A Coordinate System for Articulated 2D Shape Point Correspondences[*]

Adrian Ion
PRIP, Vienna University of Technology

Yll Haxhimusa
Purdue University, USA

Walter G. Kropatsch
PRIP, Vienna University of Technology

Salvador B. López Mármol
PRIP, Vienna University of Technology

Abstract

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

1 Introduction

Most shape matching methods output a similarity value (e.g. [4, 5, 16, 8]), some also give correspondences of the used border points/parts [14, 1, 19], but finding all point correspondences is not straightforward.

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

For correspondences of all points of the shape, the task is similar to the non-rigid registration problem used in the medical image processing community [3]. Differences include the usage of gray scale information to compute the deformation vs. the usage of a binary shape and, the registration of a whole image (in most cases) vs. the registration of a (in this paper) connected 2D shape. In [4], a triangulation of the shape is used as a model, which could be used to find corresponding points, but an a priori known model is still needed. In the surface parametrization community [2] a coordinate system for shapes is defined, but articulation is not considered. In [9], 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. [16], mainly due to occlusions, imperfect segmentation or feature detection.

We use the eccentricity transform [7] as a basis for a 2D polar like coordinate system. To support this, shapes are decomposed into connected parts. Initial ideas are presented in [6]. Sec. 2 recalls the eccentricity transform and graph pyramids. Sec. 3 and 4 describe the proposed methods, with the experiments given in Sec. 5.

2 Eccentricity and Pyramids

Eccentricity Transform [7]: Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^2$ and $\partial\mathcal{S}$ be its border[1]. A (geodesic) path π is the continuous mapping from the interval $[0,1]$ to $\mathcal{S}$. Let $\Pi(\mathbf{p_1},\mathbf{p_2})$ be the set of all paths 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 $\pi \in \Pi(\mathbf{p_1},\mathbf{p_2})$ i.e. $d(\mathbf{p_1},\mathbf{p_2}) = \min\{\lambda(\pi(\mathbf{p_1},\mathbf{p_2}))|\pi \in \Pi\}$, where $\lambda(\pi(t)) = \int_0^1 |\dot{\pi}(t)|dt$, and $\pi(t)$ is a parametrization of the path from $\mathbf{p_1} = \pi(0)$ to $\mathbf{p_2} = \pi(1)$.

The *eccentricity transform* of $\mathcal{S}$ is $ECC(\mathcal{S},\mathbf{p}) = \max\{d(\mathbf{p},\mathbf{q})|\mathbf{q} \in \mathcal{S}\}, \forall\mathbf{p} \in \mathcal{S}$ i.e. to each point $\mathbf{p}$ it assigns the length of the shortest geodesic path(s) to the points farthest away. The class of 4-connected discrete shapes $\mathcal{S}$ defined by points on a 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*.

The shape bounded single source distance transform, $DT(\mathcal{S},\mathbf{p})$, computes the geodesic distance of all points of a shape $\mathcal{S}$ to the point $\mathbf{p}$, and is the main tool used for computing $ECC(\mathcal{S})$. $DT(\mathcal{S},\mathbf{p})$ can be efficiently computed using discrete circles [7] or fast marching [18].

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

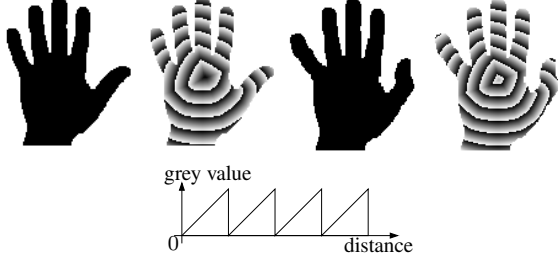[1]It can be generalized to any continuous and discrete space.

Figure 1. Eccentricity image transform.

the points on a circle, all points are eccentric and they all make up the center). The smallest eccentricity is the *radius* of the shape, and the highest one is the *diameter*.

Due to using geodesic distances, the variation of ECC is bounded under articulated deformation to the width of the 'joints' [14]. The transform is robust with respect to Salt & Pepper noise, and the positions of eccentric points and center are stable [13]. Fig. 1 shows two hand shapes (taken from the Kimia99 database [17]) and their eccentricity transform.

Irregular Graph Pyramids: A *graph pyramid* $P = \{G_0, \ldots, G_t\}$ is a stack of successively reduced graphs. Each level $G_k = (V_k, E_k)$ is obtained by *contracting* and *removing* edges in the level below. Successive levels reduce the size of the data by $\lambda > 1$. The *reduction window* relates a vertex at a level G_k with a set of vertices in the level directly below (G_{k-1}). Higher level descriptions are related to the original input data - the *receptive field* (RF) of a given vertex $v \in G_k$ aggregates all vertices in G_0 of which v is an ancestor.

Each level represents a partition of the base level into connected subgraphs i.e. *connected subsets of pixels* in our case. The construction of an irregular pyramid is iteratively local [15]. In G_0 the vertices represent single pixels. The union of neighboring vertices on level $k-1$ (children) to a vertex on level k (parent) is controlled by trees called contraction kernels (CK) [11] chosen by the algorithm (e.g. segmentation, connected component labeling, etc.). Every vertex computes its values independently of other vertices on the same level. Thus local independent (and parallel) processes propagate information up and down and laterally in the pyramid [12].

3 ECC Isoheight Lines - Decomposition

The *level set* [20] of a function $f : \mathbb{R}^n \to \mathbb{R}$, corresponding to a value h, is the set of points $\mathbf{p} \in \mathbb{R}^n$ s.t. $f(\mathbf{p}) = h$. A *level set* of the ECC of $\mathcal{S}$ is the set $LS(e) = \{\mathbf{q} \in \mathcal{S} \mid ECC(\mathcal{S}, \mathbf{q}) = e\}$, with $e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}]$. For $\mathcal{S} \in R^2$, $LS(e)$ can be a closed curve or a set of disconnected

---

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

Input: Discrete shape $\mathcal{S}$.
1: $iECC = \lfloor ECC(\mathcal{S}) \rfloor$ /*at least 8 connected IL*/
2: $G_0 \leftarrow$ oriented neighborhood graph of $iECC$
   /* pixels with same $iECC$ connected, G_0 planar,
   orient from small to high $iECC$*/
3: $k \leftarrow 0$
4: $\forall v \in V_0$ **do**
   $v.maxlength \leftarrow 1, v.ecc \leftarrow [ECC(v), ECC(v)]$
   /* init max length of isoheight lines and ecc. interval*/
5: **repeat**
6:    $A \leftarrow \{e = (v, w) \in E_k \mid v.ecc = w.ecc\}$
      /* merge isoheight line parts*/
7:    $A \leftarrow A \bigcup \{e = (v, w) \mid$ out-deg$(v) =$ in-deg$(w) = 1$
      and closed(v)=closed$(w)\}$
      /* closed(v)=true iff RF(v) contains only closed IL*/
8:    **if** $|A| > 0$ **then**
9:       $K \leftarrow$ CK as subset of A
         /*choose optimal subset of A with e.g. MIS [12] */
10:      $G_{k+1} \leftarrow$ contract(G_k, K) /* also simplify*/
11:      $\forall v \in V_{k+1}$ **compute** $v.maxlength, v.ecc$ from G_k
         /* use reduction window*/
12:      $k \leftarrow k + 1$
13: **until** $|A| = 0$
14: $t \leftarrow k$

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

open curves. The connected components of $LS(e)$ are called *isoheight lines*, $IL \subseteq LS(e)$, IL connected.

$HD(\mathcal{S}) = \{R_1, \ldots, R_n\}$ is a *a decomposition of $\mathcal{S}$ based on the connectivity of the ECC isoheight lines* (Fig. 2) if: HD is a partition of $\mathcal{S}$ into simply connected regions; $\forall R_i$ and $\forall e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}] \Rightarrow R_i \bigcap LS(e)$ is connected; the number n of regions is **minimal**. $HD(\mathcal{S})$ exists for any connected shape $\mathcal{S}$.

The top level G_t of the graph pyramid created by Alg. 1 is a region adjacency graph describing the topology of the decomposition $HD(\mathcal{S})$. Edges of G_t are oriented from regions with lower eccentricity to regions with higher eccentricity. Each vertex contains the length of the longest isoheight line in its RF.

If $\mathcal{S}$ is simply connected, the obtained region adjacency graph is a tree (Theorem 7.9 in [10]), with the RF of the root vertex containing the (unique) center pixel. Such a decomposition can be done for other transforms also (e.g. the $DT(\mathcal{S}, \mathbf{p})$). The eccentricity transform is used because its center is a robust starting point [13]. A study of the decomposition of shapes based on ECC isoheight lines in the context of shape matching is planned.

4 The Non-rigid Coordinate System

A system of *curvilinear coordinates* [20] is composed of intersecting surfaces. If all intersections are

2

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

Input: $G = (V, E)$ from Alg. 1, vertex v, interval $[\theta_1, \theta_2]$.

1: $v.\theta_1 \leftarrow \theta_1, v.\theta_2 \leftarrow \theta_2$
2: $A \leftarrow$ isoheight line of v with highest ecc.
3: **for all** $e = (v, v_o) \in E$ /*all edges oriented away*/ **do**
4:     $B \leftarrow$ isoheight line of v_o with lowest ecc.
5:     $[\theta'_1, \theta'_2] \leftarrow$ project B to A and compute from $[\theta_1, \theta_2]$ (Eq. 2)
6:   call $CtoP(G, v_o, [\theta'_1, \theta'_2])$

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

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. A single curve of each class passes through any $\mathbf{p} \in \mathcal{S}$.

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

$$\boldsymbol{r}(\mathbf{p}) = ECC(\mathcal{S}, \mathbf{p}) - \min\{ECC(\mathcal{S}, \mathbf{p})\} \quad (1)$$

is a linear mapping from the eccentricity value and the *angular coordinate* θ is mapped to the isoheight lines of the ECC based on the structure of the shape.
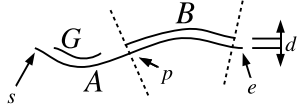
The figure above shows three adjacent isoheight lines (A, B, G) of different regions. A has eccentricity e, and B, G have $e + k$. If $k \rightarrow 0$ then $d \rightarrow 0$, and maximum smoothness of θ is achieved when each point of B has the same θ as his projection on A. This assumption puts the values θ for A and B into relation. An approximation is to project the endpoints of B onto A, to find their θ values, and interpolate along B:

$$\theta'_1 = \theta_1 + \frac{(\theta_2 - \theta_1) \int_s^p dl}{\int_s^e dl} \quad (2)$$

The root vertex of G_t from Sec. 3, contains only closed isoheight lines and is the only such vertex. Its associated θ interval is 2π. Other vertices have an 'input interval' and 0 or more 'output intervals' (edge orientation in G). Smoothness along region boundaries is assumed as above, and intervals of θ inside each region are kept constant. Alg. 2 assigns the θ intervals to each vertex. The parameters are the top level of the pyramid from Alg. 1, the root vertex of G_t, and $[0, 2\pi]$.

For the origin of θ, a path connecting the center (minimum eccentricity) with a point having the maximum eccentricity can be used. This path is called the *zero path*. It is used in the inner most region (root vertex of G_t) to set the 0 for the θ of each isoheight line. Outside this region, linear interpolation is used (Eq. 2). The point with maximum ECC can be selected using any shape orientation method (e.g. [21]) - taking into consideration the possible deformations would be optimal.

Fig. 2 shows the results of Alg. 1 and 2, and Eq. 1 and 2 for the two hands. The jagged isoheight lines of θ are due to the smoothness/roughness of the shape boundary i.e. curvature of the shape boundary at the endpoints of isoheight lines, and partly due to the simple implementation (point projection by closest point search and integral along line estimation by sum of line segment lengths for Eq. 2, etc.).

5 Experiments

A pattern was laid on each hand - the *source*, and copied to the other one - the *destination*, by finding for each pixel $p_d(r_d, \theta_d)$ of the *destination* the "closest" pixel $p_s(r_s, \theta_s)$ in the *source*. The local variation of θ is not constant over the whole shape, making the Euclidean metric not the best option for finding the closest pixel to a given point $p_d(r_d, \theta_d)$. To avoid compensating for this variation, a two step approach is used. First, normalize r in both shapes to $[0, 1]$. This makes finding $ecc_d \rightarrow r \rightarrow ecc_s$ a linear scaling problem. $L \leftarrow (ecc_s \leq ECC(source) < ecc_s + 1)$ gives at least 8 connected isoheight lines of r. Second, the pixel of L which minimizes $|\theta_d - \theta_s|$ is chosen. The results are promising (see Fig. 2) with the texture of the "articulated" finger being nicely copied from one shape to the other i.e. points are copied to their corresponding region in the articulated version of the shape.

The noise like errors on the pattern are due to the approximations mentioned above and to using "nearest point" for finding the color of each pixel when copying the pattern (instead of interpolating gray values). Errors on the boundaries of fingers are due to certain coordinates not existing in both shapes. The more global perturbation (palm of the hands in Figure 2) is mainly due to the slightly different position of the centers and isoheight line shape. Improvements can be made by considering both shapes when mapping the coordinates to them, or by a more complex method for finding corresponding points. Finding a matching between the regions of the decomposition of the two shapes is an important step planned for the future.
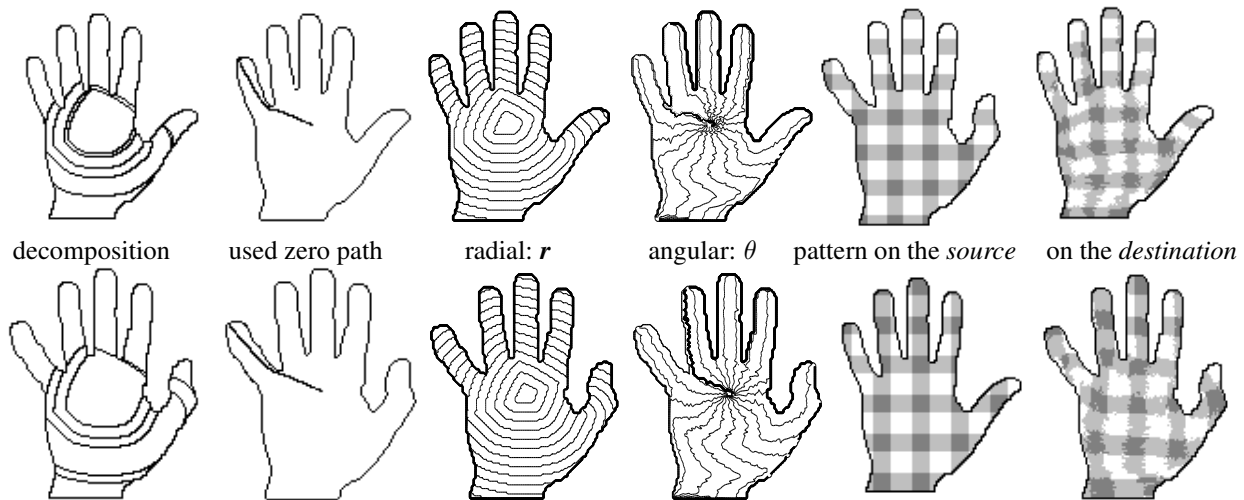
| decomposition | used zero path | radial: *r* | angular: *θ* | pattern on the *source* | on the *destination* |

Figure 2. Results for the shapes in Figure 1.

6 Conclusion and Outlook

This paper presents a framework for mapping a polar-like coordinate system to a non-rigid binary shape and finding corresponding points between two shapes. Promising initial results are presented. More global decisions will provide smoother angular isoheight lines, and additional correspondences between part structures can help to solve failed correspondences. Further quantitative evaluation and extension to non simply connected shapes is planned.

References

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

[2] C. Brechbuhler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3-D shape-description. *Comp. Vis. Image Und.*, 61(2):154–170, Mar. 1995.

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

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

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

[6] A. Ion and W. G. Kropatsch. Mapping a coordinate system to a non-rigid shape. In *32nd OAGM/AAPR*, Linz, Austria, May 2008. OCG.

[7] A. Ion, W. G. Kropatsch, and E. Andres. Euclidean eccentricity transform by discrete arc paving. In *14th DGCI*, 2008, volume 4992. Springer.

[8] A. Ion, G. Peyré, Y. Haxhimusa, S. Peltier, W. G. Kropatsch, and L. Cohen. Shape matching using the geodesic eccentricity transform - a study. In *31st OAGM/AAPR*, Schloss Krumbach, Austria, 2007. OCG.

[9] C. Kambhamettu and D. B. Goldgof. Curvature-based approach to point correspondence recovery in conformal nonrigid motion. *Comp. Vis., Graph. and Img. Proc.: Image Understanding*, 60(1):26–43, 1994.

[10] R. Klette and A. Rosenfeld. *Digital Geometry*. Morgan Kaufmann, 2004.

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

[12] W. G. Kropatsch, Y. Haxhimusa, Z. Pizlo, and G. Langs. Vision pyramids that do not grow too high. *Pattern Recognition Letters*, 26(3):319–337, 2005.

[13] W. G. Kropatsch, A. Ion, Y. Haxhimusa, and T. Flanitzer. The eccentricity transform (of a digital shape). In *13th DGCI*, volume 4245 of *LNCS*. Springer, 2006.

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

[15] P. Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, 1989.

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

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

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

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

[20] E. W. Weisstein. Mathworld–a wolfram web resource. http://mathworld.wolfram.com/.

[21] J. D. Zunic, P. L. Rosin, and L. Kopanja. On the orientability of shapes. *IEEE Trans. on IP*, 15(11):3478–3487, 2006.