BOUNDING BOX SPLITTING FOR ROBUST SHAPE CLASSIFICATION

Christian Bauckhage and John K. Tsotsos

Centre for Vision Research
York University
Toronto, ON, M3J 1P3

ABSTRACT

This paper presents a fast method to compute homeomorphisms between 2D lattices and shapes found in binary images. Unlike many other methods, this mapping is not restricted to simply connected shapes but applies to arbitrary topologies. Moreover, it provides an avenue to the embedding of shapes in vectorspaces over $\mathbb{R}$ and $\mathbb{C}$ and thus enables robust shape recognition.

1. INTRODUCTION AND MOTIVATION

Shape and silhouette provide significant cues for human visual perception. This has been documented in a vast and venerable body of literature in cognitive psychology or biological neuroscience (cf. e.g. [12, 14, 15]). It is thus no surprise that shape based approaches are popular in computer vision, too [1, 7]. Our interest in shape analysis arose from the fact that lately it has become the basis for many methods in automatic gait classification (cf. e.g. [3, 13]).

The results presented here hence originate from work on gait analysis. Dealing with automatic behavior recognition in monitored real world environments, we were in need of vision algorithms that allow a robust classification of shapes that vary throughout consecutive frames of video data. In addition, methods were required that perform in real time.

Well known shape representation techniques like moments, shocks, or shock graphs enable reliable retrieval from large image databases [5, 10, 11]. However, concerning noise corrupted shapes that almost inevitably will have to be dealt with in video processing these methods come along with lesser robustness or prohibitive computational costs. In gait analysis, image patches containing shapes are often donwsampled and treated as vectors [3]. This induces a certain overhead because the surroundings of a shape have to be considered. Dealing with shapes only, topological representations have been shown to cope well with shape dynamics. Florez et al. [4] champion the use of self organizing neural gases [8] for modeling. Even though their technique works well once a network has learned the topology of a shape, it suffers from the considerably long learning phase of the algorithm and thus does not comply with our real time constraints. However, this powerful but costly neural approach can be replaced by simple and *fast* methods from computational geometry.

In the next section, we will present a rapid shape encoding scheme that maps arbitrary shapes onto 2D lattices. Section 3 will show that this approach leads to efficient and reliable shape recognition. Note that although this technique was developed for gait analysis, it is generally applicable. To underline this claim and for the sake of comparison, the reminder of this paper will present results obtained on a standard testbed for shape matching which was introduced by Kimia et al.[10].

2. SHAPE ENCODING

Dealing with binary images, we assume a shape $\mathcal{S}$ to be a set of L pixels, $\mathcal{S} = \{\mathbf{p}_i \in \mathbb{R}^2 \mid i = 1, \ldots, L\}$. Under this assumption, the following procedure computes an $m \times n$ array of *bounding boxes* that can be thought of as a coarser representation of a shape.

```
procedure bounding box splitting ($\mathcal{S}$)
  $w \leftarrow$ width of bounding box $\mathcal{B}(\mathcal{S})$
  $x \leftarrow x$ coord. of left border of $\mathcal{B}(\mathcal{S})$
  for $j \in \{1, \ldots, n\}$
    $\mathcal{S}_j \leftarrow \{\mathbf{p} \in \mathcal{S} \mid x \leq p_x \leq x + w/n\}$
    $h \leftarrow$ height of $\mathcal{B}(\mathcal{S}_j)$
    $y \leftarrow y$ coord. of lower border of $\mathcal{B}(\mathcal{S}_j)$
    for $i \in \{1, \ldots, m\}$
      $\mathcal{S}_{ij} \leftarrow \{\mathbf{p} \in \mathcal{S}_j \mid y \leq p_y \leq y + h/m\}$
      $\mathbf{B}_{ij} \leftarrow \mathcal{B}(\mathcal{S}_{ij})$
      $y \leftarrow y + h/m$
    $x \leftarrow x + w/n$
  return($\mathbf{B}$)
```

An illustration of this procedure and its result is shown in Fig. 1. Obviously, the algorithm is linear in mn. It does not require the computation of interpixel relations nor iterative maximization (or minimization) steps but only relies on linear search. Its average complexity is therefore $O(mn\hat{p})$ where $\hat{p}$ denotes the average number of pixels in a box $\mathbf{B}_{ij}$.

0-7803-9134-9/05/$20.00 ©2005 IEEE

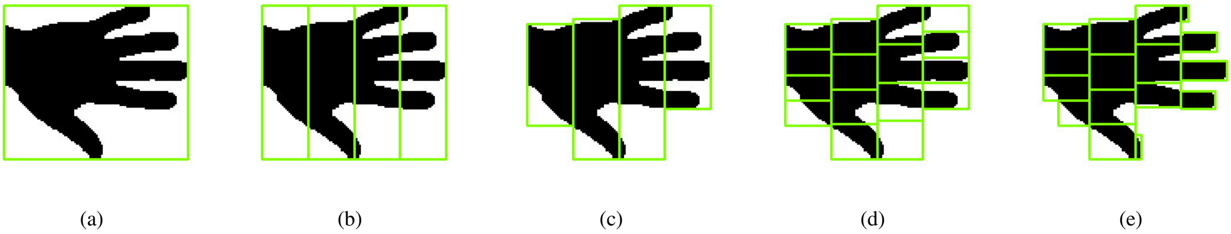
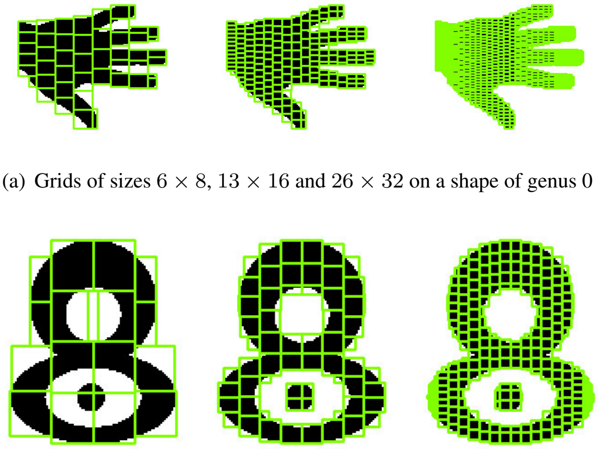


Fig. 1. The five steps of bounding box splitting: 1(a) compute the bounding box $\mathcal{B}(\mathcal{S})$ of a pixel set $\mathcal{S}$; 1(b) subdivide it into n vertical slices; 1(c) compute the bounding box $\mathcal{B}(\mathcal{S}_j)$ of each resulting pixel set $\mathcal{S}_j$ where $j = 1, \ldots, n$; 1(d) subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices; 1(e) compute the bounding box $\mathcal{B}(\mathcal{S}_{ij})$ of each resulting pixel set $\mathcal{S}_{ij}$ where $i = 1, \ldots, m$.



(a) Grids of sizes 6 × 8, 13 × 16 and 26 × 32 on a shape of genus 0

(b) Grids of sizes 4 × 4, 9 × 8 and 19 × 16 on a shape of genus 2

Fig. 2. Shape descriptions resulting from bounding box splitting cope with different topological structures; in each of these examples m was chosen according to Eq. 1.

Moreover, as each box $\mathbf{B}_{ij}$ can be understood as a generalized pixel of height h_{ij} and width w_{ij} at location $\mathbf{x}_{ij}$, the storage requirement of a corse shape is a mere $4mn$. For small values of m and n bounding box splitting will thus be fast and storage efficient.

Figures 2(a) and 2(b) show that already moderate $m \times n$ arrays produce fairly accurate representations. Table 1 summarizes measurements obtained from 266 shapes of an average size of 17887 pixels. Given the height h and width w of a shape's initial bounding box, m was computed as a function of n:

$$m(n) = \left\lfloor \frac{h}{w} n \right\rfloor \tag{1}$$

where $\lfloor x \rfloor$ indicates rounding $x \in \mathbb{R}$ to the nearest lower integer, i.e. $\lfloor x \rfloor = \sup\{y \in \mathbb{N} \mid y \leq x\}$. The table lists the average compression rate, the normalized Hamming distance $D = d_H(\mathcal{S}, \mathbf{B})/L$ between a shape $\mathcal{S}$ and its coarse representation $\mathbf{B}$ as well as the variance of these distances. While the compressions rate decreases slowly with a growing n, the reconstruction error decreases quickly. At a compression rate of 90%, it already drops below 6%.

n	compression rate	D	σ^2
4	99.9 %	0.532	0.0414
8	99.4 %	0.228	0.0054
16	97.4 %	0.103	0.0009
32	90.0 %	0.055	0.0002
64	59.2 %	0.038	0.0001

Table 1. Average compression rates, normalized Hamming distances and variances obtained from bounding box splitting with n columns and m resulting from Eq. 1.

Therefore, even bounding box arrays of small sizes provide shape descriptions that are precise enough for most video based applications. The runtime required for bounding box splitting will thus be far from threatening real time constraints. Furthermore, this simple method relying on basic computational geometry procedures is not restricted to simply connected shapes or shapes that are homeomorph to the unit circle. Figure 2 exemplifies how it performs on shape with a genus greater than 0, i.e. on shapes with holes. As the next section will show, this increases robustness against shape distortions.

3. SHAPE RECOGNITION

Bounding box splitting realizes a homeomorphism between 2D shapes and 2D lattices: bounding boxes $\mathbf{B}_{ij}$ below other boxes will always have lesser lattice coordinates i, boxes left of other boxes will always come along with lesser lattice coordinates j and vice versa (s. Fig. 3). Due to this topology preserving nature of the bounding box representation of shapes, a consistent vector space embedding of shapes is

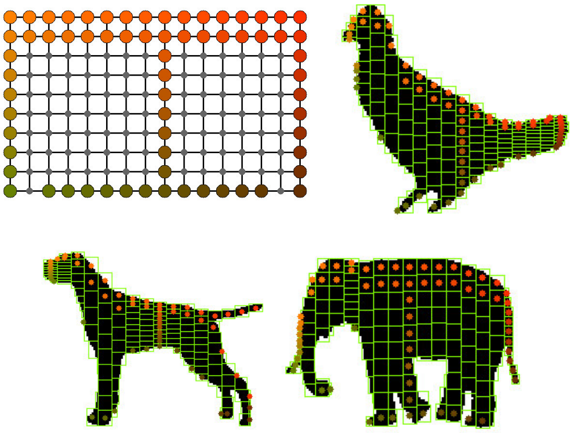


Fig. 3. A sample of $k = 67$ points on lattice of size 10×16 and examples of how it is mapped onto different shapes.

straightforward[1].

If $\mathbf{v}$ denotes the location of the bottom left corner of the initial bounding box of $\mathcal{S}$, w and h denote its width and height and $\mathbf{u}^{ij}$ denotes the center of box $\mathbf{B}_{ij}$, then the coordinates

$$\boldsymbol{\mu}^{ij} = \begin{pmatrix} (u_x^{ij} - v_x)/w \\ (u_y^{ij} - v_y)/h \end{pmatrix} \tag{2}$$

provide a scale invariant representation of $\mathcal{S}$. Sampling k points of an $m \times n$ lattice therefore allows to represent $\mathcal{S}$ as a vector

$$\mathbf{r} = [\mu_x^{i(1)j(1)}, \mu_y^{i(1)j(1)}, \ldots, \mu_x^{i(k)j(k)}, \mu_y^{i(k)j(k)}] \in \mathbb{R}^{2k}$$

where $i(\alpha) < i(\beta)$ if $\alpha < \beta$ and likewise for the index j.

As this embedding leads to vectors in $[0,1]^{2k}$, the Euclidian distance between two shapes $\mathcal{S}_a$ and $\mathcal{S}_b$ can be scaled by the maximum possible distance, leading to the normalized shape distance

$$d_E(\mathbf{r}_a, \mathbf{r}_b) = \|\mathbf{r}_a - \mathbf{r}_b\| \;/\; \left(\sum_{2k} 1^2\right)^{\frac{1}{2}} \tag{3}$$

such that $0 \leq d_E \leq 1$.

Apart from real vector spaces, the mapping between lattices and shapes also enables complex vector space embeddings. To this end, we consider the center $\boldsymbol{\mu}$ of the initial bounding box of $\mathcal{S}$ and the centers $\mathbf{u}^{ij}$ of the boxes $\mathbf{B}_{ij}$ and compute complex numbers

$$z^{ij} = (u_x^{ij} - \mu_x)/2w + \boldsymbol{i}(u_y^{ij} - \mu_y)/2h \tag{4}$$

which guarantee that $\mathrm{Re}(z^{ij}), \mathrm{Im}(z^{ij}) \in [-1, +1]$. Sampling k points of an $m \times n$ lattice thus results in vectors

$$\mathbf{z} = [z^{i(1)j(1)}, \ldots, z^{i(k)j(k)}] \in \mathbb{C}^k.$$

For any two shapes $\mathcal{S}_a$ and $\mathcal{S}_b$ encoded by means of complex vectors $\mathbf{z}_a, \mathbf{z}_b$ we can hence determine their Procrustean distance [2, 13]. With $\mathbf{z}^*$ denoting the conjugate transpose of a complex vector $\mathbf{z}$, this common measure in statistical shape analysis is given by

$$d_P(\mathbf{z}_a, \mathbf{z}_b) = \sqrt{1 - \frac{\mathbf{z}_a^* \mathbf{z}_b \mathbf{z}_b^* \mathbf{z}_a}{\mathbf{z}_a^* \mathbf{z}_a \mathbf{z}_b^* \mathbf{z}_b}} \tag{5}$$

where we will always have $0 \leq d_P \leq 1$.

Note that while both embeddings are scale invariant, they are not invariant against rotations. However, since our interest in shapes originated from gait analysis, rotation invariance was not our primary concern (shapes of walking people usually appear to be upright). In our experiments conducted for this paper we therefore focused on natural, unrotated shapes in the database collected by Kimia et al.

Table 2 shows examples of Euclidian and Procrustean distances between shape vectors resulting from our method. Each of the shapes shown in the table was mapped onto a lattice of size 10×16. The figures in the first two rows of each quadruple of rows result from sampling the 67 lattice points indicated in Fig. 3. The first row in each quadruple lists the normalized Euclidian distance in the corresponding 134 dimensional real vectorspace; each second row describes the Procrustean distance in $\mathbb{C}^{67}$. The third and fourth row in each quadruple indicate the corresponding distances after sampling the whole lattice. It is noticeable that shapes of the same class have distances ≤ 0.1 regardless of the type of field and dimension of the corresponding vectorspace. Distances between different classes vary between 0.1 and 0.42. Procrustean distances seem to be more accurate in the sense that they are smaller for similar shapes and larger for different classes of shapes. Note that that the four lowermost shapes in the table have distorted boundaries. Nevertheless, our method yields reasonable distances.

As the maximum distance found in the table is 0.42, shape vectors seem to occupy only a small volume within the vectorspaces considered here. We thus investigated if reliable shape classification is possible at all, that is if meaningful boundaries between classes can be established. To this end, we considered a total of 145 images showing 7 classes of animal shapes (birds, cats, dogs, dinosaurs, elephants, fishes and rabbits). $\mathbb{R}^{134}$ embeddings of 96 of these shapes were used to train support vector machines with radial basis kernel functions. Applying the *one against one* paradigm in training the classifiers and using a majority voting scheme for classification produced very accurate results. Out of the 49 test shapes only two were misclassified, leading to a correct classification rate of 96%.

[1] Recent contributors to shape matching argued that geodesics in shape spaces resulting from conformal mappings or Riemannian manifold embeddings are better suited for shape characterization [6, 9]. Of course, rigorous approaches like this are very appealing, but if it comes to speed and simplicity, vector spaces have their merits; first and foremost they provide access to some powerful techniques well established in pattern recognition and machine learning.

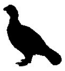
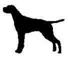
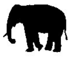
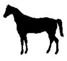
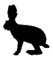
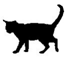
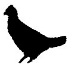
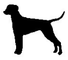
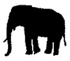
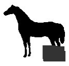
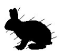
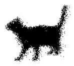
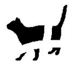
						
	0.10	0.14	0.20	0.14	0.09	0.17
	0.08	0.20	0.33	0.22	0.10	0.19
	0.09	0.13	0.17	0.13	0.08	0.14
	0.07	0.21	0.34	0.24	0.07	0.21
	0.17	0.08	0.19	0.11	0.17	0.21
	0.24	0.07	0.28	0.14	0.30	0.30
	0.15	0.08	0.16	0.11	0.16	0.18
	0.28	0.07	0.31	0.18	0.31	0.34
	0.23	0.24	0.08	0.20	0.20	0.20
	0.37	0.38	0.05	0.34	0.31	0.38
	0.20	0.21	0.07	0.18	0.18	0.19
	0.39	0.42	0.06	0.35	0.35	0.41
	0.16	0.10	0.17	0.09	0.15	0.22
	0.25	0.10	0.20	0.07	0.24	0.27
	0.15	0.11	0.16	0.09	0.14	0.21
	0.26	0.15	0.22	0.10	0.23	0.34
	0.12	0.15	0.13	0.15	0.08	0.18
	0.12	0.20	0.15	0.25	0.06	0.25
	0.10	0.14	0.12	0.14	0.08	0.14
	0.10	0.22	0.17	0.26	0.06	0.20
	0.25	0.25	0.17	0.18	0.22	0.06
	0.35	0.35	0.27	0.23	0.31	0.04
	0.19	0.20	0.15	0.17	0.17	0.06
	0.28	0.35	0.28	0.28	0.23	0.06
	0.23	0.24	0.19	0.18	0.21	0.04
	0.32	0.36	0.33	0.26	0.30	0.06
	0.19	0.21	0.18	0.19	0.17	0.08
	0.28	0.40	0.38	0.35	0.25	0.06

Table 2. Exemplary distances between shapes. In each quadruple of lines, the first line describes the normalized Euclidean distance in $\mathbb{R}^{134}$ and the second line lists the Procrustean distance in $\mathbb{C}^{67}$. The third and fourth line contain the corresponding distances in $\mathbb{R}^{320}$ and $\mathbb{C}^{160}$, respectively.

4. CONCLUSION

This paper presented a simple computational geometry approach to compute homeomorphisms between shapes and lattices. The proposed method copes with shapes of different topologies, is storage and time efficient. Since the proposed mapping is topology preserving, orders defined on samples of lattice points lead to consistent vectorspace embeddings of shapes. Experimental results underline that real or complex valued embeddings based on our approach are robust against noisy shape boundaries. Moreover, applying ensembles of support vector machines enables very accurate shape classification.

5. REFERENCES

[1] L. da F. Costa and R. M. Cesar. *Shape Analysis and Classification.* CRC Press, 2000.

[2] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis.* Wiley, Chichester, 1998.

[3] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proc. CVPR*, volume II, pages 681–688, 2004.

[4] F. Florez, J.M. Garcia, J. Garcia, and A. Hernandez. Hand gesture recognition following the dynamics of a topology preserving network. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, pages 303–308, 2002.

[5] A.K. Jain and A. Vailaya. Shape-based retrieval: A case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, 1998.

[6] E. Klassen, A. Srivastava, W. Mio, and S.H. Joshi. Analysis of planar shapes using geodesic paths on shape space. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3):372–383, 2004.

[7] S. Loncaric. A survey of shape analysis techniques. *Pattern Recogntion*, 31(8):983–1001, 1998.

[8] T. Martinetz and K. Schulten. A neural gas network learns topologies. In T. Kohonen, editor, *Artificial Neural Networks*, pages 397–402. Elsevier, 1991.

[9] E. Sharon and D. Mumford. 2d-shape analysis using conformal mapping. In *Proc. CVPR*, volume II, pages 350–357, 2004.

[10] D. Sharvit, J. Chan, H. Tek, and B.B. Kimia. Symmetry-based indexing of image databases. *J. of Visual Communication and Image Representation*, 9(4):366–380, 1998.

[11] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *Int. J. of Computer Vision*, 35(1):13–32, 1999.

[12] M.J. Tarr and H.H. Bülthoff, editors. *Object Recognition in Man, Monkey, and Machine.* MIT Press, Cambridge, 1999.

[13] L. Wang, H. Ning, W. Hu, and T. Tan. Gait recognition based on procrustes shape analysis. In *Proc. ICIP*, volume III, pages 433–436, 2002.

[14] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt I. *Psychologische Forschung*, 1:47–58, 1922.

[15] L. Zusne. *Visual Perception of Form.* Academic Press, New York, 1970.