Computer Vision Winter Workshop 2009, Adrian Ion and Walter G. Kropatsch (eds.)
Eibiswald, Austria, February 4–6
Publisher: PRIP, Vienna University of Technology, Austria

A Coordinate System for Articulated 2D Shape Point Correspondences*

(further aligning theta)

Adrian Ion, Yll Haxhimusa, and Walter G. Kropatsch

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

Abstract *A framework for mapping a polar-like coordinate system to a non-rigid shape is presented. Using a graph pyramid, a binary shape is decomposed into connected parts, based on its structure as captured by the eccentricity transform. The decomposition is used to derive domains for the angular like coordinate. A closest point search is employed to find point correspondences. We discuss aligning the angular like coordinate to optimize the mapping quality.*

1 Introduction

Most shape matching methods output a similarity value (e.g. [4, 5, 17, 8]), some also give correspondences of the used signature, usually border points/parts [15, 1, 21], but finding all point correspondences based on the obtained information is in most of the cases not straightforward.

This work maps a coordinate system to an articulated shape, with the purpose of addressing the corresponding point (or a close one) in other instances of the same shape. It is mainly motivated by observations like: 'one might change his aspect, alter his pose, but the wristwatch is still located in the same place on the hand'.

For correspondences of all points of the shape, the task is similar to the non-rigid registration problem used in the medical image processing community [3]. Differences 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) vs. the registration of a (in this approach) connected 2D shape. In [4], a triangulation of the shape is used as a model, which could be used to find corresponding points, but an a priory known model is still needed. In the surface parametrization community [2] a coordinate system for shapes is defined, but articulation is not considered. In [10], for small variations, correspondences between points of 3D articulated shapes are found. Recently shape matching has also moved toward decomposition and part matching, e.g. [17], mainly due to occlusions, imperfect segmentation or feature detection.

In this paper, we use the Euclidean eccentricity transform [7] 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. This paper is an extention of [6] and disuses in detail aligning the angular like coordinate (θ), Section 6. Refinements have been made throughout the whole paper.

The structure of the paper is as follows. 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, followed by discussion in Section 6. Section 7 concludes the paper.

2 Recall

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

2.1 Eccentricity Transform

The following definitions and properties follow [7].

Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^2$ and $\partial\mathcal{S}$ be its boundary[1]. 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 within the set $\mathcal{S}$, between two points $\mathbf{p_1},\mathbf{p_2}\in\mathcal{S}$.

The *geodesic distance* $d(\mathbf{p_1},\mathbf{p_2})$ between $\mathbf{p_1},\mathbf{p_2}$ is defined as the length $\lambda(\pi)$ of the shortest path $\pi\in\Pi(\mathbf{p_1},\mathbf{p_2})$, more formally:

$$d(\mathbf{p_1},\mathbf{p_2}) = \min\{\lambda(\pi(\mathbf{p_1},\mathbf{p_2}))|\pi\in\Pi\}, \qquad (1)$$

where $\lambda(\pi(t)) = \int_0^1 |\dot{\pi}(t)|dt$, $\pi(t)$ is a parametrization of the path from $\mathbf{p_1} = \pi(0)$ to $\mathbf{p_2} = \pi(1)$, and $\dot{\pi}(t)$ is the differential of the arc length. A path $\pi\in\Pi(\mathbf{p_1},\mathbf{p_2})$ with $\lambda(\pi) = d(\mathbf{p}_1,\mathbf{p}_2)$ is called a *geodesic*.

The *eccentricity transform* (ECC) of $\mathcal{S}$ can be defined as:

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

$\forall\mathbf{p}\in\mathcal{S}$ i.e. to each point $\mathbf{p}$ it assigns the length of the shortest geodesic path(s) to the points farthest away. The transform is quasi-invariant to articulated motion and robust against Salt & Pepper noise [14] (which creates holes in the shape).

* Partially supported by the Austrian Science Fund under grant S9103-N13.

[1] It can be generalized to any continuous and discrete space.

In this paper, the class of 4-connected, planar, and simply connected discrete shapes $\mathcal{S}$ defined by points on the square grid $\mathbb{Z}^2$ are considered. Paths are 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 [7], efficient computation algorithms are given. The shape bounded single source distance transform[2], $DT(\mathcal{S}, \mathbf{p})$, 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 [7] or fast marching [20].

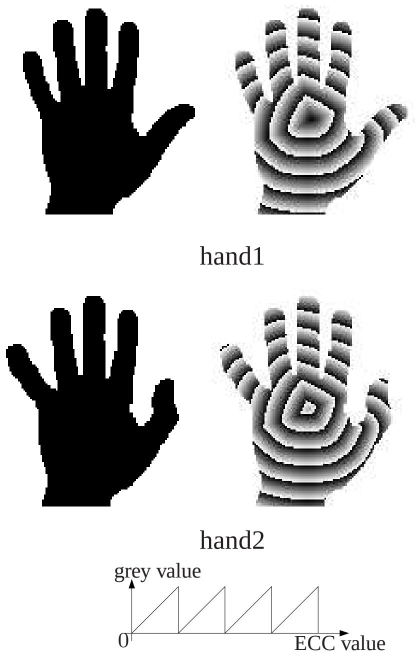
Terminology: An *eccentric point* of a shape $\mathcal{S}$ is a point $\mathbf{e} \in \mathcal{S}$ that is farthest away from at least one other point $\mathbf{p} \in \mathcal{S}$ i.e. $\exists \mathbf{p} \in \mathcal{S}$ such that $ECC(\mathcal{S}, \mathbf{p}) = d(\mathbf{p}, \mathbf{e})$. The *center* $\mathcal{C} \subseteq \mathcal{S}$ is the set of points with the smallest eccentricity i.e. $\mathbf{c} \in \mathcal{C}$ iff $ECC(\mathcal{S}, \mathbf{c}) = min\{ECC(\mathcal{S}, \mathbf{p}) \mid \forall \mathbf{p} \in \mathcal{S}\}$. If $\mathcal{S}$ is simply connected, $\mathcal{C}$ is a single point. Otherwise it can be a disconnected set of arbitrary size (e.g. for $\mathcal{S}$ = 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*.

Properties: Due to using geodesic distances, the variation of ECC is bounded under articulated deformation to the width of the 'joints' [15]. The transform is robust with respect to Salt & Pepper noise, and the positions of eccentric points and center are stable [14]. Figure 1 shows two hand shapes (taken from the Kimia99 database [19]) and their eccentricity transform.

2.2 Irregular Graph Pyramids

A *graph pyramid* [9] $P = \{G_0, \ldots, G_t\}$ is a stack of successively reduced graphs ($G_i, i = 1, \ldots, t$, where G_0 is the base level, and G_t is the top of the pyramid). Each level $G_k = (V_k, E_k)$, $1 \leq k \leq t$, is obtained by *contracting* and *removing* edges in the level G_{k-1} below. Successive levels reduce the size of the data by $\gamma > 1$. Edges and vertices of G_k can be attributed. The *reduction window* relates a vertex at a level G_k with a set of vertices in the level directly below (G_{k-1}). Higher level descriptions are related to the original input data by so called the *receptive field* (RF) of a given vertex $v \in G_k$, which aggregates all vertices in G_0 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 [16]. In the base level G_0 of an irregular pyramid the vertices represent single pixels and the neighborhood of the cells is defined by the 4-connectivity of the pixels (higher connectivity can be used locally, but graph planarity should be kept). The union of neighboring vertices on level $k-1$ (children) to a vertex on level k (parent) is controlled by trees called *contraction kernels* (CK) [12] chosen by the algorithm (e.g. segmentation, connected component labeling, etc.). Every vertex computes its values independently of other vertices on the same level. Thus local independent (and parallel) processes propagate information up and down and laterally in the pyramid [13].


hand1

hand2

Figure 1: Eccentricity transform of two shapes from [19]. The grayvalues are the eccentricity value modulo a constant.

In [13], methods for optimally building irregular pyramids are presented. Methods like MIS and MIES ensure logarithmic height of the pyramid by choosing efficient contraction kernels i.e. contraction kernels achieving high reduction factors.

3 ECC Isoheight Lines - Decomposition

The *level set* [23] of a function $f : \mathbb{R}^n \to \mathbb{R}$, corresponding to a value h, is the set of points $\mathbf{p} \in \mathbb{R}^n$ such that $f(\mathbf{p}) = h$.

A *level set* of the ECC of $\mathcal{S}$ is the set

$$LS(e) = \{\mathbf{q} \in \mathcal{S} \mid ECC(\mathcal{S}, \mathbf{q}) = e\},$$

with $e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}]$. For $\mathcal{S} \in \mathbb{R}^2$, $LS(e)$ can be a closed curve or a set of disconnected open curves. The connected components of $LS(e)$ are called *isoheight lines*, $IL \subseteq LS(e)$, IL is connected.

For a shape $\mathcal{S}$, $HD(\mathcal{S}) = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ is a *a decomposition of $\mathcal{S}$ based on the connectivity of the ECC isoheight lines* (Figure 3) if:

- HD is a partition of $\mathcal{S}$ into simply connected regions;
- $\forall \mathcal{R}_i$ and $\forall e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}]$ $\Rightarrow \mathcal{R}_i \bigcap LS(e)$ is connected; and
- the number n of regions is **minimal**.

[2] Also called *geodesic distance function* [22].

Algorithm 1 HD - Decompose $\mathcal{S}$ based on ECC LS

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

```
$iECC = \lfloor ECC(\mathcal{S}) \rfloor$ /*at least 8 connected $IL$*/
$G_0 \leftarrow$ oriented neighborhood graph of $iECC$
/* pixels with same iECC connected, $G_0$ planar,
orient from small to high iECC*/
$k \leftarrow 0$
$\forall v \in V_0$ do
$v.maxlength \leftarrow 1, v.ecc \leftarrow [ECC(v), ECC(v)]$
/* init max length of isoheight lines and ecc. interval*/
repeat
    $A \leftarrow \{e = (v, w) \in E_k \mid v.ecc = w.ecc\}$
    /* merge isoheight line parts*/
    $A \leftarrow A \bigcup \{e = (v, w) \mid$ out-deg$(v)$ = in-deg$(w) = 1$ and
    closed$(v)$=closed$(w)\}$
    /* closed(v)=true iff RF(v) contains only closed $IL$*/
    if $|A| > 0$ then
        $K \leftarrow$ CK as subset of $A$
        /*choose optimal subset of A with e.g. MIS [13] */
        $G_{k+1} \leftarrow$ contract$(G_k, K)$ /* also simplify*/
        $\forall v \in V_{k+1}$ compute $v.maxlength, v.ecc$ from $G_k$ /*
        use reduction window*/
        $k \leftarrow k + 1$
    end if
until $|A| = 0$
$t \leftarrow k$
```

Output: Graph Pyramid $P = \{G_0, \ldots, G_t\}$.

$HD(\mathcal{S})$ exists for any connected shape $\mathcal{S}$.

The top level G_t of the graph pyramid created by Algorithm 1 is a region adjacency graph describing the topology of the decomposition $HD(\mathcal{S})$. Edges of G_t are oriented from regions with lower eccentricity to regions with higher eccentricity. Each vertex contains the length of the longest isoheight line in its RF. The result is similar to building the Reeb graph [18] of $\mathcal{S}$ with $ECC(\mathcal{S})$ as the Morse function.

The top level G_t corresponds to the following decomposition: one can imagine following 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.

If $\mathcal{S}$ is simply connected, the obtained region adjacency graph (top level of the pyramid) is a tree (Theorem 7.9 in [11]), with the RF of the root vertex containing the (unique) center pixel. Such a decomposition can be done for other transforms also (e.g. the $DT(\mathcal{S}, \mathbf{p})$). The eccentricity transform is used because its center is a robust starting point [14].

4 The Non-rigid Coordinate System

A system of *curvilinear coordinates* [23] is a system composed of intersecting surfaces. If all intersections are at angle $\pi/2$, then the coordinate system is called *orthogonal* (e.g. polar coordinate system). If not, a *skew* coordinate system is formed.

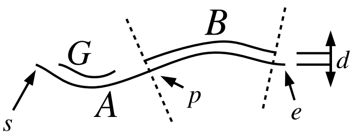


Figure 2: Mapping of points from neighboring isoheight lines

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. Any defined coordinates identify one curve of each class which intersect at a unique point.

The proposed coordinate system is intuitively similar to the polar coordinate system, but forms a skew coordinate system. We focus on simply connected shapes and their properties. The decomposition of non simply connected shapes is much more complex (general graph with cycles, etc.) and more complex algorithms are required. Note that θ is not really an angle, just denoted intuitively so. The *radial coordinate* is a linear mapping from the eccentricity value and the *angular coordinate* θ is mapped to the isoheight lines of the ECC based on the structure of the shape.

The *radial coordinate* is:

$$\boldsymbol{r}(\mathbf{p}) = (ECC(\mathcal{S}, \mathbf{p}) - m)/(M - m), \tag{3}$$

where $m = \min\{ECC(\mathcal{S})\}, M = \max\{ECC(\mathcal{S})\}$. 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. The presented approach 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 lines (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 is to project the endpoints of B onto A, to find their θ values, and interpolate along B:

$$\theta_1' = \theta_1 + \frac{(\theta_2 - \theta_1) \int_s^p dl}{\int_s^e dl} \tag{4}$$

The obtained relation can be used to control the smoothness of θ along region boundaries. Inside regions, the values of θ are computed using linear interpolation along the isoline, from the given start and end values for θ.

The root vertex of G_t (Section 3), contains only closed isoheight lines and is the only such vertex. Its associated θ interval is 2π. Other vertices have an ’input interval’ and 0 or more ’output intervals’ (edge orientation in G). Smoothness along region boundaries is assumed, and intervals of

Algorithm 2 $CtoP$ - Assign θ to $\forall v \in G$

```
Input: $G = (V, E)$ from Algorithm 1, vertex $v$, interval $[\theta_1, \theta_2]$.
$v.\theta_1 \leftarrow \theta_1, v.\theta_2 \leftarrow \theta_2$
$A \leftarrow$ isoheight line of $v$ with highest ecc.
for all $e = (v, v_o) \in E$ /*all edges oriented away*/ do
    $B \leftarrow$ isoheight line of $v_o$ with lowest ecc.
    $[\theta_1', \theta_2'] \leftarrow$ project $B$ to $A$ and compute from $[\theta_1, \theta_2]$ (Equation 4)
    call $CtoP(G, v_o, [\theta_1', \theta_2'])$
end for
Output: $G$, with $\theta$ intervals $[v.\theta_1, v.\theta_2]$ for each region
```

θ inside each region are kept constant. Inside each region, values of θ are interpolated along the isolines as mentioned above. Algorithm 2 assigns the θ intervals to each vertex. The parameters are the top level of the pyramid from Algorithm 1, the root vertex of G_t, and $[0, 2\pi]$. 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*. (the zero path does not have to be a part of the diameter, as the diameter does not always pass through the center). It is used in the inner most region (root vertex of G_t) to set the 0 for the θ of each isoheight line. Outside this region, linear interpolation is used (Equation 4). A point with maximum ECC can be selected using any shape orientation method (e.g. [24]) - taking into consideration the possible deformations would be optimal (see Section 6 for a detailed discussion).

Figure 3 shows the results of Algorithm 1 and 2, and Equation 3 and 4 for the two hands. 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 and integral along line estimation by sum of line segment lengths for Equation 4, etc.).

5 Experiments

To get a feeling of the "stability" of the mapping w.r.t. articulation we have applied the algorithms on the shapes in Figure 1. A pattern was laid on each hand - the *source*, and copied to the other one - the *destination*, by finding for each pixel $p_d(r_d, \theta_d)$ of the *destination* the "closest" pixel $p_s(r_s, \theta_s)$ in the *source* (images in the last two rows in Figure 3). The local variation of θ is not constant over the whole shape, making the Euclidean metric not the best option for finding the closest pixel to a given point $p_d(r_d, \theta_d)$.

To avoid compensating for this variation, a two step approach is used. **First**, normalize r in both shapes to $[0, 1]$. This makes finding $ecc_d \rightarrow r \rightarrow ecc_s$ a linear scaling problem. $L \leftarrow (ecc_s \leq ECC(\textit{source}) < ecc_s + 1)$ gives at least 8 connected isoheight lines of r. **Second**, the pixel of L which minimizes $|\theta_d - \theta_s|$ is chosen. The results are promising (see Figure 3) with the texture of the "articulated" finger being nicely copied from one shape to the other i.e. points are copied to their corresponding region in the articulated version of the shape.

The noise like errors on the pattern are due to the approximations mentioned above and to using "nearest point" for finding the color of each pixel when copying the pattern (instead of interpolating gray values). Errors on the boundaries of fingers are due to certain coordinates not existing in both shapes. The more global perturbation (palm of the hands in Figure 3) is mainly due to the slightly different position of the centers and isoheight line shape. Improvements can be made by considering both shapes when mapping the coordinates to them, or by a more complex method for finding corresponding points. Finding a matching between the regions of the decomposition of the two shapes is an important step and is planned in the future.

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

6 Discussion - aligning the angular like coordinate (θ)

The option of connecting the center (point with ECC global minimum) with a point having the maximum ECC is not always the best option. All shapes have at least two points with the maximum ECC and discriminating between them is an open problem. Also depending on the intra-class deformation of the shape and on segmentation errors, these points can shift (usually only in the local neighborhood).

The mapping in Algorithm 2 does not specifically use the zero value. It assigns angle intervals. The zero value is just used to set a starting point (origin) in the region containing the center. Thus, using a zero path that shifts the angles with k degrees is equivalent with computing the mapping on the whole shape, and then shifting the angles with k degrees.

The present coordinate mapping is usefull if wanting to associate points from two different poses of the same shape (for one single shape in the exact same pose, using the Cartesian coordinate system of the space in which the shape is embedded is faster). Instead of fully computing the coordinate system separately on each shape, one could consider aligning the θ values to optimize the mapping quality. For this we compute the coordinate system as proposed in Section 4, using any global ECC maximum to choose the zero path. Following this, we can align θ of one of the shapes (called *primary*) to the other one (called *secondary*), by one of the following options.

The first option is to minimize the number of pixels without any correspondence. As presented in Section 5, the point of the source $\mathbf{p}_s = (\boldsymbol{r}_s, \theta_s) \in \mathcal{S}$ corresponding to a pair of coordinates $(\boldsymbol{r}_d, \theta_d)$ is found by:

$$\mathbf{p}_s = \mathrm{argmin}_{\mathbf{p} \in L}\{|\theta(\mathbf{p}) - \theta_d|\},$$

where L is the at least 8 connected isoline with the eccentricity computed from Equation 3, for a given $\boldsymbol{r}_s = \boldsymbol{r}_d$.

For any $\boldsymbol{r} \in [0, 1]$ we get $L \neq \emptyset$, and the equation above will produce a result. To decide whether a satisfying correspondence has been found, we can consider setting a threshold on the distance between the coordinates requested $(\boldsymbol{r}_d, \theta_d)$ and the coordinates of the point found $(\boldsymbol{r}_s, \theta_s)$. The

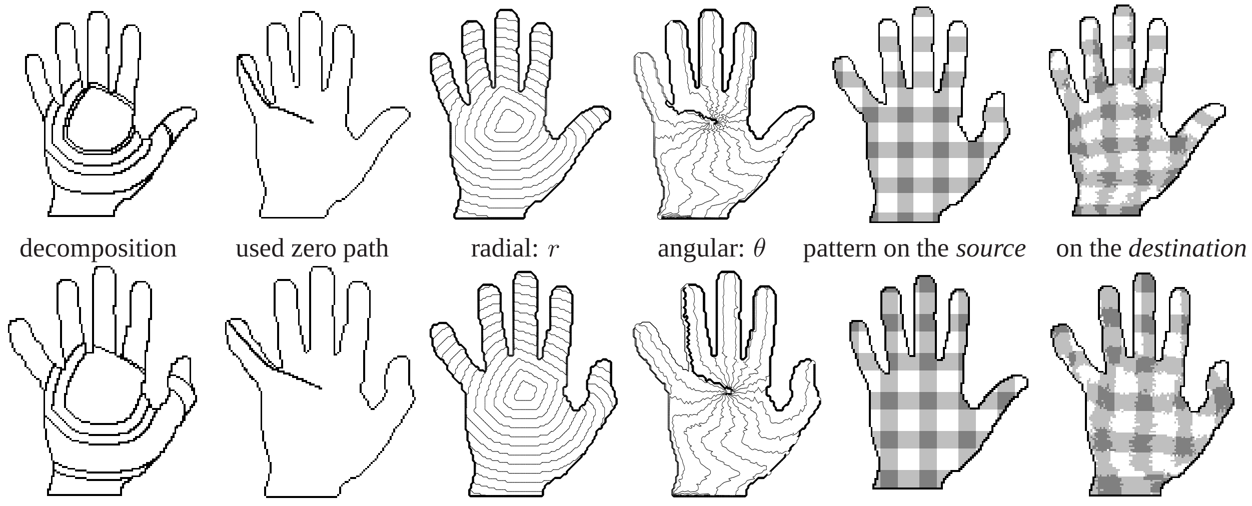


Figure 3: Results for the shapes in Figure 1 (θ increases counter-clockwise).

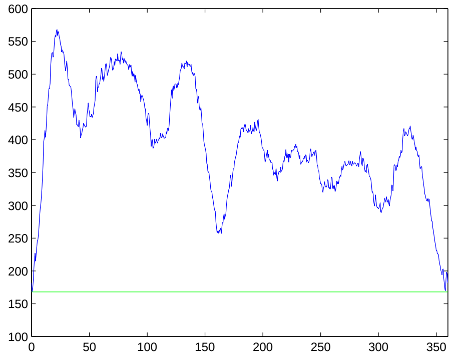


Figure 4: Rotation of θ of hand1 vs. number of pixels without a successful correspondence (minimum value is shown).

L_2-norm cannot be used as $\boldsymbol{r}$ and θ are very different in nature. $\boldsymbol{r}$ has bounded local variation, while the variation of θ can differ a lot between different parts of the same shape (e.g. two fingers of the same hand, one very thing and one many times thicker). Also θ is a periodic value.

As θ values are assigned inside regions using linear interpolation along the isolines (Section 4.1), we can also compute and store the local variation along the isoline. A multiple of this value (in our experiments 2) can be used as the threshold to differentiate between a successful correspondence and a failed one. Figure 4 shows the result of shifting θ of hand1 (left in Figure 1) and counting the number of pixels without a successful correspondence. The source 'role' is played by hand2 and the destination by hand1. In the case of hand1 and hand2 the original angle is very close to the minimum as corresponding ECC maxima where chosen as reference points and their position was stable. The disadvantage of this approach is its computational complexity. For every tested shift of θ, the correspondences for all pixels have to be computed.

To overcome this complexity, one could consider a simpler alignment. Every boundary pixel has an ECC value which if considered along the boundary, produces a 1D signal. One can consider aligning the two 1D signals of the ECC of the boundary points, to obtain a maximum overlap, and shift θ with the offset of the initial and obtained position for a certain boundary point (e.g. the starting one). Figure 5.b shows the ECC 'signals' of the boundary pixels of hand1 and hand2 for starting points with $\theta = 0$ (Figure 5.a). Figure 6 shows the correlation and angle offset of the starting point for different offset positions of the 'circular' 1D signal of hand1 (correlation computed to the 1D signal of hand2). The maximum correlation is achieved for an offset of 41 pixels (clockwise) which brings the starting point of hand1 over the one of hand2, and recommends an offset of -7.98 degrees (θ was assigned counter clockwise). Due to the fact that many boundary points can have the same eccentricity, and for a certain offset in the 1D signal of hand1 there are many shifts in the respective angles of the boundary points, between the original position and the offset-ed one, this approach has to be considered in more detail.

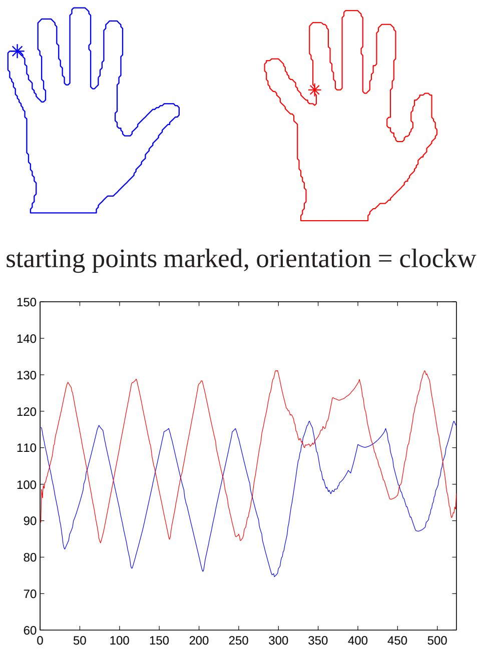
7 Conclusion

In this paper we discuss a framework for using the eccentricity transform to map a polar-like coordinate system onto a non-rigid binary shape and find corresponding points between two shapes. Promising initial results are presented. The alignment of the coordinates is discussed in more detail for optimizing the mapping quality. More global decisions will provide 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 TPAMI*, 24(4):509–522, 2002.

[2] C. Brechbuhler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3-D



a.) starting points marked, orientation = clockwise

b.) corresponding 1D ECC 'signals'.

Figure 5: Boundary points and signals of the two hands (best viewed in color).

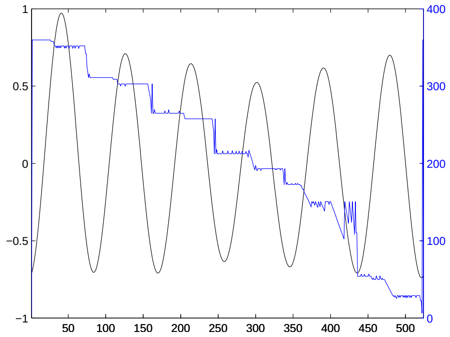


Figure 6: Correlation (black) and angle offset (blue) of starting point, for offset of signal of hand1 (best viewed in color).

shape-description. *Computer Vision and Image Understanding*, 61(2):154–170, Mar. 1995.

[3] 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.

[4] Pedro F. Felzenszwalb. Representation and detection of deformable shapes. In *CVPR*, 2003.

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

[6] Adrian Ion, Yll Haxhimusa, Walter G. Kropatsch, and Salvador B. López Mármol. A coordinate system for articulated 2d shape point correspondences. In *Proceedings of 19th International Conference on Pattern Recognition (ICPR)*. IAPR, IEEE, 2008.

[7] Adrian Ion, Walter G. Kropatsch, and Eric Andres. Euclidean eccentricity transform by discrete arc paving. In *14th DGCI*, volume 4992 of *LNCS*, Lyon, France, April 2008. Springer.

[8] Adrian Ion, Gabriel Peyré, Yll Haxhimusa, Samuel Peltier, Walter G. Kropatsch, and Laurent Cohen. Shape matching using the geodesic eccentricity transform - a study. In *31st OAGM/AAPR*, Schloss Krumbach, Austria, 2007. OCG.

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

[10] C. Kambhamettu and D. B. Goldgof. Curvature-based approach to point correspondence recovery in conformal nonrigid motion. *Computer Vision, Graphics and Image Processing: Image Understanding*, 60(1):26–43, 1994.

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

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

[13] 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.

[14] Walter G. Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer. The eccentricity transform (of a digital shape). In *13th DGCI*, volume 4245 of *LNCS*. Springer, 2006.

[15] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE TPAMI*, 29(2):286–299, 2007.

[16] 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.

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

[18] G. Reeb. Sur les points singuliers d'une forme de pfaff complémentque intégrable ou d'une fonction numérique. *Annales de l'institut Fourier, 14 no. 1*, 14(1):37–42, 1964.

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

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

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

[22] Piere Soille. *Morphological Image Analysis*. Springer, 2nd edition, 2002.

[23] Eric W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman & Hall/CRC, 2nd edition, December 2002.

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