pi_G(v, w))$ is the length of the shortest path between the two vertices v and w . One could say that eccentricity of a vertex is the longest shortest path to any other vertex in the graph. A simple example of the eccentricity transform is given in Fig. 2b), where we set the edge's cost to 1.

Some definitions concerning the eccentricity transform are of importance [15]:

- the *eccentric* vertices of v are all the vertices w at a distance $ecc_G(v)$;
- the *radius* $r(G)$ of G is the minimum eccentricity;
- the *diameter* $d(G)$ of G is the maximum eccentricity;
- v is a *central vertex* of G if $ecc_G(v) = r(G)$;
- the *center* $C(G)$ is the set of all central vertices;
- v is a *peripheral vertex* of G if $ecc_G(v) = d(G)$;
- the *periphery* $P(G)$ is the set of all peripheral vertices;

For the graph in Fig. 2b) the radius $r(G) = 2$, the diameter $d(G) = 4$, central vertex c , the center $C(G) = \{c\}$, peripheral vertices a, b, d and e , and the periphery $P(G) = \{a, b, d, e\}$. Sec. 4 presents a detailed discussion of the properties.

3.2 Eccentricity Transform of a Digital Image

Similarly to distance transform, a particular sampling scheme can be used to discretize an image. We constrain our discussion only on a 2D square grid, and two classical pixel adjacencies; the 4– and 8–neighbourhood. Let I be a binary image $I = B \cup \overline{B}$, and let a pixel p be in B . Now we can use Eq. 3 to define the eccentricity transform ecc_α on a square grid digital image for the connected set B . One can say that eccentricity of a pixel is the longest shortest path to any other pixel in the same connected region. Similarly to the distance transform also the eccentricity transform is affected by how well the Euclidean distance can be approximated. The same concerns made in Sec. 2 with respect to the Euclidean plane apply for eccentricity transform as well. Thus eccentricity transform is also dependent on the way one defines the pixel neighbourhood. In Fig. 1e,f eccentricity transforms ecc_4 and ecc_8 are shown on the same hand image as in Fig. 1b,c.

4 Properties of the Eccentricity Transform

We shortly discuss some of the properties of the eccentricity transform, some of them known from graph theory and extended to the discrete domain, some are interesting and useful in the context of describing the shape of a region.

Center: The vertices with the minimum value of the eccentricity transform are called the center of the graph. They lie in a block of the graph, i.e. the corresponding subset of vertices containing the center is connected and does not contain a cut vertex. We notice that the center of a discrete region is always a part of the region in contrast to the center of gravity which can be located outside the region in case of a concavity or a hole in the middle of the region. This may be useful in several applications, e.g. in tracking where the center of a tracked region may be used as the start for searching the region in the next frame of the sequence.

Robustness: The eccentricity transform is robust with respect to (salt and pepper) noise. This is due to the fact that a noise vertex on the path between two distant points '*just goes around*' the obstacle without prolongating the length by much. In the case of discrete metrics (like 4- or 8-connectivity) the likelihood of finding many paths with the same (shortest) length is very high. In such a case the eccentricity is affected only if all shortest paths between the vertex and its farthest vertex are interrupted by noise or a noisy pixel (vertex) p is added to B such that p is at maximum distance from the vertex.

4-connectivity: In the Euclidean space two points are connected by a unique straight line. In the discrete space with 4-connectivity this is only the case if the two points are along the two coordinate axes, in all other cases there are more than one shortest paths. In fact any permutation of the two primitive steps to connect the two end points is also a shortest path.

This fact increases the robustness of the eccentricity transform but has also two other consequences:

1. There is not a single midpoint between the two endpoints making the center of an elongated region a diagonal line. In fact the length of this line is as long as the smaller coordinate differences of the two end points.
2. Since the number of midpoints depends on the angle of the discrete line the resulting centers are no more rotationally invariant (which they are in the Euclidean case).

Maxima are all on the boundary if the graph has no inner pending vertex: (See Sec. 2.2 for the boundary of a graph) If G' is connected there are paths between any pairs of vertices $v, w \in V'$. Any non-border vertex has a degree greater than 1. If none of the neighbours of a vertex of degree greater than 1 belongs to a border face it cannot be extremal since any path leading to it can be continued.

Complementarity between distance transform and eccentricity transform: In the distance transform the smallest values are on the boundary and the highest values can be found where a circle with maximum radius

Algorithm 1 – Eccentricity Transform-naive implementation**Input:** Attributed graph $G = (V, E)$.

```

1: for all  $v \in V$  do
2:    $ecc(v) \leftarrow 0$ 
3:   for all  $u \in V - \{v\}$  do
4:      $ecc(v) \leftarrow \max\{ecc(v), \text{shortestPathLength}(v, u)\}$ 
5:   end for
6: end for

```

Output: Eccentricity $ecc(v)$ for all vertices $v \in V$ of G .

touches the boundary in at least two opposite points. These local maxima form the skeleton/medial axis/symmetry transform. Local maxima of the eccentricity transform are on the boundary while the minimum defines the center. However there are local minima along the boundary and discontinuities inside the region which give rise to interesting partitionings of the region.

Invariance: The eccentricity transform computes the lengths of paths inside a given region. It is therefore invariant to any translation and invariant to rotation for the Euclidean metric. There is some dependency on the orientation for discrete metrics but not for all shapes. Furthermore in the case of thin regions, it is robust with respect to articulated motion, it may differ by the thickness of the shape at the articulation point which in many natural cases is thin in relation to the length (arms, legs, fingers).

5 Computation

Two algorithms for computing the eccentricity transform are given here. They are both defined for graphs, but the adaptation to digital images is straight forward. One has just to decide for a neighbourhood ($\alpha = \{4, 8\}$) and choose the pixels that make up the connected region for which the transform will be applied. Note that Floyds [16] algorithm, that produces the minimum path length from all vertices to all other vertices can also be used to obtain the eccentricity (for each vertex, one has just to take the maximum of the values obtained for it).

Naive Alg. 1 iterates through all the vertices of G and for each, it calculates the maximum of the length of the shortest paths to all other vertices in the graph. Lines 3 - 5 can be implemented by taking the maximum of the lengths calculated using Dijkstra's single-source shortest path algorithm [16]. The complexity of the naive implementation is between $O(|V|^3)$ and $O(|V||E| + |V|^2 \log |V|)$ depending on the implementation of the shortest path problem.

Alg. 2 uses the fact that the set of *eccentric vertices* is a subset of V and that calculating the shortest path for each of these vertices to all the other vertices in V and combining the results (i.e. taking the maximum) is enough to obtain the eccentricity transform for the whole graph. The eccentric vertices can be found by calculating the shortest path from the center of the graph to all the other vertices and looking at the local maximum. To find the center of a graph we find its diameter, which is connecting the vertices with the highest eccentricity.

Algorithm 2 – Eccentricity Transform-optimised implementation**Input:** Attributed graph $G = (V, E)$.

```

1:  $\forall v \in V, ecc(v) \leftarrow 0$  /* initialise eccentricity cumulation table with 0 */
2:  $vp \leftarrow$  random vertex of  $V$ 
3: repeat
4:   mark  $vp$  as visited
5:    $\forall v \in V, ecc(v) = \max\{ecc(v), \text{shortestPathLength}(v, vp)\}$ 
6:    $vp \leftarrow$  vertex with maximum  $ecc$ 
7: until ( $ecc$  not changed) or ( $vp$  allready visited) /* iterate until the endpoints of a
   diameter found */
8: repeat
9:    $m \leftarrow$  one (random) or all unvisited vertices with minimum  $ecc$ 
10:  for all  $v_m \in m$  do
11:     $\forall v \in V, ecc(v) = \max\{ecc(v), \text{shortestPathLength}(v, v_m)\}$ 
12:    mark  $v_m$  as visited
13:  end for
14:   $M \leftarrow$  all unvisited vertices with local maximum  $ecc$ 
   /*  $M$  includes non monotonic maxima */
15:  for all  $v_M \in M$  do
16:     $\forall v \in V, ecc(v) = \max\{ecc(v), \text{shortestPathLength}(v, v_M)\}$ 
17:    mark  $v_M$  as visited
18:  end for
19: until  $ecc$  not changed /* repeat until converged */

```

Output: Eccentricity $ecc(v)$ for all vertices $v \in V$ of G .

Lines 3 - 7 start with a random point and iterate to find the vertices with the highest eccentricity (diameter endpoints). The calculated shortest path lengths are added to the cumulation table ecc and the vertex is marked as visited. Lines 8 - 19 iterate finding the center vertices and the local maximum until the ecc cumulation table converges. On line 9 (approximate the center), two options have been tried, taking one or all the existing minima. On shapes without holes, both have produced the correct solution, while on shapes with holes neither of them did. In our experiments, the first loop (lines 3 - 7) converged after 3 cycles (random point, first diameter end, second diameter end). The second loop is bounded by the number of vertices on the border of the graph.

Alg. 2 is much faster than Alg. 1 but gives correct results only on simply connected shapes (no holes). On shapes with holes, complex forms of the center appear e.g. for a disc with a circular hole in the middle, the center consists of a circle for euclidean distance, and a set of disconnected points for 4 connectivity, all concentrated around the hole. In such cases, Alg. 2 produces results close to the correct one, but we cannot give any upper bound for the error.

6 Experiments

We have conducted experiments to test the properties of the eccentricity transform and find the important differences compared to the distance transform.

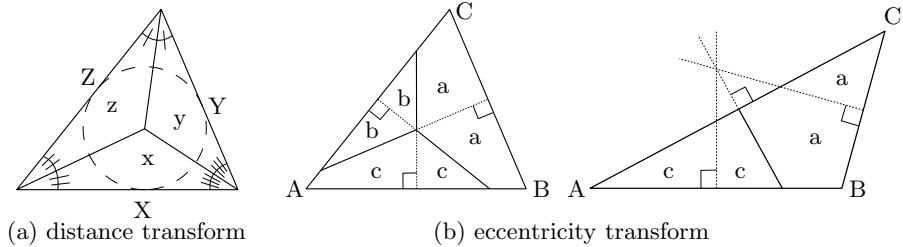


Fig. 3. Transforms on a triangle. Capital letters denote triangle elements (A,B,C points; X, Y, Z segments) and lower case letters denote their respective influence area.

6.1 Eccentricity Transform of a Triangle

First we have looked at a simple shape, the triangle, for which both transforms, when using the euclidean distance, can be solved analytically. In the case of the distance transform (see Fig. 1a and 3a), the 3 angle bisectors divide the triangle into 3 parts (x , y , z in Fig. 3a), with all the points inside the same part having the distance transform equal to the distance to the same side of the triangle. The point with the highest distance transform is the intersection of the 3 angle bisectors i.e the incenter. The isoheight lines are all polygonal lines (triangles).

In the case of the eccentricity transform, the 3 perpendicular bisectors divide the triangle into 2 or 3 parts (see Fig. 1d and 3b) depending on whether the triangle is an obtuse one or not (i.e the circumcenter lies outside or within the triangle). All points inside the same part have the eccentricity transform equal with the distance to the same point. The isoheight lines are made out of arcs.

6.2 Properties Depending on Connectivity/Metric

Fig. 1 shows the isoheight lines of the eccentricity transform for the 4 and 8 connectivities. One can see that the place of the center (global minima) and the form of the isoheight lines changed. Depending on the shape, the positions of the diameter ends/global maxima also change.

6.3 Robustness Against Salt and Pepper Noise

To test the robustness against Salt and Pepper noise, we have calculated the eccentricity and distance transforms (using both 4– and 8– neighbourhood) for 89 randomly selected shapes from [2] (for some example shapes see the top row from Tab. 1). We applied 5% Salt and Pepper noise to the images and calculated the two transforms again.

To measure the robustness, for each image, each neighbourhood, and each transform, we have calculated the root mean square error (RMSE) between the values obtained for the original and noisy images (calculation was done using the values, of the pixels part of the shape, in both images i.e. noisy pixels are excluded). We have calculated for each image and each neighbourhood, the ratio

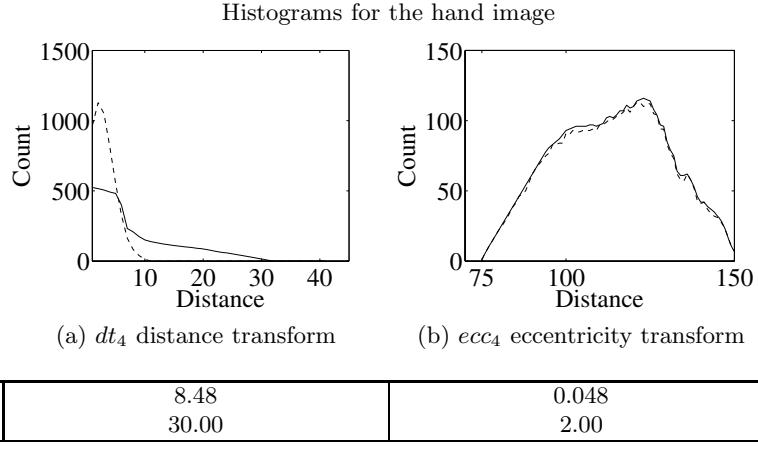


Fig. 4. Distance and eccentricity transform histograms, RMSE and Max. Diff. (solid - original image, dotted - noisy image)

between the RMSE for the eccentricity and distance transforms. Then, for each neighbourhood, we have calculated the mean of these ratios and inverted the result ($1/x$) i.e. we obtain dt error divided by ecc error. The error of the distance transform is **8.07** times higher in the case of the 4-neighbourhood and **22.63** times higher for the 8-neighbourhood, than the one of the eccentricity transform.

Fig. 4 shows the histogram of the eccentricity and distance transforms for one of the images, the hand (original and noisy) using the 4 neighbourhood. Also shown is the RMSE between the values of the transforms for the original and noisy images, and the maximum difference value for each transform. One can see that the error and maximum deviation of the eccentricity transform is much smaller than that of the distance transform. Note that in the case of the noisy image, a valid transform value has been calculated for less pixels. This makes the histogram of the eccentricity transform of the noisy image lie below the histogram of the original one.

6.4 Minor Segmentation Errors

For this experiment, we have selected a few shapes and simulated segmentation errors and partial occlusion by removing some parts of the shapes i.e. simulated noise on the border of the shape. We have calculated the correlation between the local maxima of the eccentricity transforms of the original and the images with partial occlusion (for each image, original and partially occluded, we have created a matrix where the positions of the eccentricity transform regional maxima were marked with 1, and the rest with 0, and calculated the correlation between the 2 matrices - only maxima that were located inside the partially occluded shape

Table 1. Correlation results for local maxima in eccentricity transform of original (top row) and partially occluded shapes (middle and bottom rows)

	Original Shapes									
Partially occluded set 1										
4-nb.	0.73	1.00	1.00	0.96	0.96	1.00	0.77	1.00	1.00	0.95
8-nb.	0.93	1.00	0.72	0.97	1.00	0.82	1.00	1.00	1.00	0.98
Partially occluded set 2										
4-nb.	0.71	0.79	0.97	0.96	0.89	0.97	0.71	0.98	0.87	0.92
8-nb.	0.48	0.45	0.90	0.96	0.72	0.65	0.98	0.97	0.73	0.98

were taken into consideration). The correlation Tab. 1 shows these shapes and the obtained correlation values.

6.5 Articulated Motion

To simulate articulated motion, two elongated parts have been overlapped at one of their ends in a way in which they approximate a joint (the angle between the two parts is a parameter, see Fig. 5a for some examples).

For each angle (in our experiment we have used 90° , 105° , 120° , 135° , 150° , 165° , and 180°) we have applied the eccentricity transform and calculated the minimum, maximum, and average eccentricity. Fig. 5b shows the mean and standard deviation of the 3 values over the whole spectrum of joint angles tested. Note that the values are stable under these conditions.

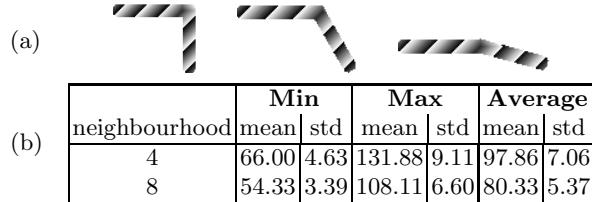


Fig. 5. Example of images used for testing the variation under articulated motion (a), and mean and standard deviation of eccentricity value for the simulated joint (b)

7 Conclusion and Outlook

We propose a new transform for a digital image called the eccentricity transform. This transform associates to every pixel the maximum length of the shortest paths connecting it with all the other vertices. The definition, several properties, and algorithms have been given. Presented experimental results verify its robustness against noise, and its increased stability compared to the distance transform, e.g. in the case of 5% Salt and Pepper noise, we obtained changes about 10 times higher of the distance transform compared to the eccentricity transform (a distance change of 30 in the case of the distance transform is 3 in the case of the eccentricity transform). Behaviour under minor segmentation errors and articulated motion has also been tested and shows promising results. Future work will include using it for shape decomposition, representation, and matching.

References

1. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(4) (2002) 509–522
2. Siddiqi, K., Shokoufandeh, A., Dickinson, S., Zucker, S.W.: Shock graphs and shape matching. *International Journal of Computer Vision* **30** (1999) 1–24
3. Felzenszwalb, P.F.: Representation and detection of deformable shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(2) (2005) 208–220
4. Mori, G.: Guiding model search using segmentation. In: *ICCV.* (2005) 1417–1423
5. Rosenfeld, A.: A note on ‘geometric transforms’ of digital sets. *Pattern Recognition Letters* **1**(4) (1983) 223–225
6. Ogniewicz, R.L., Kübler, O.: Hierarchic voronoi skeletons. *Pattern Recognition* **28**(3) (1995) 343–359
7. Borgefors, G., Nyström, I., Sanniti Di Baja, G.: Computing skeletons in three dimensions. *Pattern Recognition* **37**(7) (1999) 1225–1236
8. Haxhimusa, Y., Ion, A., Kropatsch, W.G., Illetschko, T.: Evaluating minimum spanning tree based segmentation algorithms. In: *CAIP.* (2005) 579–586
9. Gorelick, L., Galun, M., Sharon, E., Basri, R., Brandt, A.: Shape representation and classification using the poisson equation. In: *CVPR* (2). (2004) 61–67
10. Yamashita, M., Ibaraki, T.: Distances defined by neighborhood sequences. *Pattern Recognition* **19** (1986) 237–246
11. Klette, R., Rosenfeld, A.: *Digital Geometry.* Morgan Kaufmann (2004)
12. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. *Journal of Association of Computer Machinery* **13**(4) (1966) 471–494
13. Borgefors, G.: Distance transformation in digital images. *Computer Vision, Graphics, and Image Processing* **34** (1986) 344–371
14. Kropatsch, W.G., Haxhimusa, Y., Pizlo, Z.: Integral trees: Subtree depth and diameter. In: *IWCIA 2004.* Volume 3322 of *LNCS.*, Springer (2004) 77–87
15. Buckley, F., Harary, F.: *Distances in Graphs.* Addison-Wesley Publishing Company (1990)
16. Thulasiraman, K., Swamy, M.N.S.: *Graphs: Theory and Algorithms.* Wiley-Interscience (1992)