Mapping a Coordinate System to a Non-rigid Shape[1]

Adrian Ion, Walter G. Kropatsch
Pattern Recognition and Image Processing Group 183/2, Vienna University of Technology
{ion,krw}@prip.tuwien.ac.at

Abstract
This paper presents a method to decompose binary shapes into connected parts, based on their structure, as captured by the eccentricity transform. This decomposition is then used in a graph pyramid framework for mapping a polar-like coordinate system to a non-rigid shape. Initial experimental results are presented.

1 Introduction

The usual output of shape matching methods is a shape similarity value (see e.g. [1, 3, 4, 12, 17, 14, 6]). Some also give correspondences of the used shape signature, usually border points/parts (see e.g. [12, 1, 17]), but finding all point correspondences based on the obtained information is in most of the cases not straightforward.

This paper presents a concept for using structure to map a coordinate system to an articulated shape, with the purpose of addressing the corresponding (or a close) point in the same or other instances of the articulated shape. It is mainly motivated by observations like: 'one might change his shirt/t-shirt, changing his aspect, or alter his pose a little, but the wristwatch is still located in the same place'.

If thinking of finding the correspondences of all points of the shape, the task is similar to the non-rigid registration problem widely used in the medical image processing community [2]. Differences to our approach include the usage of gray scale information, to compute the deformation vs. the usage of a binary shape and, the registration of a whole image (in most cases in the medical community) vs. the registration of a (in this paper) connected 2D shape.

The method in [3] uses a triangulation of the binary shape as a model, and the produced triangulation correspondence could probably also be used to find corresponding points. An a priory known model would still be needed for the shape class.

In this paper, we propose to use the Euclidean eccentricity transform [5] as a basis for a 2D polar like coordinate system. To support the mapping of the coordinates, a method for decomposing a shape into connected parts is first introduced. Section 2 recalls the eccentricity transform and graph pyramids and their properties relevant for this paper. Sections 3 and 4 describe the proposed methods, with the experiments given in Section 5. Section 6 concludes the paper.

[1] Supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13.

2 Recall

In this section basic definitions and properties of the eccentricity transform and graph pyramids are given.

2.1. Recall ECC

The following definitions and properties follow [5]. Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^2$ and $\partial\mathcal{S}$ be its border[2]. A (geodesic) path π 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 $\lambda(\pi)$ of the shortest path $\pi \in \Pi(\mathbf{p_1}, \mathbf{p_2})$ between $\mathbf{p_1}$ and $\mathbf{p_2}$:

$$d(\mathbf{p_1}, \mathbf{p_2}) = \min\{\lambda(\pi(\mathbf{p_1}, \mathbf{p_2}))|\pi \in \Pi\} \text{ where } \lambda(\pi(t)) = \int_0^1 |\dot{\pi}(t)|dt \quad (1)$$

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, $\forall \mathbf{p} \in \mathcal{S}$

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

i.e. to each point $\mathbf{p}$ it assigns the length of the shortest geodesics to the points farthest away from it. In [11] 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).

This paper considers the class of 4-connected discrete shapes $\mathcal{S}$ defined by points on a square grid $\mathbb{Z}^2$. Paths need to be contained in the area of $\mathbb{R}^2$ defined by the union of the support squares for the pixels of $\mathcal{S}$. The distance between any two pixels whose connecting segment is contained in $\mathcal{S}$ is computed using the ℓ_2-*norm*.

Computation:
In [5] efficient approximation and computation algorithms are presented. The shape bounded single source distance transform, $DT(\mathcal{S}, \mathbf{p})$, computes the geodesic distance of all points of a shape $\mathcal{S}$ to the point $\mathbf{p}$, and is the main tool used for computing $ECC(\mathcal{S})$. $DT(\mathcal{S}, \mathbf{p})$ can be efficiently computed using discrete circles [5] or fast marching [16].

Terminology:
An *eccentric point* of a shape $\mathcal{S}$ is a point $\mathbf{e} \in \mathcal{S}$ that is farthest away in $\mathcal{S}$ from at least one other point $\mathbf{p} \in \mathcal{S}$ i.e. $\exists \mathbf{p} \in \mathcal{S}$ s.t. $ECC(\mathcal{S}, \mathbf{p}) = d(\mathbf{p}, \mathbf{e})$.

The *center* $C \subseteq \mathcal{S}$ of a shape $\mathcal{S}$ is the set of points $\mathbf{c} \in C$ with the smallest eccentricity i.e. $\mathbf{c} \in C \iff ECC(\mathcal{S}, \mathbf{c}) = min\{ECC(\mathcal{S}, \mathbf{p}), \forall \mathbf{p} \in \mathcal{S}\}$. If the shape $\mathcal{S}$ is a simply connected shape, the center C is a single point. Otherwise it can be a disconnected set of arbitrary size (e.g. for $\mathcal{S}$ made out of the points on a circle, all points are eccentric and they all make up the center).

The smallest eccentricity is called the *radius* of the shape, and the highest one is called the *diameter*.

[2] This definition can be generalized to any dimension, continuous and discrete objects.

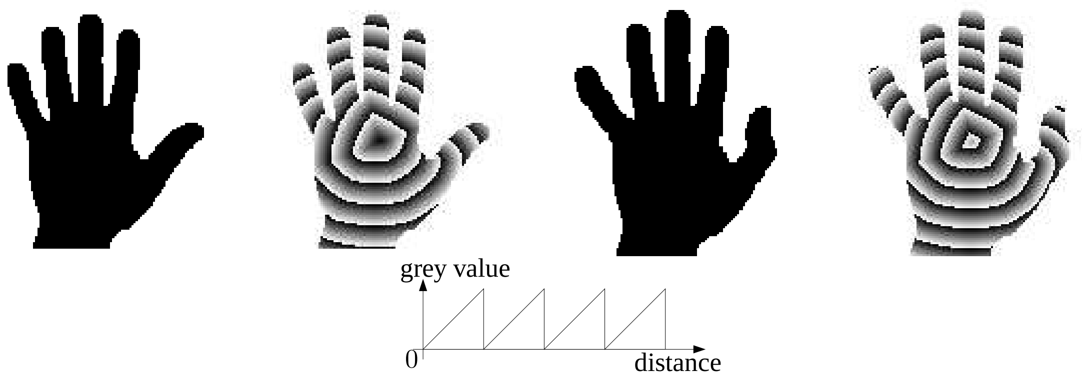


Figure 1. Two shapes and their eccentricity transform.

Properties:
The variation of geodesic distances is bounded under articulated deformation to the width of the 'joints' [12]. The eccentricity transform uses geodesic distances and is bounded in the same way.

The transform is very robust with respect to noise, and the positions of eccentric points and the center are stable [11]. They change only if all supporting pixels for a certain eccentric point are removed or if the diameter changes i.e. the shape is modified around the points with the highest eccentricity.

See Figure 1 for examples of shapes and their eccentricity transform.

2.2. Irregular Graph Pyramids

A *graph pyramid* P [7] is a stack of successively reduced graphs $P = \{G_0, \ldots, G_h\}$. Each level $G_k = (V_k, E_k)$, $1 \leq k \leq h$, is obtained by *contracting* and *removing* edges in the level G_{k-1} below. Successive levels reduce the size of the data by a reduction factor $\lambda > 1$. Edges and vertices of the graphs G_k can be weighted.

The *reduction window* relates a cell at the reduced level with a set of cells in the level directly below. The contents of a lower resolution (in a higher level) cell are computed by means of a *reduction function*, the input of which are the descriptions of the cells in the reduction window. Higher level descriptions should be related to the original input data in the base of the pyramid. This is done by the *receptive field* of a given cell $v \in G_k$. The receptive field of v aggregates all cells in the base level of which v is an ancestor.

Each level represents a partition of the base level into connected subgraphs i.e. *connected subsets of pixels*, if the pyramid is build in the context of an image. The construction of an irregular pyramid is iteratively local [13]. On the base level (level 0) of an irregular pyramid the cells represent single pixels and the neighborhood of the cells is defined by the 4/6/8-connectivity of the pixels. A cell on level k + 1 (parent) is a union of neighboring cells in level k (children). This union is controlled by so called contraction kernels (CK) [9], a spanning forest which relates two successive levels of a pyramid. Every parent computes its values independently of other cells on the same level. Thus local independent (and parallel) processes propagate information up and down and laterally in the pyramid.

In [10], methods for optimally building irregular pyramids are presented. Methods like MIS and

MIES ensure logarithmic height by choosing efficient contraction kernels i.e. contraction kernels achieving high reduction factors.

3 Shape Decomposition based on ECC isoheight Lines

The *level set* of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ corresponding to a real value h is the set of points $\{\mathbf{p} \in \mathbb{R}^n : f(\mathbf{p}) = h\}$ [19]. If $n = 2$ the level set is a set of plane curves.

A *height level* of the eccentricity transform of $\mathcal{S}$ is the level set:

$$HL(e) = \{\mathbf{q} \in \mathcal{S} | ECC(\mathcal{S}, \mathbf{q}) = e\}, \tag{3}$$

with $e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}]$ an eccentricity value. $HL(e)$ can be a single closed curve or a set of disconnected open curves. The connected components of $HL(e)$ are called *isoheight lines*, $HS \subseteq HL(e)$, HS connected.

Given a shape $\mathcal{S}$, a *decomposition of $\mathcal{S}$ into simply connected regions* is the set $\{R_1, \ldots, R_n\}$ s.t. $\bigcup R_k = \mathcal{S}, k = 1, \ldots, n$; $R_i \bigcap R_j = \emptyset, \forall i, j \in \{1, \ldots, n\}$; and R_i is a simply connected region.

$HD(\mathcal{S}) = \{R_1, \ldots, R_n\}$ is a *a decomposition of $\mathcal{S}$ based on the connectivity of the ECC isoheight lines* if HD is a decomposition into simply connected regions (as defined above), and $\forall R_i$ and $\forall e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}] \Rightarrow R_i \bigcap HL(e)$ is a connected region, and n, the number of regions, is **minimal**. $HD(\mathcal{S})$ exists for any connected shape $\mathcal{S}$.

Given a discrete shape $\mathcal{S}$ and its eccentricity transform $ECC(\mathcal{S})$, $HD(\mathcal{S})$ can be computed by:

1. **in a graph theoretical framework:** Algorithm 1 creates a graph pyramid s.t. the top level G_h is an oriented region adjacency graph describing the topology of the decomposition $HD(\mathcal{S})$. Edges of G_h are oriented from regions with lower eccentricity to regions with higher eccentricity. Each vertex contains the length of the longest isoheight segment in its receptive field.
2. **sequential approach:** 'follow' the isoheight lines from the minimum eccentricity to the maximum eccentricity. Whenever an isoheight line gets disconnected, or merged, new regions are started for the formed isoheight line part(s). (This approach is more intuitive, but needs building the adjacency graph for the decomposition 'over' it. In addition, it is lacking the fast access advantages when searching for the pixel with a known coordinate.).

Figure 3 shows example decompositions based on the connectivity of isoheight lines.

If the shape $\mathcal{S}$ is simply connected, the obtained region adjacency graph (top level of the pyramid) is a tree (Theorem 7.9 in [8]), with the root being the vertex whose receptive field contains the (unique) center pixel. Also, the edges oriented toward each vertex and the ones oriented away are nicely grouped together.

Note that such a decomposition can be done in the same way, for other transforms also (e.g. the single source shape bounded distance transform). We use the eccentricity transform because its center is a stable [11] and there is no need to give a starting point.

Algorithm 1 HD - Decomposition of $\mathcal{S}$ based on the connectivity of ECC isoheight lines

Input: Discrete shape $\mathcal{S}$.

$iECC = \lfloor ECC(\mathcal{S}) \rfloor$ */*compute ECC, round = at least 8 connected isoheight lines*/*
$G_0 \leftarrow$ oriented neighborhood graph of $iECC$ */*ensure proper connectivity of isoheight lines while keeping G_0 planar, orient from small to high eccentricity*/*
$k \leftarrow 0$
for all $v \in V_k$ **do** $v.maxlength \leftarrow 1, v.ecc \leftarrow [ECC(v), ECC(v)]$, */*init max length of isoheight lines and eccentricity interval for each vertex*/*
repeat
    $A \leftarrow \{e = (v_i, v_j) \in E_k | v_i.ecc = v_j.ecc\}$ */*merge isoheight segment parts*/*
    $A \leftarrow A \bigcup \{e = (v_i, v_j) |deg(v_i)$,deg$(v_j) \leq 2$ and closed(v_i)=closed$(v_j)\}$ */*same region, closed(v)=true $\iff$ receptive field of v contains only closed isoheight lines*/*
    if $|A| > 0$ **then**
        $K \leftarrow$ contraction kernels as subset of A */*use MIS or MIES [10] to optimally break A into valid contraction kernels*/*
        $G_{k+1} \leftarrow$ contract(G_k, K) */*contract edges in K and simplify*/*
        for all $v \in V_{k+1}$ **do** compute $v.maxlength, v.ecc$ from G_k */*use reduction window*/*
        $k \leftarrow k + 1$
    end if
until $|A| = 0$
$h \leftarrow k$

Output: Graph Pyramid $G_0, \ldots, G_h$.

Motivated by the need to match partially occluded shapes, 2D shape matching related research has recently moved toward shape decomposition and part matching (e.g. [14]). A study of the decomposition of shapes based on ECC isoheight lines in the context of shape matching is planned, but is outside the scope of this paper.

4 The Non-rigid Coordinate System

A system of *curvilinear coordinates* [18] is a coordinate system composed of intersecting surfaces. If the intersections are all at right angles, then the curvilinear coordinates are said to form an *orthogonal coordinate system* (e.g. two-dimensional Cartesian coordinates and polar coordinates). If not, they form a *skew coordinate system*.

Based on the above, to define a planar system of curvilinear coordinates, two classes of curves need to be defined - one for each coordinate. For any point $\mathbf{p} \in \mathcal{S}$ there exists exactly one curve of each class passing through it. Also, any defined coordinates identify one curve of each class, and the intersection of the two curves gives a unique point.

The proposed coordinate system is intuitively similar to the polar coordinate system, with the *radial coordinate* ***r*** being a linear mapping from the eccentricity value and the *angular coordinate* θ being mapped to the isoheight lines of the eccentricity transform based on the structure of the shape. The first approach presented in this paper forms a skew coordinate system (see Section 4.1.). The second approach has disconnected 'angular' coordinate curves, thus it does not correspond to the definition of curvilinear coordinates. Note that θ is not really an angle, just denoted intuitively so.

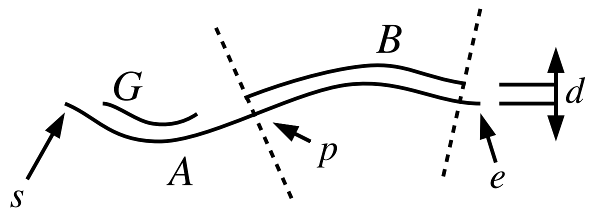


Figure 2. Mapping of points from neighboring isoheight lines

$$\boldsymbol{r}(\mathbf{p}) = ECC(\mathcal{S}, \mathbf{p}) - \min\{ECC(\mathcal{S}, \mathbf{p})\} \tag{4}$$

Figure 3 shows the isoheight lines of the eccentricity transform i.e. of $\boldsymbol{r}(\mathbf{p})$.

4.1. Setting the angular coordinate

As mentioned above, the angular coordinate θ is not really an angle - it has been intuitively named like this. This section focuses on simply connected shapes and their properties. For non simply connected shapes, the result of the decomposition is much more complex (general graph with cycles, etc.) and more complex algorithms are required.

Figure 2 shows three adjacent isoheight segments (A, B, G) of different regions. A has eccentricity e, and B, G have $e + k$. If $k \to 0$ then $d \to 0$, and maximum smoothness of θ is achieved when each point of B has the same θ as his projection on A. This assumption puts the values θ for A and B into relation. An approximation can be made by projecting the endpoints of B onto A, to find their θ values, and interpolating along B (see Figure 2 for the notation):

$$\theta_1' = \theta_1 + \frac{(\theta_2 - \theta_1) \int_s^p dl}{\int_s^e dl} \tag{5}$$

The obtained relation can be used to controll the smoothness of θ along region boundaries.

In the following, two algorithms for assigning θ are presented.

Center to Periphery

The root vertex of $G = G_h$ from Section 3, contains only closed isoheight lines and is the only such vertex. The angle interval associated to vertices with closed isoheight lines is 360 degrees. The other vertices have an associated 'input interval' and 0 or more 'output intervals' (edge orientation in G). Smoothness along region boundaries is assumed as above, and intervals of θ inside each region are kept constant.

Algorithm 2 shows the algorithm for assigning the θ intervals to each vertex. These can then be down projected in the pyramid. The Algorithm should be called with the top level of the pyramid in Algorithm 1 as the parameter G, the root vertex of the tree G as v, and $[0, 360]$ as $[\theta_1, \theta_2]$.

Algorithm 2 $CtoP$ - Assign real valued intervals for θ, for all vertices of G

Input: Graph $G(V, E)$ as produced by Algorithm 1, vertex v, interval $[\theta_1, \theta_2]$.

```
v.θ1 ← θ1, v.θ2 ← θ2
A ← isoheight segment of v with highest eccentricity
/*for all edges oriented away from v*/
for all e = (v, v_o) ∈ E do
    B ← isoheight segment of v_o with lowest eccentricity
    [θ'1, θ'2] ← project B to A and compute from [θ1, θ2] as in Equation 5 and Figure 2
    call CtoP(G, v_o, [θ'1, θ'2])
end for
```

Output: Graph G, with θ intervals $[v.\theta_1, v.\theta_2]$ computed for each region

This approach works only with real valued θ, as two isoheight segments of the same region can contain a different number of pixels and still get the same interval assigned.

For the origin of θ, a path connecting the center (minimum eccentricity) with a point having the maximum eccentricity can be used. This path is called the *zero path*. Note that the zero path does not necessarely have to be a part of the diamether, as the diamether does not always pass through the center. The zero path is used in the inner most region (root vertex of G_h) to set the 0 for the $theta$ of each isoheight line. Outside this region, the propagation of θ and linear interpolation, as described above, are applied. The point with maximum eccentricity can be given, or automatically chosen using any of the existing shape orientation methods, see for example [20], even though an orientation method taking into consideration the desired deformation freedom would be optimal.

Periphery to Center

An ordering of the pixels in each isoheight line can be used to assign integer values of θ to each pixel.

During the process of decomposing the shape into regions (see Section 3), each vertex is assigned the highest number of pixels in an isoheight segment included in its receptive field.

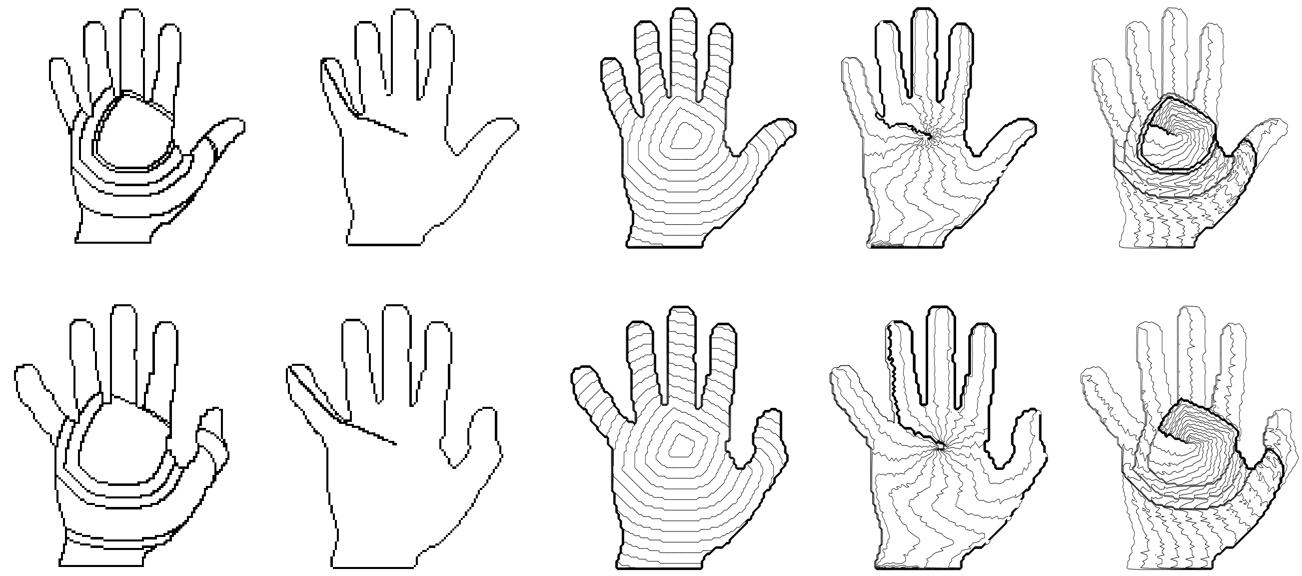
Algorithm 3 starts from the leaves of the RAG of the decomposition of $\mathcal{S}$ and propagates the allocated

Algorithm 3 $PtoC$ - Assign integer intervals for θ for all vertices of G

Input: Graph $G(V, E)$ as produced by Algorithm 1.

```
for all v ∈ V, deg(v) = 1 do v.widthθ = v.maxwidth,
repeat
    for all v_o ∈ V, with v_o.θ computed ∀(v, v_o) ∈ E do
        v.widthθ = max{v.maxwidth, Σ_{(v,v_o)∈E} v_o.widthθ} /*compute maximum number of values required for θ for the subtree rooted at v*/
    end for
until all v.widthθ are computed
having the interval width v.θ2 − v.θ1 = v.widthθ for all v ∈ V, compute v.θ1 (the interval beginning) starting with the root, like in Algorithm 2
```

Output: Graph G, with θ intervals $[v.\theta_1, v.\theta_2]$ computed for each region



decomposition zero path radial coordinate ***r*** $\theta\ (CtoP)$ $\theta\ (PtoC)$

Figure 3. Decomposition, zero path, and isolines for the two shapes in Figure 1.

values to the center. A 'zero' value has to still be decided for (see section above). The value of θ for each pixel can be computed by doing an additional parsing from the root to all vertices, and then, down the pyramid to all pixels.

5 Experiments

An implementation for Algorithms 1, 2, and 3 has been made. Figure 3 shows for two poses of a hand from the Kimia99 database [15] their segmentation into parts (Algorithm 1). It also shows the used zero path, the isoheight lines of the radial coordinate ***r***, derived from the eccentricity transform, and isoheight lines of the two mappings for θ computed with Algorithms 2 and 3.

In the case of Algorithm 2, the jagged isoheight lines of θ are due to the smoothness/roughness of the shape boundary i.e. curvature of the shape boundary at the endpoints of isoheight lines, and partly due to the simple implementation (point projection by closest point search, integral along line estimation by sum of line segment lengths, etc.).

In the case of Algorithm 3, the jagged isoheight lines of θ are due to the smoothness/roughness of the shape boundary. Around the region boundaries, it is due to the way integer values for θ are set (in this case, θ is not smooth over the region boundaries). Correspondences between connected subparts of the shapes have to be found in order to find correspondences between the integer θ values.

In both cases improvements can be made by a more global decision of the θ interval allocated for each region and each isoheight line.

Quantitative error measurements for the mapping from one pose to the other are planned.

6 Conclusion and Outlook

This paper presents a concept for mapping a polar-like coordinate system to a non-rigid binary shape. Initial experimental results are presented. More global decisions can be used to obtain smoother angular isoheight lines, and additional correspondences between part structures can help to solve failed correspondences. Further quantitative evaluation and extension to non simply connected shapes is planned.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions Pattern Analysis Machine Intelligence*, 24(4):509–522, 2002.

[2] W. R. Crum, T. Hartkens, and D. L. Hill. Non-rigid image registration: theory and practice. *The British Journal of Radiology*, 77 Spec No 2, 2004.

[3] Pedro F. Felzenszwalb. Representation and detection of deformable shapes. In *CVPR (1)*, pages 102–108. IEEE Computer Society, 2003.

[4] Pedro F. Felzenszwalb and Joshua D. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*. IEEE Computer Society, 2007.

[5] Adrian Ion, Walter G. Kropatsch, and Eric Andres. Euclidean eccentricity transform by discrete arc paving. In *14th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI)*, Lecture Notes in Computer Science, Lyon, France, April 2008. Springer.

[6] Adrian Ion, Gabriel Peyré, Yll Haxhimusa, Samuel Peltier, Walter G. Kropatsch, and Laurent Cohen. Shape matching using the geodesic eccentricity transform - a study. In *The 31st Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR)*, pages 97–104, Schloss Krumbach, Austria, May 2007. OCG.

[7] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer, 1994.

[8] Reinhard Klette and Azriel Rosenfeld. *Digital Geometry*. Morgan Kaufmann, 2004.

[9] Walter G. Kropatsch. Building irregular pyramids by dual graph contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366–374, December 1995.

[10] Walter G. Kropatsch, Yll Haxhimusa, Zygmunt Pizlo, and Georg Langs. Vision pyramids that do not grow too high. *Pattern Recognition Letters*, 26(3):319–337, 2005.

[11] Walter G. Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer. The eccentricity transform (of a digital shape). In *13th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI)*, Lecture Notes in Computer Science, pages 437–448, Hungary, October 2006. Springer.

[12] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions Pattern Analysis Machine Intelligence*, 29(2):286–299, 2007.

[13] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, March 1989. Also as UM CS TR-1871, June, 1987.

[14] Ozge C. Ozcanli and Benjamin B. Kimia. Generic object recognition via shock patch fragments. In *BMVC'07*, pages 1030–1039. Warwick Print, 2007.

[15] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions Pattern Analysis Machine Intelligence*, 26(5):550–571, 2004.

[16] James A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[17] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.

[18] Eric W. Weisstein. "curvilinear coordinates." from mathworld–a wolfram web resource. http://mathworld.wolfram.com/CurvilinearCoordinates.html.

[19] Eric W. Weisstein. "level set." from mathworld–a wolfram web resource. http://mathworld.wolfram.com/LevelSet.html.

[20] Jovisa D. Zunic, Paul L. Rosin, and Lazar Kopanja. On the orientability of shapes. *IEEE Transactions on Image Processing*, 15(11):3478–3487, 2006.