Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/183-2
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: ion@prip.tuwien.ac.at
yll@prip.tuwien.ac.at
krw@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-126 August 16, 2011

Skewed Coordinate System for Dense Point Correspondences Inside Articulated Shapes

Adrian Ion, Yll Haxhimusa and Walter G. Kropatsch

Abstract

This report considers using a non-rigid coordinate system to find corresponding points in different poses of the same articulated 2D shape. The shape-centered coordinate system is mapped on top of the eccentricity transform of the shape, which uses maximal geodesic distances and is bounded under articulation. The isolines of the eccentricity transform are used as one of the coordinates, the radial-like, and the other one, the angular-like, is stretched to compensate for changes in the widths of parts. The polar-like coordinate system is first computed on inter-pixel isolines and then mapped to the pixels. The angular-like coordinates are aligned using the 1D signals of the eccentricity values along the boundaries of the two shapes. Correspondences between points are established by minimizing the difference of their coordinates. Detecting failed correspondences is done using an adaptive threshold which adjusts to the changing local variation of the coordinates. Experimental results are shown on a set of hand poses, ranging from minor movement to touching or missing fingers.

Contents

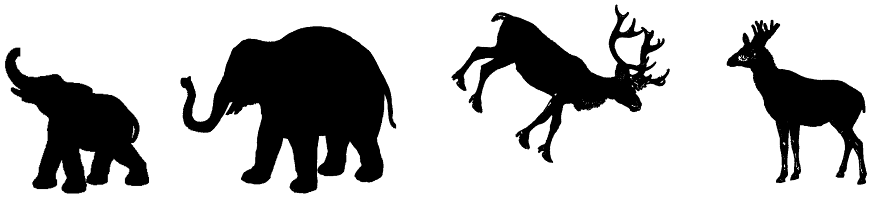


Figure 1: Shapes of elephants and deers (taken from MPEG7 CE Shape-1 Part B).

1 Introduction

Shape as well as other perceptual properties such as depth, motion, speed, and color, are important for object detection and recognition. What makes shape a unique perceptual property is its sufficient complexity to make object recognition possible, often without any additional cues [24]. For example, the mammals in Figure 1 can be easily recognized by humans based only on their shape. Motivated by the discriminating power of shape, the problem of shape matching has received a lot of attention [20, 3, 29, 10, 22]. Given two shapes, the task is to compute a similarity measure that has a higher value if the shapes belong to objects of the same class, and a lower one otherwise. Shapes can appear at different scales and orientations, different articulated poses, etc. Based on labeled examples, shape matching can be used to predict the class of a query shape or find similar looking ones.

Assume we are given two shapes of the same class, or two shapes corresponding to different poses of the same transformed object (scaled, rotated, articulated). If we are interested in finding correspondences between their points a shape similarity value does not help. While some matching methods produce correspondences of the used signature, usually border points/parts (e.g. [20, 3, 29]), finding all point-correspondences based on the obtained information is in most cases not straightforward or even impossible. In [9] a triangulation of the shape is used as a model, which could be used to find corresponding points, but an a priori known model is still needed.

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”, we address the problem of mapping 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. We study the case where only the two shapes are known, and except one optional boundary pixel, the coordinate systems are mapped independently to each shape[1].

Our 2D polar-like coordinate system is mapped on top of the Euclidean eccentricity transform [16]. Each pixel of a shape is uniquely identified by a pair of coordinates, a "radius" and an "angle" like coordinate. The coordinate system is first computed on inter-pixel isolines and then mapped to the pixels, which avoids a discretization related problem in a previous approach [15]. The zero of the angular-like coordinates are aligned in advance using the 1D signals of the eccentricity values along the boundaries of the two shapes. A locally determined threshold which adjusts to the changing local variation of the coordinates is used to detect failed correspondences.

The task of finding correspondences of all points of the shape is similar to the non-rigid registration problem used in the medical image processing community [8]. In this work, gray level information is used to compute the deformation and to register (in most cases) a whole image. In our approach we consider the registration of all pixels (the interior) of a binary, simply connected 2D shape. In the surface parametrization community a coordinate system for shapes is defined, but articulation is not considered [6]. In [18], for small variations, correspondences between points of 3D articulated shapes are found.

A preliminary version of this work was presented in [14], which was extended with the coordinate system for continuous shapes, the method to align the zeros of the angular-like coordinate, the detection of failed correspondences, and additional experimental evaluation.

The organization of the report is as follows. Section 2 recalls the eccentricity transform and its properties relevant for this report. Section 3 briefly recalls the concept of coordinate system and the problem of finding correspondences between points. In Section 4 we give the definition of the proposed coordinate system in the case of continuous shapes, and discuss discretization options, alignment, and failed correspondences in Section 5. We show experiments in Section 6, followed by discussion in Section 7, and conclude the report in Section 8.

2 The Eccentricity Transform

The definitions and properties follow [16, 19]. Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^2$ (a 2D shape). A path π in $\mathcal{S}$ is the continuous mapping from the

[1] As opposed to also having a number of point-correspondences uniformly covering the two shapes, where for example interpolation could be used to estimate the correspondence for any point.

interval $[0, 1]$ to $\mathcal{S}$. Let $\Pi^{\mathcal{S}}(\mathbf{p}_1, \mathbf{p}_2)$ 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* $d^{\mathcal{S}}(\mathbf{p}_1, \mathbf{p}_2)$ between $\mathbf{p}_1, \mathbf{p}_2$ is the length $\lambda(\pi)$ of the shortest path $\pi \in \Pi^{\mathcal{S}}(\mathbf{p}_1, \mathbf{p}_2)$:

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

$\pi(t)$ 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 $\pi \in \Pi^{\mathcal{S}}(\mathbf{p}_1, \mathbf{p}_2)$ with $\lambda(\pi) = d^{\mathcal{S}}(\mathbf{p}_1, \mathbf{p}_2)$ is called a *geodesic.*

Definition 2.1. (eccentricity) *The eccentricity of a point* $\mathbf{p} \in \mathcal{S}$ *is defined as:*

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

Definition 2.2. (eccentricity transform) *The eccentricity transform*[2] $ECC(\mathcal{S})$ *associates to each point of* $\mathcal{S}$ *the value* $ECC(\mathcal{S}, \mathbf{p})$ *i.e. to each point* $\mathbf{p}$ *it assigns the length of the geodesic path(s) to the point(s) farthest away.*

An *eccentric point* of a point $\mathbf{p} \in \mathcal{S}$ is a point $\mathbf{e} \in \mathcal{S}$ s.t. $d^{\mathcal{S}}(\mathbf{p}, \mathbf{e}) = ECC(\mathcal{S}, \mathbf{p})$. An *eccentric point* of a shape $\mathcal{S}$ is a point $\mathbf{e} \in \mathcal{S}$ that is eccentric for at least one point $\mathbf{p} \in \mathcal{S}$ i.e. $\exists \mathbf{p} \in \mathcal{S}$ such that $ECC(\mathcal{S}, \mathbf{p}) = d^{\mathcal{S}}(\mathbf{p}, \mathbf{e})$.

The *geodesic center* $\mathcal{C} \subseteq \mathcal{S}$ is the set of points with the smallest eccentricity i.e. $\mathbf{c} \in \mathcal{C}$ iff $ECC(\mathcal{S}, \mathbf{c}) = min\{ECC(\mathcal{S}, \mathbf{p}) \mid \forall \mathbf{p} \in \mathcal{S}\}$.

2.1 Properties of the eccentricity transform

The eccentricity transform is part of a greater class of shape transforms, associating to each point/pixel a function of the distances to other points of the shape. Examples include the distance transform [26, 4], the Poisson equation [12], and the global geodesic function [2].

For a shape $\mathcal{S}$ simply connected and planar, the geodesic center $\mathcal{C}$ is a single point which is the global minimum of $ECC(\mathcal{S})$ ([13], Property 23). Otherwise it can be a disconnected set of arbitrary size (e.g. for $\mathcal{S}$ being the points on a circle, all points are eccentric and they all make up the geodesic center).

Definition 2.3. (level set) *The level set of a function* $f : \mathbb{R}^n \to \mathbb{R}$, *corresponding to a constant value* h, *is the set of points* $\mathbf{p} \in \mathbb{R}^n$ *such that* $f(\mathbf{p}) = h$.

[2] Also known as the *propagation function* [30].

For $\mathcal{S} \in \mathbb{R}^2$, the level sets of $ECC(\mathcal{S})$ can be a closed curve or a set of disconnected open curves.

Definition 2.4. (isolines) *The connected components of the level sets are called isolines.*

Figure 3 shows an example of isolines.

A *reference point* of a point $\mathbf{p} \in \mathcal{S}$ is a point $\mathbf{m}$ on the boundary of $\mathcal{S}$, $\mathbf{m} \neq \mathbf{p}$ s.t. $\mathbf{m}$ is on the geodesic from $\mathbf{p}$ to its eccentric point $\mathbf{e}$, and $d(\mathbf{p}, \mathbf{m})$ is minimal ($\mathbf{m}$ is closest to $\mathbf{p}$ in terms of the geodesic distance d). Reference points have the property that $\pi(\mathbf{p}, \mathbf{m})$ is a line segment and all points at the same distance to $\mathbf{e}$ that have $\mathbf{m}$ as a reference point, will lie on a circle centered at $\mathbf{m}$. As a result, the isolines of $ECC(\mathcal{S})$ are made out of circular arcs, corresponding to different *reference points.*

For $\mathcal{S}$ simply connected and planar, the shapes $\mathcal{P}_k = \{\mathbf{p} \in \mathcal{S} | ECC(\mathcal{S}, \mathbf{p}) \leqslant k\}$ for $k \in [\min(ECC(\mathcal{S})), \max(ECC(\mathcal{S}))]$ are geodesically convex in $\mathcal{S}$ [13] [3]. As a result, the isolines corresponding to k have 'the smaller angle' on the inside of $\mathcal{P}_k$ (the unit normal vector points towards the inside of $\mathcal{P}_k$).

Due to using geodesic distances, the variation of ECC is bounded under articulated deformation to the width of the "joints" [20]. The positions of eccentric points and geodesic center are robust w.r.t shape distortion [19].

2.2 Discrete shapes and their eccentricity transform

In this report, the class of 4-connected, planar, and simply connected[4] discrete shapes $\mathcal{S}$ defined by points on the square grid $\mathbb{Z}^2$ are considered.

Geodesics in $\mathcal{S}$ are contained in the area $\mathcal{S}' \subset \mathbb{R}^2$ enclosed by the polygon connecting the centers of the boundary 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.*

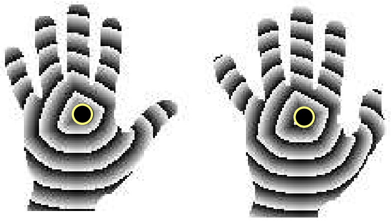
The eccentricity of a pixel of the discrete shape $\mathcal{S}$ is:

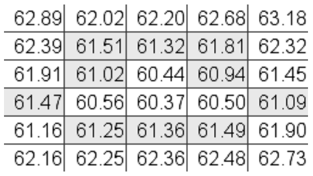
$$ECC(\mathcal{S}, \mathbf{p}) = \max\{d^{\mathcal{S}'}(\mathbf{p}, \mathbf{q}) \mid \mathbf{q} \in \mathcal{S}\}. \tag{3}$$

i.e. the only points considered are the ones of $\mathcal{S}$ but the geodesics are contained in $\mathcal{S}'$, thus $ECC(\mathcal{S})$ is a regular sampling from the corresponding continuous eccentricity transform $ECC(\mathcal{S}')$.

[3] A subset $\mathcal{P}$ of $\mathcal{S}$ is said to be geodesically convex if for every $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{P}$, the geodesic $\pi(\mathbf{p}_1, \mathbf{p}_2)$ is contained in $\mathcal{P}$ [23].

[4] The continuous shape in $\mathbb{R}^2$ obtained by the union of the squares of size one, centered at the points of $\mathbb{Z}^2$ belonging to the shape is simply connected.





62.89	62.02	62.20	62.68	63.18
62.39	61.51	61.32	61.81	62.32
61.91	61.02	60.44	60.94	61.45
61.47	60.56	60.37	60.50	61.09
61.16	61.25	61.36	61.49	61.90
62.16	62.25	62.36	62.48	62.73

eccentricity values (extract)

Figure 2: Left: Eccentricity transform of two shapes from the hand class (gray values are the eccentricity value modulo a constant). Geodesic center marked with '●'. Right: for the first hand (straight thumb), extract of eccentricity values around the geodesic center – discrete isoline with ECC values in $[60.87, 61.87)$ emphasized, the geodesic center has ECC 60.37.

Definition 2.5. (discrete level set and isoline) *The discrete level sets of the $ECC(\mathcal{S})$ are the sets given by:*

$$h - o \leqslant ECC(\mathcal{S}) < h + 1 - o$$

for $h = \lfloor \min(ECC(\mathcal{S})) \rfloor, \lfloor \min(ECC(\mathcal{S})) \rfloor + 1, \ldots,$ and a fixed offset $o \in [0, 1)$ which is used to control the thickness of the discrete level set containing the geodesic center. The 8-connected components of the discrete level sets are called discrete isolines and correspond to isolines in the continuous domain $\mathcal{S}'$.

As the continuous isolines are made of circular arcs defined by the Euclidean distance to the corresponding reference points, selecting pixels in a distance interval of the form $[h, h+1)$ produces 8 connected discrete curves [1].

Figure 2 shows two hand shapes (taken from the Kimia99 database, [27]) their eccentricity transform and one table with the eccentricity values around the center. Note the articulation of the thumb.

2.3 Computation

The geodesic distance function [30] $D^{\mathcal{S}}(\mathbf{p})$ with a starting point $\mathbf{p}$, 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 $ECC(\mathcal{S})$. For 2D shapes $D^{\mathcal{S}}(\mathbf{p})$ can be efficiently computed using discrete circles [16] or the fast marching algorithm [28]. Efficiently estimating $ECC(\mathcal{S})$ using evaluations of $D^{\mathcal{S}}(\mathbf{p})$ is discussed in [16].

3 Coordinate System: definition, point correspondences

A system of *curvilinear coordinates* is composed of intersecting surfaces. If all intersections are at angle $\pi/2$, then the coordinate system is called *orthogonal* (e.g. the polar and Cartesian coordinate systems). If not, a *skew* coordinate system is formed. Two classes of curves are needed for a planar system of curvilinear coordinates, one for each coordinate.

The problem of mapping a coordinate system to a shape $\mathcal{S}$ is equivalent with defining the following mapping:

Definition 3.1. (Coordinate map) *A 2D coordinate map, is an injective map* $coord(\mathbf{p}) : \mathcal{S} \to \mathbb{R} \times \mathbb{R}$ *that associates to each point a pair of coordinates i.e.*

1. *(defined)* $\forall \mathbf{p} \in \mathcal{S}$, $coord(\mathbf{p})$ *is defined;*
2. *(distinct)* $\forall \mathbf{p}, \mathbf{q} \in \mathcal{S}$, $\mathbf{p} \neq \mathbf{q}$, *we have* $coord(\mathbf{p}) \neq coord(\mathbf{q})$*;*
3. *(single)* $\forall \mathbf{p} \in \mathcal{S}$, $coord(\mathbf{p})$ *returns a single pair of coordinates.*

Consider two **continuous shapes** $\mathcal{S}_1$ and $\mathcal{S}_2$, and the continuous coordinate maps $coord_1$ and $coord_2$ corresponding to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. $\mathcal{S}_2$ can be a translated, rotated, scaled or/and by articulation deformed verison of $\mathcal{S}_1$. Given a point $\mathbf{p} \in \mathcal{S}_1$ its corresponding point in $\mathcal{S}_2$ is $\mathbf{q} \in \mathcal{S}_2$ s.t. $coord_2(\mathbf{q}) = coord_1(\mathbf{p})$.

For two **discrete shapes** $\mathcal{S}_1$ and $\mathcal{S}_2$, and the discrete maps $dcoord_1$ and $dcoord_2$, a point $\mathbf{q} \in \mathcal{S}_2$ with exactly the same coordinates as $\mathbf{p} \in \mathcal{S}_1$ might not exist. The corresponding point $\mathbf{q}$ is defined to be the one that has its coordinates 'closest' to $dcoord_1(\mathbf{p})$. As the local variation of the two coordinates might be very different and depends on both coordinates of $\mathbf{q}$, just using the ℓ_2-norm $||dcoord_1(\mathbf{p}) - dcoord_2(\mathbf{q})||$ to find the 'closest' coordinates is not the best solution. For example, for polar coordinates the variation of the angular coordinate of two neighboring pixels having the same r can be as large as the variation of the 'radial' coordinate between e.g. 10 pixels[5].

We employ a two step scheme (details follow in Section 5.3): given a point $\mathbf{p} \in \mathcal{S}_1$ with $dcoord_1(\mathbf{p}) = (r_p, \theta_p)$:

Step 1 use the **first coordinate** (r_p) to find the points $\mathbf{q} \in \mathcal{Q} \subset \mathcal{S}_2$, $dcoord_1(\mathbf{q}) = (r_q, \theta_q)$ that minimize $|r_p - r_q|$; and

[5] The modified ℓ_2-norm $||(\Delta r, r \cdot \Delta\theta)||$ which is valid for polar coordinates [17], cannot be used in our case, as the variation of one coordinate depends on both coordinate values.

Step 2 use the **second coordinate** (θ_p) to choose from the preselected points $\mathcal{Q}$ the point that minimizes the difference in the second coordinate $|\theta_q - \theta_p|, \mathbf{q} \in \mathcal{Q}$.

4 Eccentric Polar Shape Coordinates (EPS) for Continuous Shapes

In the following we define the coordinate system for continuous domains $\mathcal{S}$ to support the understanding of the discrete cases targeted by our work. The rest of the article considers $\mathcal{S}$ to be a discrete shape.

We are interested in a coordinate system that is *rotation* and *scale invariant*, *robust* [6] with respect to *articulation* and changes in the *geometry of parts* (e.g. thickening of a finger), and have certain local invariance to *addition/removal of parts* (e.g. missing the tip of a finger).

The proposed coordinate system is intuitively similar to the polar coordinate system, but forms a skew coordinate system. It is defined for *simply connected, planar 2D shapes*, and has two coordinates denoted by r and θ. The origin of the coordinate system is taken to be the geodesic center of the shape $\mathcal{S}$ (Figure 2). In the following we introduce the required notions, followed by the definitions of the two coordinates, $\mathbf{r}$ and θ.

4.1 Prerequisites

Let S be a simply-connected planar shape without an isthmus, a narrow part of the shape, the removal of which disconnects the shape.

Definition 4.1. (descendant/ascendant) *An isoline $\mathcal{L}_2$ of $ECC(\mathcal{S})$ with ECC value e_2 (i.e. $\forall \mathbf{p} \in \mathcal{L}_2, ECC(\mathbf{p}) = e_2$) is said to be a descendant of an isoline $\mathcal{L}_1$ of $ECC(\mathcal{S})$ with ECC value $e_1 < e_2$* **iff** *there exist $\mathbf{p}_1 \in \mathcal{L}_1, \mathbf{p}_2 \in \mathcal{L}_2$ and a path $\pi \in \Pi^{\mathcal{S}}(\mathbf{p}_1, \mathbf{p}_2)$ such that $ECC(\mathcal{S}, \pi(t_1)) < ECC(\mathcal{S}, \pi(t_2))$ for any $0 \leqslant t_1 < t_2 \leqslant 1$. The isoline $\mathcal{L}_1$ is said to be an ascendant of $\mathcal{L}_2$.*

Intuitively $\mathcal{L}_2$ is a descendant of $\mathcal{L}_1$ if one can travel inside $\mathcal{S}$ from $\mathcal{L}_1$ to $\mathcal{L}_2$ following a path with strictly ascending eccentricity values. E.g. for the hands in Figure 2 two isolines on different fingers are descendants of the geodesic center, but none of them is a descendant of the other.

Definition 4.2. (decomposition) *Given a shape $\mathcal{S}$ and its eccentricity transform $ECC(\mathcal{S})$, the shape $\mathcal{S}$ is decomposed into connected regions $\cup \mathcal{P}_i = \mathcal{S}$,*

[6] i.e. small changes of the shape will result in a small change in the coordinate system.

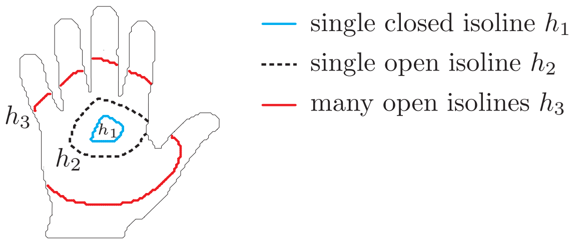


Figure 3: Level sets made of different configurations of isolines. $ECC(\mathbf{p}_1) < ECC(\mathbf{p}_2) < ECC(\mathbf{p}_3)$, $\mathbf{p}_1 \in h_1, \mathbf{p}_2 \in h_2, \mathbf{p}_3 \in h_3$.

$\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ *such that each isoline is fully included in one of the parts and any two isolines* $\mathcal{L}_1, \mathcal{L}_2$ *with* ECC *values* $e_1 \leqslant e_2$ *are in the same part* $\mathcal{P}_i$ **iff** $e_1 \neq e_2$*,* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *have the same topology (i.e. they are both closed*[7] *or both open curves),* $\mathcal{L}_2$ *is a descendant of* $\mathcal{L}_1$*,* $\mathcal{L}_1$ *has no other descendants with* ECC *value* e_2*, and all isolines* $\mathcal{L}_3$ *which are descendants of* $\mathcal{L}_1$ *and ascendants of* $\mathcal{L}_2$*, are in the same part with* $\mathcal{L}_1$ *(and thus with* $\mathcal{L}_2$*).*

Intuitively, imagine a wave starting from the geodesic center of the shape and traveling in $\mathcal{S}$ s.t. its position at any given time corresponds to a level set of $ECC(\mathcal{S})$. Whenever a connected component of the wavefront opens or breaks into several connected components, the current part region "ends" and new ones are started, one for each created component. Figure 3 shows example level sets of the ECC. The isolines of the level sets drawn belong to different regions.

The obtained decomposition can be related to the Reeb graph [25] of $ECC(\mathcal{S})$. The Reeb graph associates a point to every isoline[8] of $ECC(\mathcal{S})$. Points having exactly two neighbors are joined to form arcs. Points not belonging to arcs form nodes. The proposed decomposition associates one region $\mathcal{P}_i$ to every arc of the corresponding Reeb graph. Except for the geodesic center, the isolines corresponding to the nodes of the Reeb graph are added to the region corresponding to the incident arc with smaller eccentricity values. The geodesic center is added to its only incident arc. Figure 4 shows an example.

Definition 4.3. (center region) *Given a shape* $\mathcal{S}$ *and its decomposition into part regions* $\mathcal{P}_i$*, the region* $\mathcal{P}_i$ *that contains the geodesic center* $\mathbf{c} \in \mathcal{P}_i$ *is called the center region.*

[7] The single point forming the geodesic center of a simply connected shape is considered as a closed isoline.

[8] also called component.

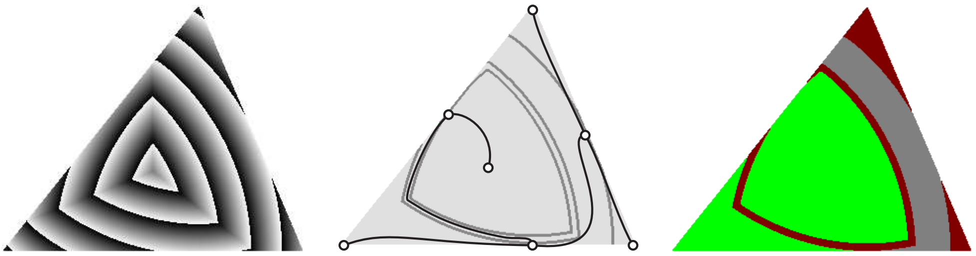


Figure 4: Reeb graph of $ECC(\mathcal{S})$ and shape decomposition based on the $ECC(\mathcal{S})$. Left: shape with $ECC(\mathcal{S})$, gray value equals eccentricity value modulo a constant. Center: Reeb graph with isolines corresponding to nodes in dark gray. Right: decomposition based on $ECC(\mathcal{S})$ (the colors are chosen to allow differentiation of adjacent regions and have no special meaning).

In the center region all isolines except the one representing the geodesic center (which is a single point) are closed curves. In all other parts they are open curves. The closed isolines in the center region are assigned counter-clockwise *orientation*. All other isolines have the same orientation as their ascendant with highest eccentricity value. For the isolines outside the center region this orientation labels the two end-points as first and last: $\mathbf{p}_f$, $\mathbf{p}_l$.

4.2 The first coordinate: r

The coordinate $r(\mathbf{p})$ is a linear mapping from $ECC(\mathcal{S}, \mathbf{p})$ to the interval $[0, 1]$:

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

As a result, the level sets of r correspond to the level sets of $ECC(\mathcal{S})$ and can be a closed curve or a set of disconnected open curves. Where $ECC(\mathcal{S})$ is smooth $dECC(\mathcal{S})/dr = 1/(\max(ECC(\mathcal{S})) - \min(ECC(\mathcal{S})))$.

4.3 The second coordinate: θ

The coordinate θ is mapped to the isolines of $\mathbf{r}$ as follows (Figure 5):

- the values of θ are in the domain $[0, 1)$ and have period 1;
- for the geodesic center $\theta(\mathbf{c}) = 0$;
- the θ values of closed isolines cover the whole domain $[0, 1)$;
- the domain $[\theta_{st}, \theta_{en}]$ of θ values of open isolines is the same for all isolines in the same part region $\mathcal{P}_i$;

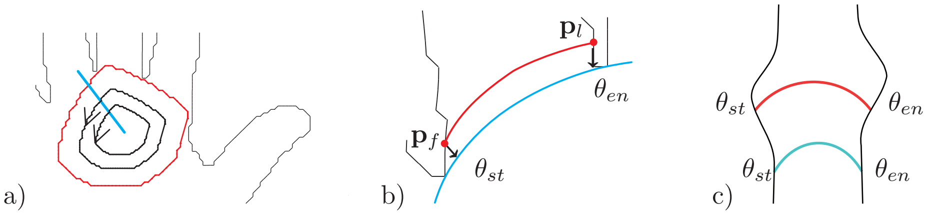


Figure 5: a) $\theta = 0$ in the center region is set using the zero-path (straight blue line), orientation of isolines shown with arrows. For isolines outside the center region: b) each region gets θ_{st}, θ_{en} from the neighboring region with smaller ECC, c) θ_{st}, θ_{en} kept constant in the same region.

- except for the center region the θ_{st}, θ_{en} values associated to each part region $\mathcal{P}_i$ are assigned using the adjacent region $\mathcal{P}_j$ with smaller eccentricity. The θ values of the points $\mathbf{p}'_f, \mathbf{p}'_l \in \mathcal{P}_j$ closest to the end points $\mathbf{p}_f, \mathbf{p}_l$ of the isoline in $\mathcal{P}_i$ with the smallest eccentricity are taken;

- for each closed isoline $\mathcal{L}$ in the center region

$$\theta(\mathbf{p}) = \lambda(\mathcal{L}(\mathbf{p}_0, \mathbf{p}))/\lambda(\mathcal{L}) \tag{5}$$

  where $\mathbf{p}_0 \in \mathcal{L}$ is the point with $\theta = 0$, $\mathcal{L}(\mathbf{p}, \mathbf{p}_0)$ is the part of $\mathcal{L}$ traversed when going from $\mathbf{p}_0$ to $\mathbf{p}$ in counter-clockwise orientation, and $\lambda(\mathcal{L})$ is the length of $\mathcal{L}$. The points with $\theta = 0$ are at the intersection between the closed isoline and a path, called the *zero-path*. Further discussion on choosing the zero-path follows in Section 5.2;

- for each isoline $\mathcal{L}$ in other parts than the center region

$$\theta(\mathbf{p}) = mod\left(\frac{\lambda(\mathcal{L}(\mathbf{p}_f, \mathbf{p}))}{\lambda(\mathcal{L})} \cdot (\theta'_{en} - \theta_{st}), 1\right) \tag{6}$$

  where $mod(x, 1)$ is the value of x modulo 1, $\mathcal{L}(\mathbf{p}_f, \mathbf{p})$ is the part of $\mathcal{L}$ from $\mathbf{p}_f$ to $\mathbf{p}$, $\theta'_{en} = \theta_{en}$ if $\theta_{en} > \theta_{st}$ or $\theta'_{en} = 1 + \theta_{en}$ otherwise, and θ_{st}, θ_{en} are the start and end θ values associated to the part. Since we exclude an isthmus of the shape $\theta_{en} \neq \theta_{st}$

- dividing by the length of the isolines assumes that the shape has a width larger than one point. In the case of the outer-most isolines that have no ascendants, if they are made of a single point, θ_{st} is assigned.

Proposition 4.1. *Let $r(p)$ be defined by Equation 4 and $\theta(p)$ be defined by Equations 5 and 6, then (r, θ) is a 2D coordinate map.*

Proof. 1. $r(\mathbf{p})$ is defined by ECC (Equation 2). $\theta(\mathbf{p})$ is defined if $\mathbf{p}_0$ is defined and the shape does not contain isthmuses.

2. All points with different eccentricity are distinguished by $r(\mathbf{p})$. Equations 5 and 6 assign different values in $[0, 1)$ to all points with the same $\mathbf{r}(p)$. Hence, (r, θ) distinguishes all points of $\mathcal{S}$.

3. Every point $\mathbf{p}$ in $\mathcal{S}$ has unique eccentricity value $r(\mathbf{p})$. For the center region $\theta(\mathbf{p})$ measures the normalized arc length from $\mathbf{p}_0$ along the closed isoline, thus it is unique. In other parts (non center region) θ_{st} and θ_{en} are projected to isolines with smaller eccentricity until reaching a closed isoline, with monotonic values of θ. Let P_i and P_{i+1} be two neighboring segments. Thus θ_{en} of $\mathcal{P}_i$ is smaller than θ_{st} of the neighbor part $\mathcal{P}_{i+1}$. □

From the definition it follows that where $ECC(\mathcal{S})$ is smooth $dECC(\mathcal{S})/d\theta = 0$.

The eccentricity transform is invariant to *rotation* of the shape. Mapping the coordinate values to the fixed domains $[0, 1]$ respectively $[0, 1)$ gives *invariance to uniform scaling.* As $ECC(\mathcal{S})$ uses geodesic distances, the values $ECC(\mathcal{S}, \mathbf{p})$ and implicitly of $\mathbf{r}(\mathbf{p})$ are robust w.r.t *articulation.* Keeping θ_{st}, θ_{en} constant in each part region and setting them depending on the "ascendant part" (with lower ECC values) results in the robustness of θ to changes in the *geometry of other parts* with the same or higher eccentricity, and certain local invariance to *addition/removal of parts* (e.g. thinning the tip of a finger will not affect the coordinates in the other fingers). The last aspect relies on the property of ECC that its value in a point remains unchanged if the corresponding eccentric point and the geodesic distance to it remain the same.

5 The Proposed Coordinate System for Discrete Shapes

In the following we discuss mapping the coordinate system described in the previous section to discrete shapes $\mathcal{S}$.

The Pixel Mapped Coordinate System A first option to map the coordinate system to a discrete shape $\mathcal{S}$ is to extract discrete level sets as in Definition 2.5 using an offset $o = 0$, and directly compute the coordinate system on them [15]. The shape is decomposed into regions based on the topology of the discrete level sets.

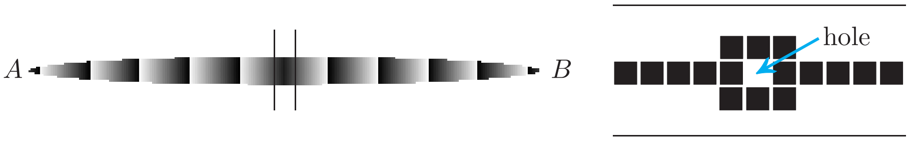


Figure 6: Example case with discretization problem. Left: elongated shape, gray values are the eccentricity values modulo a constant. The shape has two eccentric points marked A, B. The eccentricity of the points in the left and in the right half of the shape, equals the distance to B, A, respectively. The isolines are made of arcs centered at the corresponding eccentric point. Right: zoom-in on a 'discrete level set' corresponding to a closed isoline (rotated).

For mapping the θ values, the length of a discrete isoline part is computed as the length of the polygonal line connecting the centers of its pixels.

An 8-connected path that connects the geodesic center with a pixel on the border of the shape is used to choose the pixel with $\theta = 0$ for each isoline of the *center region*. This path is obtained by following a steepest descent on the geodesic distance from the boundary pixel to the geodesic center. We choose only the boundary pixel with the maximum eccentricity.

Problem due to discretization Depending on the position and distance to the reference points, the circular arcs making up the isolines of $ECC(\mathcal{S})$ can meet under very sharp angles. In the case of the center region, isolines are closed in the continuous domain. The pixels of the corresponding discrete level sets are 8-connected, but an 8-connected closed path, where each pixel would be passed through only once might not exist (Figure 6 right). In this case an ordering cannot be made between the pixels and computing the length of the discrete isoline required to map the θ coordinate is not possible.

Figure 6 shows an example. The points A and B are the eccentric points of the opposite side of the shape, respectively. As the shape is convex all geodesics are straight line segments, and the isolines of the eccentricity transform are circular arcs centered at one of the two eccentric points.

5.1 The Inter-Pixel Coordinate System

To avoid the discretization problem, instead of directly mapping the coordinate system to the discrete level sets made out of pixels, as mentioned in Section 5, we extract *"inter-pixel" isolines* and map the coordinate system on them, followed by mapping the computed coordinates to the pixels.

5.1.1 Inter-pixel isolines

To extract inter-pixel isolines we use *marching squares*, the 2D version of the classical *marching cubes* [21]. The produced isolines are polygonal lines defined by points on the line segments that connect the pixel centers. The eccentricity of points on the line segments connecting pixel centers is estimated by linear interpolation. All vertices of an inter-pixel isoline have the same eccentricity value. The inter-pixel isolines are constructed such that they do not self-intersect and the number of connected components is minimal (e.g. in the case of a 2x2 checker board, one single isoline surrounding both pixel centers and not self-intersecting). The extracted inter-pixel isolines are less constrained than the boundaries defined on the half-integer plane and more constrained than the inter pixel Euclidean paths, both of them preserving the topology [5].

We extract inter-pixel isolines of the ECC of $\mathcal{S}$ for the eccentricity values:

$$\min(ECC) + 0.5, \min(ECC) + 1.5, ...$$

Inside the center region the extracted inter-pixel isolines will be non self-intersecting, convex polygonal lines. Outside the center region, where level sets may become disconnected, each inter-pixel isoline will be a polygonal line with all angles on the side with smaller ECC, less or equal to 180 degrees Figure 7 shows examples.

The inter-pixel isolines are connected paths where the ordering of the points is unambiguous and their length is the sum of the lengths of the straight polygonal segment. After extracting inter-pixel isolines, we can map the proposed coordinate system as follows:

1. map the EPS coordinate system on the inter-pixel isolines;
2. map the r, θ values from the isolines to the pixels (discussed below).

5.1.2 Mapping r and θ values

The EPS coordinate system has been mapped to the inter-pixel isolines. All vertices of the isolines correspond to points with the same r, and each vertex has a unique θ value. The next step is to map r and θ values to the pixels of the shape based on the r and θ values on the inter-pixel isolines.

We can identify all pixels located between inter-pixel isolines corresponding to two consecutive r values with one of the following relations (Figure 7a):

$$e < ECC(\mathcal{S}) \leqslant e + 1 \tag{7}$$

$$e \leqslant ECC(\mathcal{S}) < e + 1 \tag{8}$$

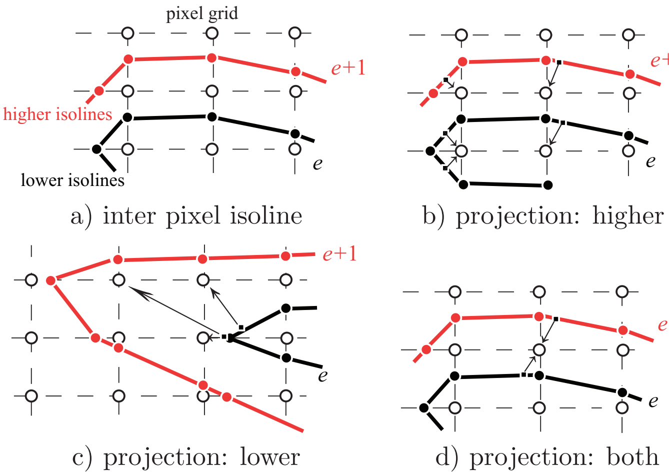


Figure 7: Inter-pixel isolines. Mapping of r, θ from the isoline to the pixels.

where $e, e+1$ are the eccentricity values corresponding to two consecutive r values r_1, r_2. The values of r are obtained from the values e using linear interpolation as mentioned in Section 4. The choice between Equations 7 and 8 depends on the method chosen to compute the θ value for each pixel and is discussed below. All pixels satisfying the chosen equation are considered to have the same r coordinate.

Inter-pixel isolines are extracted for eccentricity values with difference one. Thus for each pixel the number of inter-pixel isolines crossing any of the incident line segments that connect pixel centers, is maximum four and at least one.

To compute the θ value for a pixel we consider the inter-pixel isolines that bound the pixel center and identify the following options (Figure 7 illustrates the used notation):

higher: assign θ of the closest point(s) of the isoline with higher ECC value;

lower: assign θ of the closest point of the isoline with lower ECC value;

both: assign a function of θ of the closest points of both isolines.

Linear interpolation is used to compute the value of θ corresponding to the closest point(s) on the inter-pixel isolines.

Recall that the inter-pixel isolines are polygonal lines, and that the angles at their vertices are less or equal to 180 degrees when measured on the side with the smaller ECC values. We can state the following about each option:

higher: one pixel can get more than one θ value (Figure 7.b);

lower: more than one pixel can get the same θ value (Figure 7.c);

both: could produce smoother results, but uniqueness of θ values is not guaranteed (Figure 7.d).

In addition, option 'higher' cannot compute θ values for the border pixels that have no higher eccentricity isolines, and option 'lower' cannot compute θ values for the geodesic center.

Going back to the definition of the coordinate system, the proposed method to find correspondences, and the 3 requirements for the coordinate map (Section 3) we can state that properties 'defined' and 'distinct' from Definition 3.1 are required to be able to find a corresponding point. We can relax the requirements and give up property 'single', and if a pixel would get more than one θ value, consider only the smallest one or the average.

Following this reasoning, **option 'higher' is chosen**, as it assigns distinct θ values to all pixels having the same r.

5.2 The Zero Path

Property 5.1. *Any geodesic* $\pi(\mathbf{q}, \mathbf{c})$ *where* $\mathbf{c}$ *is the center point and* $\mathbf{q} \in \partial\mathcal{S}$ *is a boundary point of the shape* $\mathcal{S}$*, can be used as a zero path.*

Proof. The zero path is used only in the center region (Section 4). The requirement is that it intersects all isolines of the center region, to allow setting the point with $\theta = 0$ on each isoline. By definition all isolines of the center region are closed. Any path $\pi(\mathbf{p}, \mathbf{q})$ for $\mathbf{p}$ a point located in the space enclosed by the closed isoline and $\mathbf{q}$ a point outside it, will intersect the isoline. The center point $\mathbf{c}$ is located on the isoline corresponding to the smallest ECC values and is in the space enclosed by all other isolines in the center region. Any boundary point $\mathbf{q}$ is either part of the closed isoline corresponding to the largest ECC values in the center region, or outside the center region. □

For obtaining a zero path, we identify the following cases:

independent computation: the zero path is chosen by knowing only the current shape.

best alignment: both shapes for which corresponding points will be computed are known.

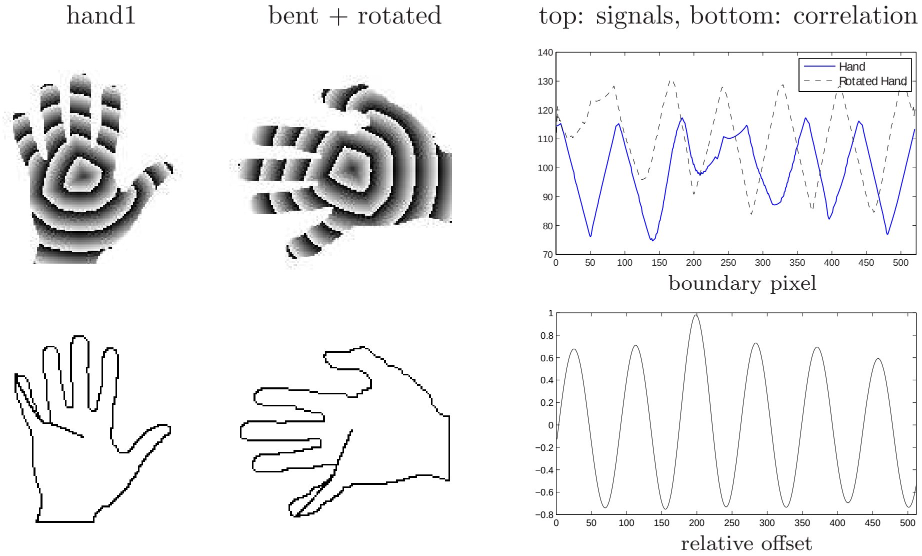


Figure 8: Alignment of zero paths using correlation between boundary eccentricity signals. Top row: shapes and their eccentricity transform, and boundary eccentricity signals (vertical axis: eccentricity values). Bottom row: zero paths for $\mathbf{q}_1$ a pixel with maximum eccentricity, and correlation between the two signals for all possible offsets (vertical axis: correlation value).

Independent computation In the case of independent computation one can consider obtaining the boundary point by using one of the existing shape orientation methods (e.g. [31]): first orient the shape, then choose a point with a fixed relative position e.g. top-left most. By definition shape orientation methods consider rigid transforms and will thus probably be less optimal for larger non-rigid deformations like part articulation.

Best Alignment As the coordinate system is used for finding corresponding points between two articulated shapes (of the same object/class), we can also assume knowing both shapes at the time the coordinate system is mapped.

Some shape matching methods (e.g. [3, 20]) use features computed at boundary points to match two shapes. The computation of the matching "score" includes correspondences between the considered points (so called "aligning of the feature points"). If the classification/matching of the shapes was done using such a shape matching method, then the computed boundary point correspondences can be used to compute the zero path (e.g. choosing one of the corresponding boundary points with the highest eccentricity).

Aligning based on *ECC* For the cases where no additional information is given, just the two shapes, we present an approach that relies on the (already computed) eccentricity transform to align boundary points of the two given shapes and obtain the corresponding boundary point required for the computation of the zero path.

A *boundary pixel* of the shape $\mathcal{S}$ is a pixel $\mathbf{p} \in \mathcal{S}$ s.t. $\exists \mathbf{q} \in \bar{\mathcal{S}}$ a 4-connected neighbor of $\mathbf{p}$, with $\bar{\mathcal{S}}$ the set of pixels not contained in $\mathcal{S}$ (the background). The *(discrete) boundary* $\mathcal{B}(\mathcal{S})$ of a discrete, simply connected shape $\mathcal{S}$, is the ordered 8-connected set formed by the boundary pixels of $\mathcal{S}$ by traversing the boundary of $\mathcal{S}$ in clockwise order. Every 4-connected region is surrounded by a closed 8-connected Freeman chain code [11] which imposes an order on the visited boundary pixels. In case of a part of width one the same pixel may be visited twice and contributes also twice to the boundary signal. Figure 8 shows two hand shapes and their eccentricity transform (second hand is rotated and with the thumb bent). The eccentricity value corresponding to a boundary pixel can be used as a feature to create a one dimensional signal. This signal will be called the *boundary ECC signal* and is obtained by taking $ECC(\mathcal{S}, \mathbf{q})$ for all $\mathbf{q} \in \mathcal{B}(\mathcal{S})$ in clock-wise order. Figure 8 shows the boundary ECC signals for the two shapes.

To find correspondences between boundary points of two shapes $\mathcal{S}_1, \mathcal{S}_2$ we align their boundary ECC signals. The signal with a smaller number of values is linearly scaled to the length of the other one. The correlation of the two signals is computed for all possible offset values and the offset that gives the largest correlation is taken. Figure 8 shows the correlation value between the two signals for all possible offsets (bottom right), and example zero paths resulting from this approach (bottom left). The zero paths for $\mathcal{S}_1, \mathcal{S}_2$ are the geodesics $\pi(\mathbf{q}_1, \mathbf{c}_1)$ and $\pi(\mathbf{q}_2, \mathbf{c}_2)$ where $\mathbf{q}_1 \in \mathcal{S}_1$ is a boundary pixel of $\mathcal{S}_1$, $\mathbf{c}_1$ the center of $\mathcal{S}_1$, $\mathbf{q}_2 \in \mathcal{S}_2$ is the boundary pixel of $\mathcal{S}_2$ corresponding to $\mathbf{q}_1$ as given by the offset with the maximum correlation, and $\mathbf{c}_2$ is the center of $\mathcal{S}_2$.

The initial choice for $\mathbf{q}_1$ was a pixel having the maximum eccentricity as this is a stable location. But this location might be far away from the center region and depending on the deformations of $\mathcal{S}_1, \mathcal{S}_2$ it can lead to significantly different zero paths in the center region. The boundary pixels with the minimum eccentricity lie on the outer most isoline of the center region, providing more control over the location of the zero-path inside the center region.

5.3 Reconstruction of image coordinates

Recall the situation of Section 3. To find the pixel $\mathbf{q} \in \mathcal{S}$ given the coordinates $(r_{\mathbf{p}}, \theta_{\mathbf{p}})$ we proceed as discussed (see Figure 9):

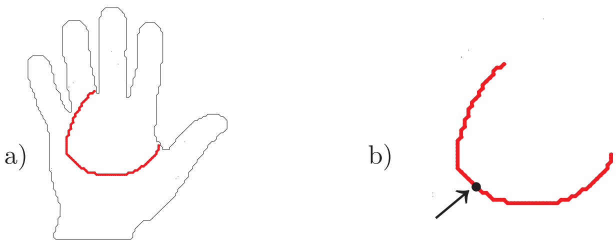


Figure 9: Find pixel given coordinates $(r_{\mathbf{p}}, \theta_{\mathbf{p}})$: a) select the discrete level set corresponding to $r_{\mathbf{p}}$; b) find the pixel of the level set with the closest θ.

1. $\mathcal{L}$ = discrete level set containing $r_{\mathbf{p}}$;
2. find $\arg\min_{\mathbf{q}\in\mathcal{L}}\{|\theta_{\mathbf{q}} - \theta_{\mathbf{p}}|\}$, where $\theta_{\mathbf{q}}$ is the θ value associated to $\mathbf{q}$,

where the discrete level sets have been extracted with $o = 0$ in Definition 2.5 for the approach in Section 5 and with $o = -0.5$ for the proposed inter-pixel mapping (corresponds to selecting the pixels between two consecutive inter-pixel isolines).

The coordinate r is chosen as the 'first coordinate' (see Section 3) because it has a constant variation and the obtained discrete level set does not contain more than one pixel with the same θ. Due to its non-constant variation, choosing θ as the 'first coordinate' could produce many pixels with the same r (thick discrete level sets) or have r values for which no pixel was selected (disconnected level sets).

5.4 Finding pixel correspondences between two shapes

Consider two hand shapes where one shape has a missing finger to simulate a segmentation error (see Section 6 for an example). Given any *'query'* coordinates $(\mathbf{r}_q, \theta_q)$ the presented two-step method will always return a pixel, even if its coordinates $(\mathbf{r}_r, \theta_r)$ differ considerably from the query coordinates $(\mathbf{r}_q, \theta_q)$. This results in corresponding points being found even for the pixels corresponding to the 'missing' finger.

To check whether the returned pixel is a good 'match' we look at the difference between the query coordinates and the ones of the returned pixel. A small difference indicates a successful correspondence and a larger one a failed one. Keeping in mind the change in local variation of the θ coordinate (Section 4), thresholding the difference between $(\mathbf{r}_q, \theta_q)$ and $(\mathbf{r}_r, \theta_r)$ with a constant will not work.

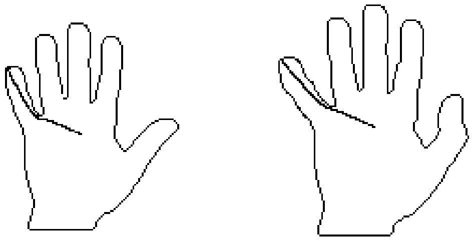


Figure 10: Zero paths for the two hand shapes from Kimia 99 [27].

The local variation of θ along an isoline can be determined when computing the values of θ for each point and is equal to $\theta_{en} - \theta_{st}$ over the length of the current isoline segment[9]. The local variation of θ, denoted by θ', can be projected from the inter-pixel isolines to the pixels, similar to the projection of θ. Having $(\mathbf{r}_r, \theta_r, \theta'_r)$ for the returned pixel, we can formulate the decision of a successful correspondence as a thresholding of $|\theta_q - \theta_r|$ with a factor t of θ'_r. Assuming that the local variation of θ in a small neighborhood around the returned pixel is constant, for small values of t, the value of t specifies a *maximum accepted deviation in pixels*. Values of t smaller than 1 tend to produce spurious failed correspondences, due to the fact that the θ sought for might not exactly match any of the pixels (we work with discrete shapes). Larger values of t will be more tolerant to wrong correspondences (e.g. a value of $t = 18$ which is half of the length of the isolines in the missing finger, will return no failed correspondences for the finger).

Note that in the case of missing parts, the change in the eccentricity transform (controlled by the presence of the original eccentric points and the length of the geodesics to them) as well as the mapping of the θ coordinate can influence the result (the adjacency graph of the part region $\mathcal{P}_i$ is a tree rooted at the vertex corresponding to the center region, having a change in a certain vertex will affect only regions corresponding the vertex and its ancestors).

6 Experiments

Discretization methods We compare the results using inter-pixel isolines with mapping directly on discrete isolines on the hand shapes used in [15]. The two shapes are from the Kimia99 database [27] and show the same hand with the thumb bent and the little finger moved. Figure 10 shows the used

[9] The periodicity of 1 of the θ value has to be correctly handled s.t. differences like $|0.1 - 0.9|$ give $1 + 0.1 - 0.9 = 0.2$.

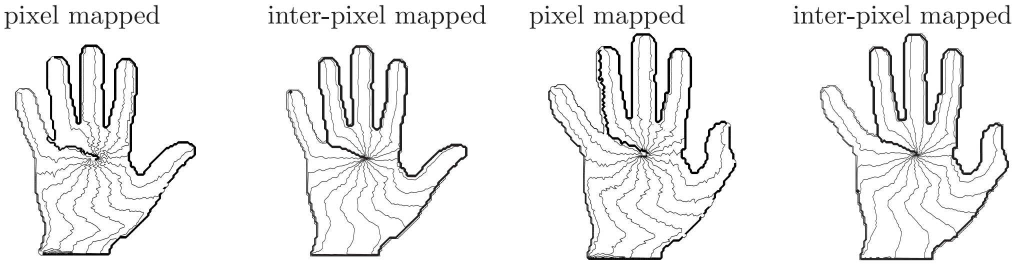


Figure 11: Isolines of θ for the two shapes, with the two methods (Sections 5, 5.1).

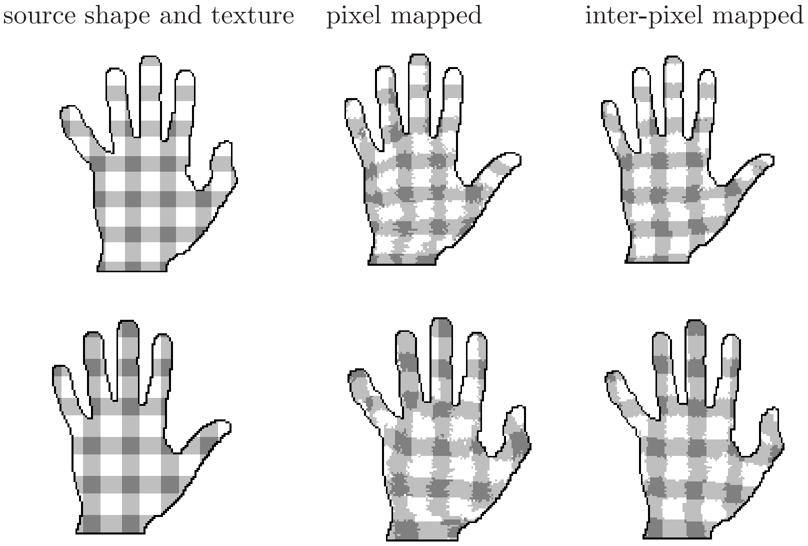


Figure 12: Texture mapping experiment. Left column: source shape and texture. Right two columns: texture mapped by corresponding points found using the two methods (Sections 5, 5.1).

zero paths, which connect the geodesic center with a boundary pixel having the maximum eccentricity. In Figure 11 the isolines of the θ coordinates are shown. The θ values computed with the inter-pixel isolines have a smoother profile as the computed *ECC* isoline lengths better approximate the continuous ones. A texture was laid on each hand - the *source*, and copied to the other one - the *destination*, by finding for each pixel $\mathbf{p}_d(r_d, \theta_d)$ of the *destination* the corresponding pixel $\mathbf{p}_s(r_s, \theta_s)$ in the *source*. Results are shown in Figure 12. Notice the smaller perturbation in the palm and the better mapping on the fingers. Note that for these two shapes, the circular arcs making the isolines in the center region do not make any sharp angles and

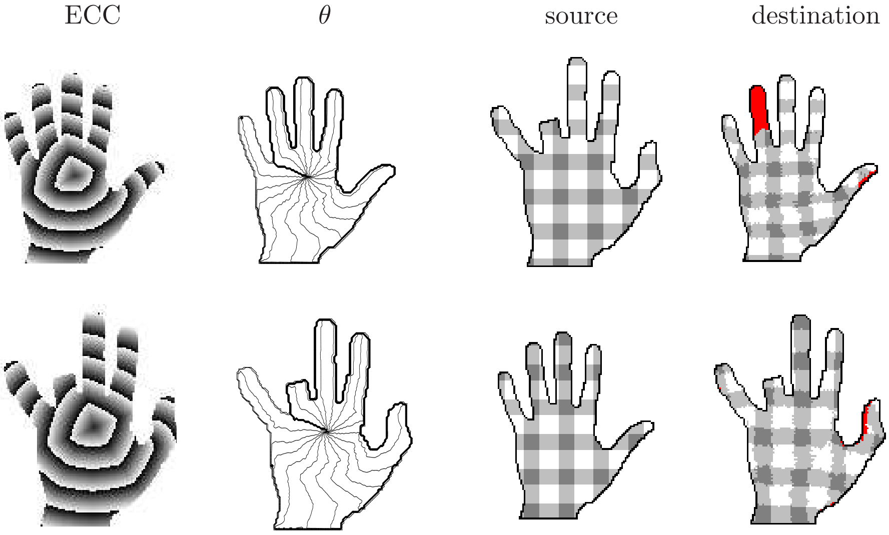


Figure 13: Correspondences with missing parts. The 'unmatched' pixels are drawn in red. (best viewed in color)

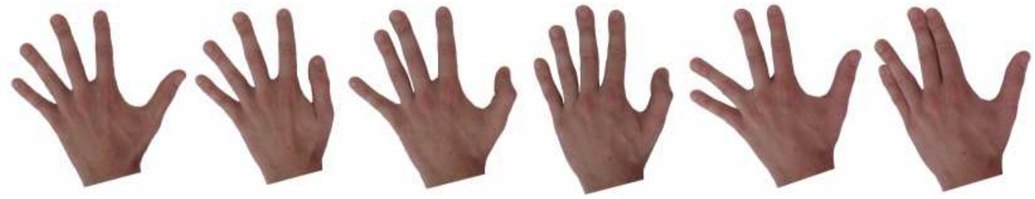


Figure 14: Six planar deformed poses of the same hand.

the problem mentioned in Section 5 does not appear. Thus, both methods could be applied.

6.1 Missing parts

Figure 13 shows the two hands with the second one having part of the ring finger cut-off. We have used a value of $t = 2$ (i.e. deviation of maximum 2 pixels) for detecting failed correspondences (Section 5.3).

6.2 Quantitative results

To measure the precision of the produced correspondences we have taken a set of six pictures of a hand undergoing close to planar deformation on a flat surface (the thumb had to turn a little in order to articulate). The poses vary

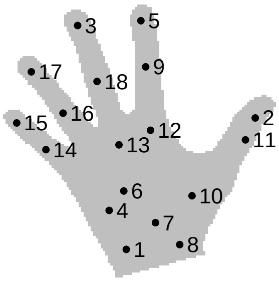


Figure 15: The positions of the 18 landmarks overlaid on the shape of the left-most hand.

from relaxed, to different kind of extreme stretching or bending, to touching respectively two neighboring fingers (Figure 14). The pictures where taken with a hand-held camera and slight changes in size and out of plane rotation are present.

We have computed the pixel correspondences between the left-most hand and each of the other five hands, resulting in five correspondence tests. To align the zero paths we have used the method in Section 5.2 and took the boundary pixel with the smallest eccentricity (located between the thumb and the pointer). We have labeled by hand 18 landmark positions in each of the 6 images. These landmarks are located in easily distinguishable positions e.g. base of the nails, birth marks, etc. Figure 15 shows the positions of the 18 landmarks overlaid on the shape of the left-most hand.

As scaling changes appear, an “absolute” scale was computed for each of the 6 hand shapes as the average of the distances between all pairs of landmark positions on the same hand. The scales computed for each image are used to factor out global changes in size, the reported ratios of geodesic distance between pairs of corresponding points are influenced only by local deformation and pixel-correspondence computation. To get a feeling of the amount of deformation present in the test, we give for the five correspondence tests the smallest ratios of the geodesic distances between any pair of landmarks as measured in both shapes: 0.79, 0.86, 0.83, 0.88, respectively 0.12. The very large change (small ratio) in the 5th test is due to the geodesic distance changing significantly between the tips of the touched fingers (e.g. between landmark 3 and 5).

The normalized[10] mean geodesic distances from the geodesic centers of the six hand poses to the 18 landmarks are: 0.61, 0.62, 0.61, 0.60, 0.61,

[10] The geodesic distances where divided by the computed image scale i.e. the mean pairwise distances between the ground truth landmark positions.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	38	6	**1**	27	17	16	25	25	19	19	9	17	19	24	7	15	9	5
2	17	10	5	15	6	6	14	14	6	9	14	11	10	8	7	9	5	7
3	37	9	7	28	17	16	23	17	19	11	14	11	7	71	**117**	34	**177**	11
4	6	3	12	3	8	3	3	3	5	5	8	5	16	9	7	39	84	42
5	23	13	12	9	9	7	13	15	8	10	21	8	17	13	**132**	4	6	8

Table 1: Deviation between the ground truth positions of the 18 landmarks (Figure 15 and 14) and the positions obtained computing the corresponding pixel using the coordinate system. The shown values are percentage of geodesic distances over the distance between landmarks 5 and 12 i.e. a value of 100 means that the computed corresponding pixel is at a distance to its ground truth position equal with the distance between landmarks 5 and 15. Columns: landmarks, Rows: correspondence test. (See the text for more details.)

respectively 0.63. The small variation shows stability of the position of the geodesic center which is taken as the origin of the coordinate system.

Table 1 shows the geodesic distances between the ground truth positions of the landmarks and the positions obtained computing the corresponding pixel using the coordinate system. For every pair of shapes two measurements can be done depending on which shape is the source and which the destination. The numbers shown are the average of mapping in both directions and are percentage values relative to the distance between ground truth positions of landmarks 5 and 12, which is one of the most stable distances. E.g. 100 means a distance between the computed and ground truth position of the landmark that is equal to the distance between the ground truth positions of landmarks 5 and 12. As in the experiment with the missing finger part we have used $t = 2$ to detect failed correspondences. In case of a failed correspondence the closest pixel with an existing correspondence is found and its corresponding pixel taken.

The largest displacement appears for the landmarks 15 and 17, both located on fingers. This is due to offsets in mapping the theta domains to neighboring fingers which results in the computed corresponding pixel shifting along $r =$const to the neighboring finger (large geodesic distance). Another landmark with rather high variation is 1, partly due to non-uniform changes in geodesic distances (the largest one is used to normalize the r coordinate), partly due to a similar effect as with the fingers.

Scaling To see the effect of discretization on the coordinate system, we computed correspondences between the first hand in Figure 15 at normal and double resolution. The different resolutions have produced minor differences

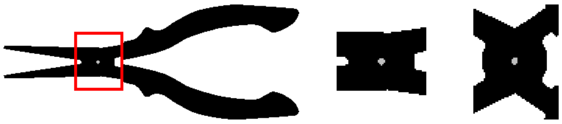


Figure 16: Left: the 2D shape of a tool from [7]. Right: zoom-in on the articulation (marked with a rectangle) with the tool closed respectively open.

in the obtained decompositions, leading to small differences in the computed theta values. All 18 landmarks had corresponding pixels in the coordinate systems, with a maximum respectively mean deviation to their ground truth positions of 10% respectively 5% of the distance between landmarks 5 and 12.

7 Discussion

Except for the zero-path no other point/part correspondences are taken as input, and once the zero path is given, mapping the coordinate systems is done independently on each shape. As a result the obtained coordinates rely strongly on the robustness of the geodesic center, the position of the zero-path and the points where isolines touch the boundary and break.

Using correlation to align the two boundary *ECC* signals is fast but is not very robust with respect to missing parts. In such cases one could consider dividing the boundary *ECC* signals into parts, finding candidate correspondences between boundary parts of the signals and then aligning those. This concept is similar in spirit to the 'local feature based' shape matching methods in [3, 20].

Using a different zero-path induces a fixed offset in the values of theta in all regions outside the center region. As no part correspondences are used, the mapping relies solely on the geometry of the isolines (length and closest points) to distribute the domains for the theta values between parts. This works well for the shape of a hand, where the size of the joint (finger) is smaller than the size of the part it is connected to (palm) and the fingers seem always attached in the same place. On the other hand for certain man made objects like the tool in Figure 16 the outer parts change the place on the boundary of the inner part where they are "attached". In such cases assignment of the theta domains for the parts is very likely to fail, resulting in pixel-correspondences between pixels of different parts.

Design decisions and future options for the coordinate system The presented coordinate system is a *skew* coordinate system. Having an *orthogonal* coordinate system would require that the isolines of the theta coordinate be orthogonal to the ones of r and go along paths of steepest descent to the geodesic center of the shape. Many geodesic paths pass through the same reference point on the inside of an articulation, thus a unique assignment of the theta values would not be possible.

In the presented approach pixels further away from the center are more likely to be affected by errors than the ones closer. If giving up the constraint that the variation of theta on the same isoline is constant, one can consider mapping the theta values going from the outer-most parts towards the center. As independent mapping of this type would be more sensitive to missing parts, one can consider jointly mapping the coordinates for both shapes, preceded by a robust algorithm to find the corresponding parts (e.g. the method in [20]).

One could also consider separating the task of finding pixel correspondences from mapping the coordinate systems. Finding all pixel correspondences could be posed as an energy minimization problem of finding a smooth mapping (minimize local deformation) which covers as many pixels as possible and neighboring pixels have neighboring correspondences. Once the mapping is computed, the coordinates can be assigned with a different and simpler strategy.

Both options require having both shapes when mapping the coordinates, but they are expected to be more robust to extreme deformations and missing parts.

8 Conclusion

We addressed the problem of finding all pixel correspondences between different poses of the same articulated shape, in the case where only the two shapes are known and one optional boundary-point correspondence is given. To each pixel distinct coordinates are associated. These coordinates are used to address corresponding pixels. Instead of mapping the coordinate system directly to the pixels where ordering problems can appear, inter-pixel isolines are extracted and the coordinate system is computed on them. The coordinates for the pixels are computed based on the computed coordinate system. Aligning the angular-like coordinate of two coordinate systems is done by aligning features of boundary pixels. Verifying failed correspondences is achieved by means of an adaptive threshold. The new method is able to identify missing parts (e.g. all pixels for which corresponding EPS coordinates are missing). Even smoother coordinates could be obtained by more refined mapping from the coordinate system in the transformed space to the pixels. Extension to non-simply connected 2D- and 3D shapes has to consider the possibility of non-convex isolines/isosurfaces, and connected components of the level sets merging at higher ECC values.

9 Acknowledgments

The presented work was partially supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13.

References

[1] Eric Andres and Marie-Andrée Jacob. The discrete analytical hyperspheres. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):75–86, 1997.

[2] Djamila Aouada, David W. Dreisigmeyer, and Hamid Krim. Geometric modeling of rigid and non-rigid 3d shapes using the global geodesic function. In *NORDIA workshop in conjunction with IEEE Conf. on CVPR*, Anchorage, Alaska, USA, June 2008. IEEE.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(4):509–522, 2002.

[4] Gunilla Borgefors. *Handbook of Pattern Recognition & Computer Vision*, chapter Digital distance transforms in 2D, 3D, and 4D. World Scientific, 3rd edition, 2005.

[5] Jean-Pierre Braquelaire, Luc Brun, and Anne Vialard. Inter-pixel euclidean paths for image analysis. In *6th Int. Conf. on Discrete Geometry for Computer Imagery (DGCI)*, volume 1176 of *LNCS*, pages 193–204. Springer, 1996.

[6] C. Brechbuhler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3-D shape-description. *Computer Vision and Image Understanding*, 61(2):154–170, March 1995.

[7] Alexander M. Bronstein, Michael M. Bronstein, Alfred M. Bruckstein, and Ron Kimmel. Analysis of two-dimensional non-rigid shapes. *Int. J. of Comp. Vis.*, 78(1):67–88, 2008.

[8] 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.

[9] Pedro F. Felzenszwalb. Representation and detection of deformable shapes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[10] Pedro F. Felzenszwalb and Joshua D. Schwartz. Hierarchical matching of deformable shapes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[11] Herbert Freeman. Computer processing of line-drawing images. *ACM Comput. Surv.*, 6(1):57–97, 1974.

[12] Lena Gorelick, Meirav Galun, Eitan Sharon, Ronen Basri, and Achi Brandt. Shape representation and classification using the poisson equation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 61–67, 2004.

[13] Adrian Ion. *The Eccentricity Transform of n-Dimensional Shapes with and without Boundary*. PhD thesis, Vienna Univ. of Technology, Faculty of Informatics, 2009.

[14] Adrian Ion, Yll Haxhimusa, and Walter G. Kropatsch. An improved coordinate system for point correspondences of 2d articulated shapes. In *15th Int. Conf. on Discrete Geometry for Computer Imagery (DGCI)*, volume 5810 of *LNCS*, pages 92–103. Springer, September 2009.

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

[16] Adrian Ion, Walter G. Kropatsch, and Eric Andres. Euclidean eccentricity transform by discrete arc paving. In *14th Int. Conf. on Discrete Geometry for Computer Imagery (DGCI)*, volume 4992 of *LNCS*, pages 213–224. Springer, 2008.

[17] J urgen Jost. *Riemannian Geometry and Geometric Analysis.* Springer, September 2005.

[18] 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.

[19] Walter G. Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer. The eccentricity transform (of a digital shape). In *13th Int. Conf. on Discrete Geometry for Computer Imagery (DGCI)*, volume 4245 of *LNCS*, pages 437–448. Springer, 2006.

[20] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Patt. Anal. Mach. Intell.*, 29(2):286–299, 2007.

[21] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In Maureen C. Stone, editor, *Proceedings of the 14st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, pages 163–169. ACM, 1987.

[22] Ozge C. Ozcanli and Benjamin B. Kimia. Generic object recognition via shock patch fragments. In *British Machine Vis. Conf.*, pages 1030–1039, 2007.

[23] Athanase Papadopoulos. *Metric Spaces, Convexity and Nonpositive Curvature.* European Mathematical Society, December 2003.

[24] Zygmunt Pizlo. *3D Shape: Its Unique Place in Visual Perception.* Springer, 2nd edition, 2002.

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

[26] A. Rosenfeld. A note on 'geometric transforms' of digital sets. *Pattern Recognition Letters*, 1(4):223–225, 1983.

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

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

[29] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. of Comp. Vis.*, 30:1–24, 1999.

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

[31] Jovisa D. Zunic, Lazar Kopanja, and Jonathan E. Fieldsend. Notes on shape orientation where the standard method does not work. *Pattern Recognition*, 39(5):856–865, 2006.