An Improved Coordinate System for Point Correspondences of 2D Articulated Shapes*

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. To find corresponding points in different poses of the same articulated shape, a non rigid coordinate system is used. Each pixel of each shape is identified by a pair of distinct coordinates. The coordinates are used to address corresponding points. This paper proposes a solution to a discretization problem identified in a previous approach. The polar like coordinate system is computed in a space where the problem cannot occur, followed by mapping the computed coordinates to pixels.

Keywords: eccentricity transform, discrete shape, coordinate system.

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 detection and recognition possible [1]. In many cases object recognition is possible even without taking other perceptual properties into account. The animals in Fig. 1 can be recognized by humans without the need of other perceptual properties.

Object recognition has to be robust against scaling, rotation, and translation, as well as against some deformation (e.g. partial occlusion, occurrence of tunnels, missing parts etc). In order to recognize objects based on their shape we assume that shapes have enough 'complexity' to allow the discrimination between different objects. This 'complex' measure has to be small for intra-class shapes (eg. shapes of an elephant), and big for inter-class shapes (e.g. shapes of elephants vs deers). Usually, the measure is not computed on the shapes directly, but on the result of an image transformation. One class of image transforms that is applied to shapes, associates to each point of the shape a value that characterizes in some way its relation to the rest of the shape. This value can be the distance to other points of the shape (e.g. landmarks). Examples of such transforms include the distance transform [2, 3], the Poisson equation [4], the global geodesic function [5], and the eccentricity transform [6].

Having a measure for the shapes, matching methods can be employed to decide which shapes are 'similar'. Many matching methods use a similarity value to 'find

* Partially supported by the Austrian Science Fund under grants S9103-N13, P18716-N13.

S. Brlek, C. Reutenauer, and X. Provençal (Eds.): DGCI 2009, LNCS 5810, pp. 92–103, 2009.

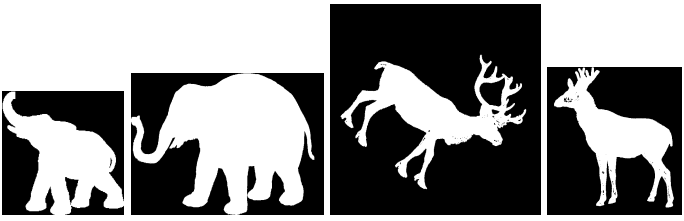

Fig. 1. Shapes of elephants and deers (taken from MPEG7 CE Shape-1 Part B)

similar shapes' (e.g. [7–9]). Similarity values are well suited to find the class of a shape, like the elephant class, based on labeled examples.

Assuming that we already have a shape class (e.g. elephants), and we are interested in finding correspondence between parts/points in shapes of the same class (e.g. trunk of an elephant), similarity values cannot be used. Some matching methods produce correspondences of the used signature, usually border points/parts (e.g. [10–12]), but finding all point correspondences based on the obtained information is in most of the cases not straightforward or even impossible. In [7], 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. 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 [13]. 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) simply connected 2D shape. In the surface parametrization community [14] a coordinate system for shapes is defined, but articulation is not considered. In [15], 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. [9], mainly due to occlusions, imperfect segmentation or feature detection.

The approach presented in this paper is similar in concept with the one in [16], but can handle cases where the method in [16] is not defined. We are interested in finding point correspondence of binary shapes of objects of the same class. 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' (i.e. shapes under an articulation create a class). We map 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. Our 2D polar like coordinate system is created on the Euclidean eccentricity transform [6]. Each pixel of each shape will be uniquely identified by a pair of coordinates, a 'radius' and an 'angle' like coordinate. Discretization results in the problems encountered in [16], where for some pixels the coordinates cannot be computed as intended. In this paper we propose a solution to this problem, and identify each pixel by a distinct pair of coordinates.

The structure of the paper is as follows. In Section 2 we define the coordinate system and describe shortly the problem of finding correspondence of points. Section 3 recalls the eccentricity transform and its properties relevant for this paper. In Section 4 we recall the coordinate system directly mapped to the pixels of the shape [16]. Section 5 describes the proposed solution, with the experiments given in Section 5.2. Section 6 concludes the paper.

2 Coordinate System: Definition, Point Correspondences

Coordinate System: a system of *curvilinear coordinates* [17] is 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. 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 1. (Coordinate map) *A 2D coordinate map, is an injective map* $coord(\mathbf{p}) : \mathcal{S} \rightarrow \mathbb{R} \times \mathbb{R}$ *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.*

Finding point correspondences: Consider the **continuous shapes** $\mathcal{S}_1$ and $\mathcal{S}_2$, and the continuous coordinate maps $coord_1$ and $coord_2$ corresponding to $\mathcal{S}_1$ respectively $\mathcal{S}_2$. 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 $\mathcal{S}_1$ and $\mathcal{S}_2$ two **discrete shapes**, and the discrete maps $dcoord_1, dcoord_2$, a point $\mathbf{q} \in \mathcal{S}_2$ with the exact 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 also depend 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 (the modified ℓ_2-norm $||(\Delta r, r \cdot \Delta\theta)||$ cannot be used in our case, as the variation of one coordinate depends on both coordinate values).

We employ a two step scheme (Section 4 gives details): given a point $\mathbf{p} \in \mathcal{S}_1$ with $dcoord_1(\mathbf{p}) = (c_1, c_2)$:

1. use the **first coordinate** to find points $\mathbf{q} \in \mathcal{S}_2$ that have 'approximately' the same 'first' coordinate $dcoord_2(\mathbf{q}) = (c_1, c)$; and
2. use the **second coordinate** to select from the above the point that minimizes the difference in the second coordinate $|c - c_2|$.

3 The Eccentricity Transform

The definitions and properties follow [6, 18]. Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^2$ (a 2D shape) and $\partial\mathcal{S}$ be its boundary[1]. A 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:

$$d(\mathbf{p}_1, \mathbf{p}_2) = \min\{\lambda(\pi(\mathbf{p}_1, \mathbf{p}_2)) \mid \pi \in \Pi\} \tag{1}$$

$$\text{where } \lambda(\pi) = \int_0^1 |\dot{\pi}(t)| dt \,, \tag{2}$$

$\pi(t)$ 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* of a point $\mathbf{p} \in \mathcal{S}$ is defined as:

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

The *eccentricity transform* $ECC(\mathcal{S})$ 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 points farthest away.

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*.

Fig. 2 shows two hand shapes (taken from the Kimia99 database [19]) and their eccentricity transform. Note the differences on the fingers of the shapes.

Fig. 2. Eccentricity transform of two shapes from the hand class. Geodesic center marked with '●'. (gray values are the eccentricity value modulo a constant.)

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

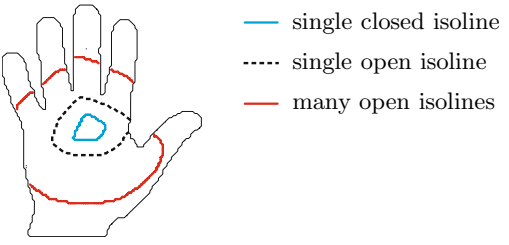

Fig. 3. Level sets made of different configurations of isolines

Terminology: An *eccentric point* of a point $\mathbf{p} \in \mathcal{S}$ is a point $\mathbf{e} \in \mathcal{S}$ s.t. $d(\mathbf{p}, \mathbf{e}) = ECC(\mathcal{S}, \mathbf{p})$. 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 $ECC(\mathcal{S}, \mathbf{p}) = d(\mathbf{p}, \mathbf{e})$.

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 $ECC(\mathcal{S}, \mathbf{c}) = min\{ECC(\mathcal{S}, \mathbf{p}) \mid \forall \mathbf{p} \in \mathcal{S}\}$.

Properties: If $\mathcal{S}$ is simply connected, the geodesic center $\mathcal{C}$ 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 geodesic center). Also, for $\mathcal{S}$ simply connected and planar, $ECC(\mathcal{S})$ has a single local/global minimum which is $\mathcal{C}$ [20], and the shapes $\mathcal{P} = \{\mathbf{p} \in \mathcal{S} \mid ECC(\mathcal{S}, \mathbf{p}) \leqslant k\}$ for $k \in [\min(ECC(\mathcal{S})), \max(ECC(\mathcal{S}))]$ are geodesically convex in $\mathcal{S}$ (proof in [20]).

Definition 2. (Level Set [17]) *The level set of a function* $f : \mathbb{R}^n \rightarrow \mathbb{R}$*, corresponding to a value* h*, is the set of points* $\mathbf{p} \in \mathbb{R}^n$ *such that* $f(\mathbf{p}) = h$*.*

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. The connected components of the level sets are called *isolines* (Fig. 3).

Due to using geodesic distances, the variation of ECC is bounded under articulated deformation to the width of the 'joints' [10]. The transform is robust with respect to salt & pepper noise, and the positions of eccentric points and geodesic center are stable [18].

Computation: In [6] different methods to compute $ECC(\mathcal{S})$ are presented.

4 The Pixel Mapped Coordinate System

This section recalls the method in [16] and shows the cases where it fails to produce results.

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 $\mathcal{S}$ and has $r = 0$ (Fig. 2). r is a linear mapping from $ECC(\mathcal{S})$ and increases

as ECC increases. The isolines of r correspond to the isolines of $ECC(\mathcal{S})$. Closed isolines of r have corresponding θ values of $[0, 360)$ mapped along the isoline. For each pixel of $\mathcal{S}$ the 'second' coordinate θ is mapped as described in the following. Note that θ is not really an angle, just denoted intuitively so.

The coordinate system is computed as follows:

1. extract 'discrete level sets' of the ECC of $\mathcal{S}$, made out of at least 8 connected components, using:
$$h \leqslant ECC < h + 1,$$
for $h = \lfloor \min(ECC) \rfloor, \lfloor \min(ECC) \rfloor + 1, ..., \lfloor \max(ECC) \rfloor$ (Fig. 4a);
2. map the value
$$r = (h - \lfloor \min(ECC) \rfloor)/(\lfloor \max(ECC) \rfloor - \lfloor \min(ECC) \rfloor),$$
to each 'discrete level set';
3. decompose $\mathcal{S}$ into simply connected regions s.t.:
   - in each region there is at most one connected component of the 'discrete level sets' of $ECC(\mathcal{S})$;
   - all isolines in the same region are either closed or all are open;
   - the number of regions is minimal.

   This decomposition produces regions $\mathcal{R}$ where 'discrete level sets' correspond to all ECC values from some $e_1, e_1 + 1, ..., e_2$, with $\min(ECC(\mathcal{S})) \leqslant e_1 \leqslant e_2 \leqslant \max(ECC(\mathcal{S}))$, and for each e, $e_1 \leqslant e \leqslant e_2$, the pixels in $\mathcal{R}$ corresponding to e form a single at least 8 connected set;
   The region that contains the geodesic center and where all isolines are closed paths is called the *center region*.
   For example, the isolines of the level sets drawn in Fig. 3 belong to different regions.
4. use a precomputed geodesic, called the *zero path*, that connects the geodesic center with the border of the shape, to choose the pixel with $\theta = 0$ on each isoline of the *center region* i.e. the region where all isolines are closed (Fig. 4b). This geodesic can for example connect the geodesic center with a point having maximum eccentricity (Fig. 9), but could also use corresponding border points obtained by a matching method (e.g. [10]), or can be chosen to maximize the number of matched pixels between two given shapes $\mathcal{S}_1, \mathcal{S}_2$.
5. map angle values $[0, 360)$ to each closed isoline in the center region: $\theta = 0$ was computed before, the other values are given proportional to the arc length, along the same orientation (Fig. 4b);
6. for each region:
   - the 'start' and 'end' θ of the isolines $(\theta_{st}, \theta_{en})$ in the region are obtained by projecting the end points of the isoline with the smallest ECC to the isoline with the highest ECC in the adjacent region with θ already computed (Fig. 5a).
   - map angle values $[\theta_{st}, \theta_{en})$ to each isoline: θ_{st} and θ_{en} is given to the end points of each isoline, the other values are given proportional to the arc length (Fig. 5b);

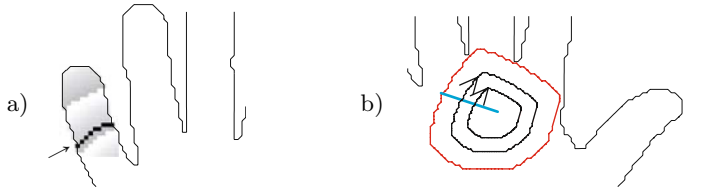

Fig. 4. Map the coordinate system: a) discrete isolines, b) zero θ in the center region

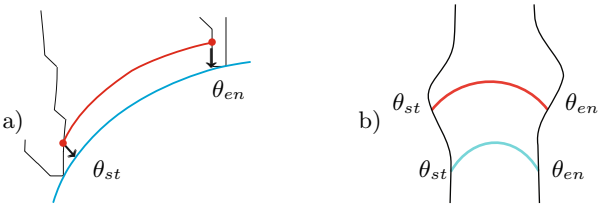

Fig. 5. Start and end theta θ_{st}, θ_{en} for isolines outside the center region: a) project from neighboring region, b) keep constant inside region

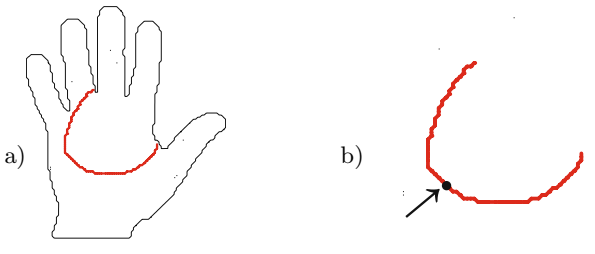

Fig. 6. Finding point correspondences: a) identify level set, b) find 'closest' point

Fig. 7. Discretization problem

Finding point correspondences: To find the point $\mathbf{q} \in \mathcal{S}$ given the coordinates (c_r, c_θ) we proceed as discussed in Section 2 (Fig. 6):

1. $\mathcal{L}$ = 'discrete level set' corresponding to r;
2. find $\arg\min_{\mathbf{q}\in\mathcal{L}}\{|c'_\theta - c_\theta|\}$, where c'_θ is the θ value associated to $\mathbf{q}$.

r is chosen as the 'first coordinate' (see Section 2) 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).

4.1 Problems due to Discretization

A reference point is a point $\mathbf{m} \in \partial\mathcal{S}$ s.t. $\mathbf{m} \in \pi(\mathbf{p}, \mathbf{e})$, where $\mathbf{e}$ is an eccentric point of $\mathbf{p}$, $\mathbf{m} \neq \mathbf{p}$, $\pi(\mathbf{p}, \mathbf{e})$ is a geodesic from $\mathbf{p}$ to $\mathbf{e}$, and $d(\mathbf{p}, \mathbf{m})$ is minimal. 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 $\mathbf{m}$.

The isolines of $ECC(\mathcal{S})$ are made out of circle arcs, corresponding to different *reference points*. Depending on the position and distance to the reference points, these circle arcs can meet under very sharp angles. In the case of the center region, isolines are closed in the continuous domain. For the 'discrete level set' the pixels are at least 8 connected, but an 8 connected closed path, where each pixel would be passed through only once might not exist. Fig. 7 shows an example. In this case an ordering cannot be made between the pixels, and mapping the θ coordinate as mentioned before is not possible.

5 The Inter-pixel Coordinate System

Instead of extracting 'discrete level sets' made out of pixels, as mentioned in Section 4, we propose to extract 'inter-pixel' isolines. These isolines are polygonal lines made out of points on the line segments that connect the pixel centers. The inter-pixel isolines can be efficiently extracted using *marching squares*, the 2D version of the classical *marching cubes* [21]. The eccentricity of the points on these line segments is estimated by linear interpolation. 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 get 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.* Fig. 8 shows examples.

The inter-pixel isolines can be seen as connected paths where the ordering of the points is given and unique, and thus we can map the new coordinate system as follows:

1. extract inter-pixel isolines with marching squares;
2. map coordinate system as in Section 4 on the inter-pixel isolines extracted;
3. map the θ values from the isolines to the pixels (discussed below).

5.1 Mapping θ Values

The 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

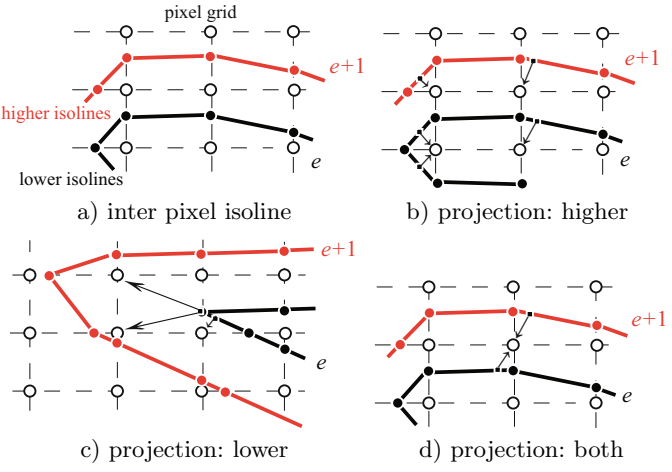

Fig. 8. Inter-pixel isolines. Projection of r, θ from the isoline to the pixels.

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 (Fig. 8a):

$$e < ECC(\mathcal{S}) \leqslant e + 1 \tag{4}$$

$$e \leqslant ECC(\mathcal{S}) < e + 1 \tag{5}$$

where $e, e{+}1$ are the eccentricity values corresponding to two consecutive r values r_1, r_2. The choice between Equations 4 and 5 depends on the method chosen to compute the θ value for each pixel and is discussed below. All pixels satisfying the chosen equation (4 or 5) are considered to have the same r coordinate.

We extract inter-pixel isolines for eccentricity values with difference one. Thus, for each pixel the number of inter-pixel isolines crossing any of the adjacent 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 (Fig. 8 illustrates the used notation):

1. **higher:** assign θ of the closest point(s) on the isoline with higher ECC value;
2. **lower:** assign θ of the closest point on the isoline with lower ECC value;
3. **both:** assign a function of θ of the closest points on both isolines.

Considering that the inter-pixel isolines are polygonal lines with angles less or equal to 180 degrees on the side with smaller ECC values, we can state the following about each option above:

higher: one pixel can get more than one θ value (Fig. 8b);
lower: more than one pixel can get the same θ value (Fig. 8c);

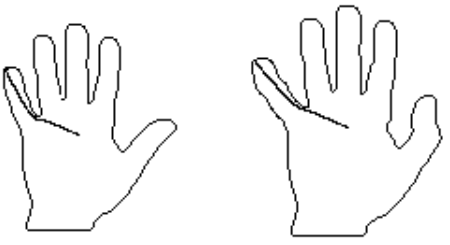

Fig. 9. Zero paths used in our experiments

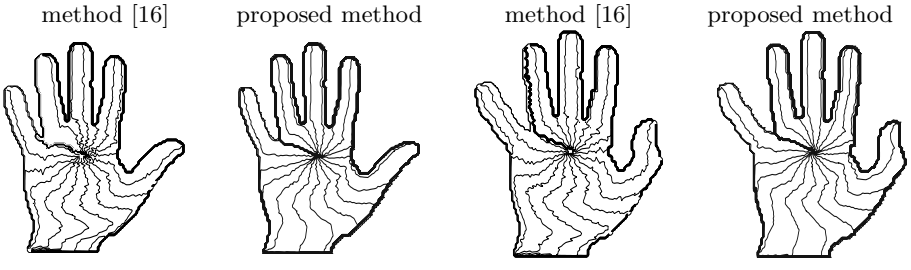

Fig. 10. Isolines of θ for the two shapes, with the two methods

both: could produce smoother results, but uniqueness of θ values is not guaranteed (Fig. 8d).

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 2) we can state that properties 'defined' and 'distinct' are required to be able to find a corresponding point. We can relax the requirements and give up property 'single', and if a pixel gets more than one θ value, consider only the smallest one.

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

5.2 Experiments

Fig. 9 shows the used zero path for the two hands in Fig. 2. In Fig. 10 the isolines of the θ coordinate mapped with the method in Section 4 and the proposed one (Section 5) is given. The θ values computed by the proposed method have a smoother profile, due to the inter-pixel isolines better approximating the continuous isolines.

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

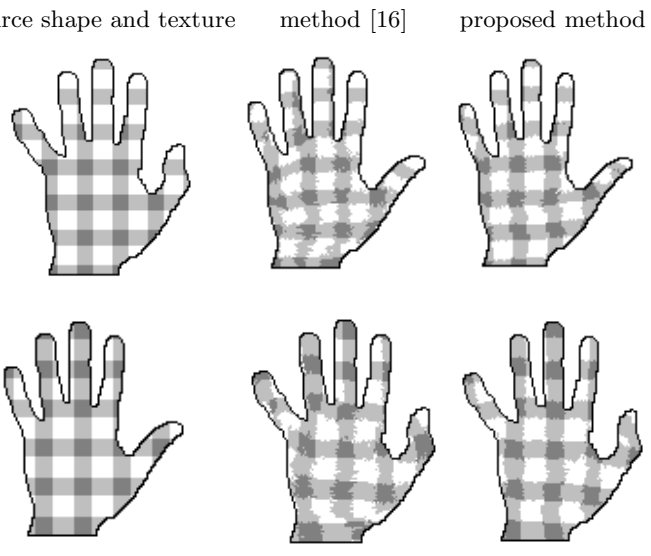

Fig. 11. Texture mapping experiment. Left column: source shape and texture. Right two columns: texture mapped by corresponding points found using the two methods.

corresponding pixel $\mathbf{p}_s(r_s, \theta_s)$ in the *source*. Results are shown in Fig. 11. Notice the smaller perturbation in the palm and the better mapping on the fingers.

Note that for these two shapes, the circle arcs making the isolines in the center region do not make any sharp angles and the problem mentioned in Section 4.1 does not appear. Thus, both methods could be applied.

6 Conclusion

This paper considered using a non-rigid coordinate system to find corresponding points in different poses of the same articulated shape. To each pixel distinct coordinates are associated. These coordinates are used to address corresponding points. A solution to a discretization problem identified in a previous approach was proposed. Instead of mapping the coordinate system directly to the pixels, the coordinate system is computed in a space where the problem cannot occur. The coordinates for the pixels are computed based on the computed coordinate system. 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 non-convex isolines, and connected components of the level sets merging at higher ECC values.

References

1. Pizlo, Z.: Shape: Its Unique Place in Visual Perception, 2nd edn. Springer, Heidelberg (2002)

2. Rosenfeld, A.: A note on 'geometric transforms' of digital sets. Pattern Recognition Letters 1(4), 223–225 (1983)
3. Borgefors, G.: Digital distance transforms in 2D, 3D, and 4D. In: Handbook of Pattern Recognition & Computer Vision, 3rd edn. World Scientific, Singapore (2005)
4. Gorelick, L., Galun, M., Sharon, E., Basri, R., Brandt, A.: Shape representation and classification using the poisson equation. In: IEEE CVPR, pp. 61–67 (2004)
5. Aouada, D., Dreisigmeyer, D.W., Krim, H.: Geometric modeling of rigid and non-rigid 3d shapes using the global geodesic function. In: NORDIA workshop in conjunction with IEEE CVPR, Anchorage, Alaska, USA. IEEE, Los Alamitos (2008)
6. Ion, A., Kropatsch, W.G., Andres, E.: Euclidean eccentricity transform by discrete arc paving. In: Coeurjolly, D., Sivignon, I., Tougne, L., Dupont, F. (eds.) DGCI 2008. LNCS, vol. 4992, pp. 213–224. Springer, Heidelberg (2008)
7. Felzenszwalb, P.F.: Representation and detection of deformable shapes. In: CVPR (2003)
8. Felzenszwalb, P.F., Schwartz, J.D.: Hierarchical matching of deformable shapes. In: CVPR (2007)
9. Ozcanli, O.C., Kimia, B.B.: Generic object recognition via shock patch fragments. In: BMVC 2007, Warwick Print, pp. 1030–1039 (2007)
10. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. IEEE TPAMI 29(2), 286–299 (2007)
11. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE TPAMI 24(4), 509–522 (2002)
12. Siddiqi, K., Shokoufandeh, A., Dickinson, S., Zucker, S.W.: Shock graphs and shape matching. International Journal of Computer Vision 30, 1–24 (1999)
13. Crum, W.R., Hartkens, T., Hill, D.L.: Non-rigid image registration: theory and practice. The British Journal of Radiology 77 Spec. No. 2 (2004)
14. Brechbuhler, C., Gerig, G., Kubler, O.: Parametrization of closed surfaces for 3-D shape-description. CVIU 61(2), 154–170 (1995)
15. Kambhamettu, C., Goldgof, D.B.: Curvature-based approach to point correspondence recovery in conformal nonrigid motion. Computer Vision, Graphics and Image Processing: Image Understanding 60(1), 26–43 (1994)
16. Ion, A., Haxhimusa, Y., Kropatsch, W.G., López Mármol, S.B.: A coordinate system for articulated 2d shape point correspondences. In: Proceedings of 19th International Conference on Pattern Recognition (ICPR), IAPR. IEEE, Los Alamitos (2008)
17. Weisstein, E.W.: CRC Concise Encyclopedia of Mathematics, 2nd edn. Chapman & Hall/CRC, Boca Raton (2002)
18. Kropatsch, W.G., Ion, A., Haxhimusa, Y., Flanitzer, T.: The eccentricity transform (of a digital shape). In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 437–448. Springer, Heidelberg (2006)
19. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE TPAMI 26(5), 550–571 (2004)
20. Ion, A.: The Eccentricity Transform of n-Dimensional Shapes with and without Boundary. PhD thesis, Vienna Univ. of Technology, Faculty of Informatics (2009)
21. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Stone, M.C. (ed.) Proceedings of the 14st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, pp. 163–169. ACM, New York (1987)