Part-based Shape Retrieval

Mirela Tanase
Department of Computer Science
Utrecht University, The Netherlands
mirela@cs.uu.nl

Remco C. Veltkamp
Department of Computer Science
Utrecht University, The Netherlands
Remco.Veltkamp@cs.uu.nl

ABSTRACT

This paper introduces a measure for computing the dissimilarity between multiple polylines and a polygon based on the turning function, and describes a part-based retrieval system using that dissimilarity measure. This dissimilarity can be efficiently computed in time $O(kmn \log mn)$, where m denotes the number of vertices in the polygon, and n is the total number of vertices in the k polylines that are matched against the polygon. This dissimilarity measure identifies similarities even when a significant portion of one shape is different from the other, for example because the shape is articulated, or because of occlusion or distortion. The effectiveness of the dissimilarity measure in demonstrated in a part-based shape retrieval system. Quantitative experimental verification is performed with a known ground-truth, the MPEG-7 Core Experiment test set, in a comparison with the Curvature Scale Space method, and a global turning angle function method.

Categories and Subject Descriptors: I.5 Pattern Recognition, I.3.5 Computational Geometry.
General Terms: Algorithms.
Keywords: Shape matching, retrieval.

1. INTRODUCTION

The motivation for part-based shape retrieval is twofold. Firstly, partial matching is an important problem that has not received much attention. The matching of shapes has been done mostly by comparing them as a whole [1, 9, 12]. Such a matching fails when a significant part of one shape is occluded, for example. In this paper, we address the partial shape matching problem, which is concerned with matching portions of two given shapes.

Secondly, partial matching methods are effective for database retrieval problems. Partial matching helps identifying similarities even when a significant portion of one shape boundary is occluded, or seriously distorted. It could also help in identifying similarities between contours of an articulated object in different configurations of its moving parts, like the contours of a sitting and a walking cat. Partial matching also helps alleviating the problem of unreliable object segmentation from images, over or undersegmentation, giving only partially correct contours.

MM'05, November 6–11, 2005, Singapore.

Contribution Firstly, we introduce a measure for computing the dissimilarity between multiple polylines and a polygon, see figure 1. The polylines could for example be pieces of an object contour incompletely extracted from an image, or could be boundary parts in a decomposition of an object contour. The measure we propose is based on the turning function representation, and can be computed in $O(kmn \log mn)$ time, where m denotes the number of vertices in the polygon, and n is the total number of vertices in the k polylines that are matched against the polygon.

Secondly, we describe a part-based shape retrieval system. Given a large collection of shapes, and a query consisting of a set of polylines, we want to retrieve those shapes in the collection that best match our query. The set of polylines forming the query are boundary parts in a decomposition of a database shape – both this database shape and the parts in the query are selected by the user. The quantitative evaluation on the basis of a known ground-truth indicate that a part-based approach to matching consistently improves the global matching performance for difficult categories of shapes.

1.1 Previous Work

Most of the approaches to partial shape matching are based on computing local features of the contour, and then looking for correspondences between the features of the two shapes, see e.g. [8], [11]. These local features-based solutions work well when the matched subparts are almost equivalent up to a transformation such as translation, rotation or scaling, because for such subparts the sequences of local features are very similar. This makes them useful for applications like detecting instances of a model shape in a cluttered scene. However, the problem of shape-based retrieval in a general database is more involved, since it requires to report matchings between subparts that we perceive as similar, but may have quite different local features (different number of curvature

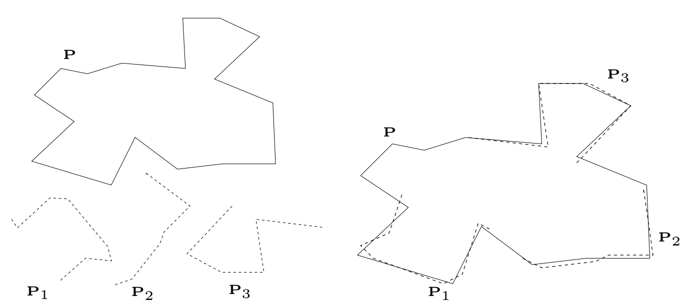

Figure 1: Matching an ordered set $\{P_1, P_2, P_3\}$ of polylines against a polygon P.

extrema for example).

Geometric hashing [15] is designed for partial matching, but does not easily transform to our case of matching multiple polylines to a polygon. Also the Hausdorff distance allows partial matching. It is used by [4] for image matching, but only for discrete point sets.

Partial matching based on the turning function (see section 2.1) of two polylines under scaling, translation and rotation, can be done in time $O(m^2n^2)$, see [2]. However, that works for only two single polylines.

In this paper we address the partial shape matching problem for the purpose of shape-based retrieval. Another approach to this problem is that of Latecki et al. [7], which is based on establishing the best correspondence of parts in a decomposition of the matched shapes, simplified by a discrete curve evolution process. A serious drawback of this approach is that the matching is done between parts of simplified shapes at "the appropriate evolution stage", which is not further detailed, though it certainly has an effect on the quality of the matching process.

2. POLYLINES-TO-POLYGON MATCHING

In this section we concentrate on the problem of matching an ordered set $\{P_1, P_2, \ldots, P_k\}$ of k polylines against a polygon P (see figure 1). For this purpose, the polylines are rotated and shifted along the polygon P, in such a way that the pieces of the boundary of P "covered" by the k polylines are mutually disjoint except possibly at their endpoints.

2.1 Turning function

The *turning function* Θ_A of a polygon A measures the angle of the counterclockwise tangent with respect to a reference orientation as a function of the arc-length s, measured from some reference point on the boundary of A. It is a piece-wise constant function, with jumps corresponding to the vertices of A. A rotation of A by an angle θ corresponds to a shifting of Θ_A over a distance θ in the vertical direction. Moving the location of the reference point $A(0)$ over a distance $t \in [0, l_A)$ along the boundary of A corresponds to shifting Θ_A horizontally over a distance t.

The distance between *two polygons* A and B is defined as the L_2 norm between their two turning functions Θ_A and Θ_B, minimized with respect to the vertical and horizontal shifts of these functions (in other words, minimized with respect to rotation and choice of reference point). More formally, suppose A and B are two polygons with perimeter length l_A and l_B, respectively, and the polygon B is placed over A in such a way that the reference point $B(0)$ of B coincides with point $A(t)$ at distance t along A from the reference point $A(0)$, and B is rotated clockwise by an angle θ with respect to the reference orientation. We define the quadratic dissimilarity $\mathsf{f}(A, B, t, \theta)$ between A and B for a given placement (t, θ) of B over A, as the square of the L_2 norm between their two turning functions Θ_A and Θ_B, shifted relative to each other corresponding to the values of t and θ:

$$\mathsf{f}(A, B, t, \theta) = \int_0^{l_B} (\Theta_A(s + t) - \Theta_B(s) + \theta)^2 ds.$$

The dissimilarity between two polygons A and B is then given by:

$$d(A, B) = \min_{\theta \in \mathbb{R},\, t \in [0, l_A)} \sqrt{\mathsf{f}(A, B, t, \theta)}.$$

To achieve invariance under scaling, Arkin et al. [1] propose to normalize the two polygons to unit length prior to the matching.

For measuring the difference between a polygon A and a polyline B the same measure can be used. For the purpose of our part-based retrieval application, we want that a polyline B included in a polygon A to match polygon A perfectly, that is: their dissimilarity should be zero. For this reason we do not scale the polyline or the polygon prior to the matching process. Thus, our dissimilarity measure is not scale-invariant. In our part-based retrieval application (see section 3) we achieve robustness to scaling by normalizing all shapes in the collection to the same diameter of their circumscribed circle.

The turning function is sensitive to unevenly spread noise, since that distorts the parameterization of the curves. Evenly spread noise is less a problem.

2.2 Dissimilarity measure

The new dissimilarity measure we introduce in this paper is based on the turning function, but it is redesigned to measure the dissimilarity between *a set of polylines and a polygon*. Let $\Theta : [0, l] \to \mathbb{R}$ be the turning function of a polygon P with m vertices, and of perimeter length l. Since P is a closed polygon, the domain of Θ can be easily extended to the entire real line, by $\Theta(s + c\ l) = \Theta(s) + c\ 2\pi$, for c any integer. Let $\{P_1, P_2, \ldots P_k\}$ be a set of polylines, and let $\Theta_j : [0, l_j] \to \mathbb{R}$ denote the turning function of the polyline P_j of length l_j. If P_j is made of n_j segments, Θ_j is piecewise-constant with $n_j - 1$ jumps (see figure 2).

For simplicity of exposition, we denote by $\mathsf{f}_j(t, \theta)$ the quadratic dissimilarity $\mathsf{f}(P, P_j, t, \theta)$ between the polyline P_j and the polygon P, for a given placement (t, θ) of P_j over P. Thus,

$$\mathsf{f}_j(t, \theta) = \int_0^{l_j} (\Theta(s + t) - \Theta_j(s) + \theta)^2 ds.$$

We define a measure for the dissimilarity between an ordered set $\{P_1, P_2, \ldots, P_k\}$ of k polylines and a polygon P. We assume the polylines satisfy the condition $\sum_{j=1}^{k} l_j \leq l$. The dissimilarity measure, which we denote by $\mathsf{d}(P_1, \ldots, P_k; P)$, is the square root of the sum of quadratic similarities f_j, minimized over all valid placements of $P_1, \ldots, P_k$ over P (or in other words, minimized over all valid horizontal and vertical shifts of their turning functions): $\mathsf{d}(P_1, \ldots, P_k; P) =$

$$\min_{\substack{\text{valid placements} \\ (t_1, \theta_1) \ldots (t_k, \theta_k)}} \left(\sum_{j=1}^{k} \mathsf{f}_j(t_j, \theta_j) \right)^{1/2}.$$

It remains to define what the valid placements are. The horizontal shifts $t_1, \ldots, t_k$ correspond to shiftings of the starting points of the polylines $P_1, \ldots, P_k$ along P. These horizontal shifts cannot be independent of each other, due to the required validity of the match and the ordering condition. The validity of the match is the condition that all k polylines should cover pieces of P that are mutually disjoint except possibly at their endpoints. The ordering condition implies that the starting points of $P_1, \ldots, P_k$ are matched with points on the boundary of P in counterclockwise order around P, that is: $t_{j-1} \leq t_j$ for all $1 < j \leq k$, and $t_k \leq t_1 + l$. Furthermore, the validity of the match implies a sharpening of the constraints to $t_{j-1} + l_{j-1} \leq t_j$ for all $1 < j \leq k$, and $t_k + l_k \leq t_1 + l$ (see figure 2). Without loss of generality, we may restrict the possible choices for t_1 to $[0, l)$. Thus, $t_2, ..., t_k$ must lie in a subinterval of $[0, 2l)$. The vertical shifts $\theta_1, \ldots, \theta_k$ correspond to rotations of the polylines $P_1, \ldots, P_k$ with respect to the reference orientation, and are independent of each other.

The dissimilarity measure between the polylines $P_1, \ldots, P_k$ and

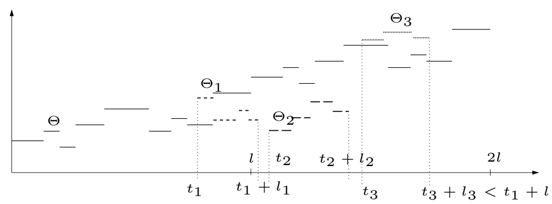

Figure 2: In order to measure the degree of matching $\mathsf{d}(P_1, \ldots, P_k; P)$ between the polylines P_1, P_2, P_3 and the polygon P, we shift the turning functions Θ_1, Θ_2, and Θ_3 horizontally and vertically over the turning function Θ.

the polygon P is thus given by: $\mathsf{d}(P_1, \ldots, P_k; P) =$

$$\min_{\substack{t_1 \in [0,l),\, t_2,\ldots,t_k \in [0,2l);\\ \forall j \in \{2,\ldots,k\}:\ t_{j-1}+l_{j-1} \le t_j; \quad t_k + l_k \le t_1 + l}} \left(\sum_{j=1}^{k} \mathsf{f}_j^*(t_j) \right)^{1/2}, \tag{1}$$

where $\mathsf{f}_j^*(t_j) = \min_{\theta \in \mathbb{R}} \mathsf{f}_j(t_j, \theta)$ is the quadratic dissimilarity between P_j and P for a given positioning t_j of the starting point of P_j over P, minimized over all rotations of P_j.

The properties of $\mathsf{d}(P_1, \ldots, P_k; P)$ and the characterization of the optimal solution lead to a straightforward dynamic programming algorithm that runs in $O(km^2n^2)$ time using $O(km^2n^2)$ storage. We also have developed a more efficient algorithm that runs in $O(kmn \log mn)$ time and uses $O(kmn \log mn)$ storage [14].

3. PART-BASED SHAPE RETRIEVAL

In order to demonstrate the effectivenes of the new dissimilarity measure for part-based shape retrieval, we have developed an experimentation system that uses the dissimilarity measure introduced in the previous section. The retrieval problem we are considering is the following: given a large collection of polygonal shapes, and a query consisting of a set of polylines, we want to retrieve those shapes in the collection that best match the query. The query represents a set of disjoint boundary parts of a single shape, and the matching process evaluates how closely these parts resemble pieces of a shape in the collection. Thus, instead of querying with complete shapes, we make the query process more flexible by allowing the user to search for only certain parts. The parts in the query are selected by the user from an automatically computed decomposition of a given contour.

The part-based retrieval works for any decomposition method, but in our implementation of the system, we have used a decomposition based on the medial axis [13].

3.1 Experimental Results

As test collection for the retrieval application we used the MPEG-7 shape silhouette database. We have used the Core Experiment “CE-Shape-1” part B [6], a test set devised by the MPEG-7 group to measure the performance of dissimilarity-based retrieval for shape descriptors. This test set consists of 1400 images: 70 shape classes of 20 images. The outer closed contour of the object in each image was extracted. In this contour, each pixel corresponds to a vertex. In order to decrease the number of vertices, we then used the Douglas-Peucker [3] polygon approximation algorithm. This also alleviates the potential problem of noise.

Each simplified contour was then decomposed into parts. The polygonal decomposition has proven to be robust against the level of polygon approximation. A smaller number of verties in the approximation has little effect on the simplified medial axis, and thus on the polygon decomposition, see [13].

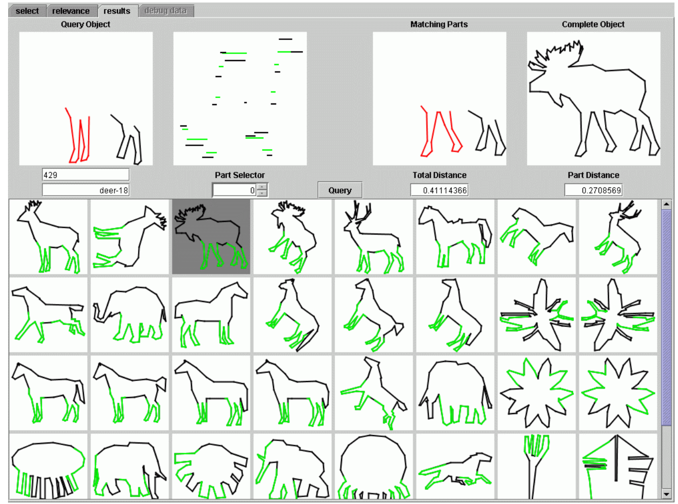

Figure 3: The retrieved results interface of our part-based shape retrieval application.

The matching of a query consisting of k polylines to an arbitrary database contour is based on the dissimilarity measure described in section 2. This dissimilarity measure is not scale invariant, as we noticed in section 2.1. The MPEG-7 shape collection, however, contains shapes at different scales. In order to achieve robustness to scaling, we scaled all shapes in the collection to the same diameter of the circumscribed disk. The reason we opted for a normalization based on the circumscribed disk, instead of the bounding box, for example, is that a class in the collection may contain images of an object at different rotational angles.

In order to formulate a query, the user selects an arbitrary shape in the collection, and then a set of parts from its decomposition. The selected parts can be treated either as separate chains of the query, or as adjacent concatenated parts. Figure 3 depicts the results interface of our experimentation platform. The best 40 matches are shown to the user, who is allowed to select any retrieved shape in order to find out detail information about the matching process. The query parts are depicted in the upper left side of the screen, in red. The selected retrieved shape appears in the upper right side, and its pieces matched by the query appear to its left. Through the dialog box “Part Selector”, the user can select a query part. For any retrieved shape, and any query part the system visualizes the turning functions of the query part and the piece of the retrieved shaped that is matched to. The quadratic dissimilarity between these polylines appears in the dialog box “Part Distance”, while the overall dissimilarity between the query and the selected retrieved shape appears in the dialog box “Total Distance”.

The selection of parts comprising the query has a big influence on the results of part-based retrieval. The MPEG-7 collection includes classes “horse”, “dog”, “deer”, “cattle”, whose shapes have similar parts, such as limbs. Querying with such parts retrieves shapes from all these classes, see for example figure 3. Though only a few shapes from the same class are ranked in the first 20 retrieved images, the part-based matching results cannot be regarded as poor. If we want to retrieve as many shapes from the same class, the selection of parts should capture more relevant and specific characteristics of the shape.

The MPEG-7 Visual Standard is initiated in order to specify standard content-based descriptors that allow to measure dissimilarity

image	Bull's Eye Performance (%)			True Positives in class size (%)		
	CSS	GTA	PBR	CSS	GTA	PBR
beetle-10	10	35	60	5	20	55
beetle-20	10	30	65	10	10	55
butterfly-4	15	25	55	10	20	55
butterfly-11	20	45	65	20	35	50
bird-9	20	25	50	15	25	40
bird-11	5	20	45	5	15	35
bird-17	20	15	60	20	15	40
carriage-18	70	80	95	45	80	90
crown-13	30	30	50	25	25	40
deer-13	20	40	45	15	40	40
deer-15	15	40	50	5	30	35
dog-11	10	15	50	10	15	45
horse-3	15	25	65	10	25	40
horse-4	25	30	50	20	20	45
horse-17	10	10	70	10	5	60
ray-3	15	15	70	15	5	60
ray-17	15	25	50	15	20	40

Figure 4: Experiment results. A comparison of the Curvature Scale Space (CSS), the global matching based on the turning function (GTA), and our Part-based Retrieval (PBR).

in images or video based on visual criteria. Each visual descriptor incorporated in the MPEG-7 Standard was selected from a few competing proposals, based on an evaluation of their performance in a series of tests called Core Experiments. The Core Experiment "CE-Shape-1" was devised to measure the performance of 2D shape descriptors. The performance of several shape descriptors, proposed for standardization within MPEG-7, is reported in [10]. The performance of each shape descriptor was measured using the so-called "bull's-eye performance": each image is used as a query, and the number of retrieved images belonging to the same class was counted in the top 40 (twice the class size) matches.

The shape descriptor selected by MPEG-7 to represent a closed contour of a 2D object in an image is based on the Curvature Scale Space (CSS) representation. The reported dissimilarity-based retrieval performance of the CSS described in [9] is 75.44%.

The dissimilarity measure used by our retrieval application uses a turning function representation. A whole contour turning function-based shape descriptor [5] is reported, in the proposal made to MPEG-7, to have a dissimilarity-based retrieval performance of 54.14%.

For easy classes in "CE-Shape-1", with a low variance among their shapes, the CSS matching [9] gives good results, with a retrieval rate over 90%, as measured by the "bull's-eye performance". For the class labeled "beetle", however, the different relative lengths of the antennas/legs, and the different shapes of the body pose problems for global retrieval. The average performance rate of CSS matching for this class is only 36%. For the "ray" and "deer" classes, the bad results (an average performance rate of the CSS matching of 26% and 33%, respectively) are caused by the different shape and size of the tails and antlers, respectively, of the contours in these classes. A part-based matching with a proper selection of parts, is significantly more effective in such cases.

We tested the performance of our part-based shape matching. An overall performance percentage for the matching process, like in [10], however, is untractable, since that would require 1400 interactive queries. We therefore present comparison results on a number of individual queries. We compared our approach with the global CSS matching and with a global matching based on the turning function. Figure 4 presents a set of instances when a part-based approach outperforms these global matching methods.

4. CONCLUDING REMARKS

We introduced a new measure for computing the dissimilarity between a set of parts of one shape and another shape. Its effectiveness was demonstrated in a part-based retrieval system. A prerequisite of effectiveness of the part-based matching is the selection of query parts that capture relevant and specific characteristics of the shape. This has to be done interactively by the user. Note that with only slight adaptations we can also solve the problem of matching an ordered set of polylines against an open polyline.

This paper focusses on the dissimilarity measure itself, not on a fully-fledged retrieval system. We have shown, however, that the use of a robust boundary decomposition method allows effective application of our new dissimilarity measure. Experimental results indicate that for those classes with a low average performance of the CSS matching, our approach consistently performs better.

It is difficult to compare our multiple polyline to polygon matching with other part-based matching methods, because they either work on feature vectors, on points, or on a single polyline, and each method requires its own very specific interactive part selection.

The algorithm to compute the dissimilarity measure was implemented in C++, the part-based retrieval interface was written in Java. The running time for a single query on the MPEG-7 test set of 1400 images is typically about one second on a 2 GHz PC.

Acknowledgment

We thank Geert-Jan Giezeman for programming support. This research was supported by the FP6 IST projects 511572-2 PROFI and 506766 AIM@SHAPE, and the Dutch Science Foundation (NWO) project 612.061.006 MINDSHADE.

References

[1] E. Arkin, L. Chew, D. Huttenlocher, K. Kedem, and J. Mitchell. An efficiently computable metric for comparing polygonal shapes. *PAMI*, 13:209–215, 1991.

[2] Scott D. Cohen and Leonidas J. Guibas. Partial matching of planar polylines under similarity transformations. In *Proc. SODA*, pages 777–786, 1997.

[3] D.H. Douglas and T.K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[4] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *PAMI*, 15:850–863, 1993.

[5] IBM. Technical summary of turning angle shape descriptors proposed by IBM. Technical report, IBM, 1999. TR ISO/IEC JTC 1/SC 29/WG 11/ P162.

[6] S. Jeannin and M. Bober. Description of core experiments for MPEG-7 motion/shape. Technical Report ISO/IEC JTC 1/SC 29/WG 11 MPEG99/N2690, March 1999.

[7] L. J. Latecki, R. Lakämper, and D. Wolter. Shape similarity and visual parts. In *Proceedings of the International Conference on Discrete Geometry for Computer Imagery (DGCI)*, pages 34–51, 2003.

[8] H.-C. Liu and M. D. Srinath. Partial shape classification using contour matching in distance transformation. *PAMI*, 12(11):1072–1079, 1990.

[9] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Workshop on Image DataBases and MultiMedia Search*, pages 35–42, 1996.

[10] J.-R. Ohm and K. Müller. Results of MPEG-7 Core Experiment Shape-1. Technical Report ISO/IEC JTC1/SC29/WG11 MPEG98/M4740, July 1999.

[11] E. Ozcan and C. K. Mohan. Partial shape matching using genetic algorithms. *Pattern Recognition Letters*, 18(10):987–992, 1997.

[12] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, and S.W. Zucker. Shock graphs and shape matching. *IJCV*, 55(1):13–32, 1999.

[13] Mirela Tanase. *Shape Decomposition and Retrieval*. PhD thesis, Utrecht University, Department of Computer Science, 2005.

[14] Mirela Tanase, Remco C. Veltkamp, and Herman Haverkort. Multiple polyline to polygon matching. Utrecht University TR UU-CS-2005-017, `http://ftp.cs.uu.nl/pub/RUU/CS/techreps/`. Accepted for ISAAC05.

[15] Haim Wolfson and Isidore Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science & Engineering*, pages 10–21, October-December 1997.