PART-BASED SHAPE RETRIEVAL WITH RELEVANCE FEEDBACK

Mirela Tănase   *Remco C. Veltkamp*

Department of Information and Computing Sciences
Utrecht University, The Netherlands

ABSTRACT

We introduce a relevance feedback mechanism for part-based shape retrieval. The database polygons are decomposed but the query polygon is not. In an initial search in the database, the database polygon parts are matched against the query polygon. The best matches are shown to the user, who has to decide which are relevant to its query. In successive iterations, the system infers which parts of the query are of interest, and makes a search with those parts only.

1. INTRODUCTION

The retrieval problem we consider is the following: given a large collection of images of polygonal shapes and a query polygon, we want to retrieve those shapes in the collection that "share" some parts with the query polygon. A possible approach is to formulate a query by selecting a part, or a combination of parts, from a polygon. The query must then be tested for similarity against the polygons in the database, and the best results are presented to the user. The retrieval performance depends on the selection of parts comprising a query, and the query process itself is labour intensive.

A natural extension to this type of retrieval is to relieve the user from the identifying the parts with good discriminative power. One way to do this is to divide the query polygon into parts, and to match all these parts and the possible combinations against the database polygons. An alternative is to decompose the database polygons and to match their parts against the query polygon. The larger the number of parts in the polygon decomposition, the smaller the number of similarities that remain undetected (false negatives) in the retrieval process. But since the number of possible combinations of parts of a polygon is exponential in the number of parts, a larger number of parts in the query or database polygon increases the query response time. Moreover, not all parts have the same discriminative power, and some combinations of parts may be irrelevant, thus it would be undesirable to search the database with them.

To avoid these problems, we propose a relevance feedback mechanism for part-based shape retrieval. In our approach, the database polygons are decomposed but the query polygon is not. In an initial search in the database, the database polygon parts are matched against the query polygon. The best matches are shown to the user, who has to decide which are relevant to its query. In successive iterations, the system tries to infer from the user's feedback which parts of the query are of interest, and makes a search with those parts only.

1.1. Previous Work

Four main techniques can be distinguished for adjusting the search.

i) Refining the query. This technique is directly inspired by Information Retrieval and consists of computing at each iteration step a "mean query" from the relevant and irrelevant examples provided by the user, typically by Rocchio formula [1].

ii) Refining the similarity measure[1]. When multiple features are used, the purpose of the relevance feedback mechanism is often to optimize the ratios of combining various feature similarities into an overall similarity. Usually a weighted linear combination of feature similarities is used, by using the ranks of the relevant images in the retrieval [2], by using the scores in a 5-level grading of a set of N most similar images [3], or by minimizing a quadratic error [4].

iii) Changing the feature space. Rather than just showing to the user a few images that match the query, [5] propose to give information about the status of the whole database. Thumbnail images are shown to the user in such a way that their relative location on the screen reflects the similarities between them. The user interacts with the system by directly manipulating the thumbnails on the screen. Based on the user's feedback the system reorganizes the entire database.

iv) Changing a predicting relevance. A retrieval system that uses a relevance feedback mechanism based on adjusting a predicted relevance is the PicHunter system [6]. A vector is used for retaining each image's probability of being the target. This vector is updated at each iteration of the relevance feedback, based on the history of the session (images displayed by the system and user's actions in previous

This research was supported by FP6 IST Network of Excellence 506766 AIM@SHAPE.

[1] Following common practice, 'similarity measure' is often used instead of dissimilarity.

0-7803-9332-5/05/$20.00 ©2005 IEEE

iterations). The updating formula is based on the Bayes' rule.

2. RF FOR PART-BASED RETRIEVAL

The boundary decomposition we used is the instantiation of the decomposition framework using the medial axis [7].

2.1. First retrieval iteration

A first, rough search in the database is done by comparing the parts of each database polygon with the query polygon. Let s denote a similarity measure between a polyline and a polygon, which measures how closely the polyline is to being part of the polygon. If B is a polyline, and A is a polygon, let A_{s}^{B} denote the portion of A which matches B through similarity s.

A similarity measure between a polyline and a polygon is the measure based on the turning functions of the two shapes [8]. If Θ_A is the turning function of the query polygon A, of length l_A, and Θ_B is the turning function of polyline B of length l_B, the similarity between the polyline B and the query polygon A is given by: $\mathsf{s}'(A,B) =$

$$\min_{\theta\in\mathbb{R},t\in[0,l_A)} \left(\int_0^{l_B} (\Theta_A(s+t) - \Theta_B(s) + \theta)^2 ds \right)^{1/2} .$$

Then $A_{\mathsf{s}'}^{B}$ is the chain of A starting at $A(t^*)$ and ending at $A(t^* + l_B)$, where t^* is the optimal value of t in the minimization problem defining $\mathsf{s}'(A,B)$, and $A(t)$ is the point of A at distance t along A from the reference point $A(0)$ of the turning function Θ_A. We use a variant of this similarity measure: $\mathsf{s}(A,B) = \mathsf{s}'(A,B)/\sqrt{l_B}$, which has the advantage that it usually ranks longer polylines higher.

Let A be a query polygon, and P be an arbitrary database polygon. As we already mentioned, the boundary of each database polygon is decomposed, before the retrieval process takes place, using any boundary decomposition method. Let $P_1, \ldots, P_m$ denote the boundary parts of the database polygon P. In the first iteration of the proposed relevance feedback mechanism, each database polygon part is matched against the query polygon A. Thus for each database polygon P, with m constituent boundary parts $P_1, \ldots, P_m$, we compute the set of similarities $\mathsf{s}(A, P_i)$, $i = 1, \ldots, m$. We also identify, for each $i \in \{1, \ldots, m\}$, the portion $A_{\mathsf{s}}^{P_i}$ of the query polygon A that best matches P_i through similarity s. Notice that, since we match the parts of P individually, we could have $A_{\mathsf{s}}^{P_i} \cap A_{\mathsf{s}}^{P_j} \neq \emptyset$.

The order in which polygons from the database are retrieved to the user is based on the set of similarities $\{\mathsf{s}(A, P_i)\}$ computed for each database polygon. Intuitively, since we are doing partial matching, the retrieval ordering should be affected mostly by the small values in the sets $\{\mathsf{s}(A, P_i)\}$. That is, if a database polygon P contains a number of parts that match well the query polygon, and also parts that give a very large value for the similarity measure s, the rank of P in the retrieval should be considerably less affected by the similarities of the later parts than by those of the former parts. Rankings based on a weighted linear combination of these similarities, or a thresholding of these similarities followed by an equally weighted linear combination, require parameter tuning prior to the retrieval process. To avoid this, we propose a simple ranking process that is based on a lexicographical order of the similarities in $\{\mathsf{s}(A, P_i)\}$. If $\{\mathsf{s}(A, P_1), \ldots, \mathsf{s}(A, P_m)\}$ is the set of similarities between the boundary parts of a database polygon P and the query A, we denote by S_P the m-dimensional vector $\mathsf{S}_P = (\mathsf{s}(A, P_{i_1}), \ldots, \mathsf{s}(A, P_{i_m}))$, with $\mathsf{s}(A, P_{i_1}) \leq \mathsf{s}(A, P_{i_2}) \leq \mathsf{s}(A, P_{i_m})$. In other words, S_P contains the sorted similarities between the boundary parts of P and A, with the smallest similarity as the first component and the largest as the last component. We call S_P, the *partial similarity vector* of P. Given two database polygons P and Q, their relative order in the retrieval ranking is given by the lexicographic order of the partial similarity vectors S_P and S_Q. The database polygons are thus presented to the user in the increasing lexicographical order of their partial similarity vectors.

2.2. Subsequent iterations

Using the retrieved results marked by the user as relevant, the system re-evaluates the query by identifying a set of boundary parts of the query polygon, that seem to be of interest to the user. A new search in the database will then be made with this set of parts.

The computation of these boundary parts of the query polygon is based on computing a *weight function* that maps each boundary point of the query to a real value (its *weight*), and then retaining only those points with a weight smaller than a threshold. The weight function on the points in A is intended to allow the system to infer from the user's feedback which are the parts of the query that the user would be interested in searching the database. The lower the weight of a point x, the larger the possibility that x is contained in a part of interest to the user. The computation of the weight function and the corresponding threshold is based on the partial similarity vectors of the polygons marked by the user as relevant.

We now describe the computation of the weight function $\omega : A \to \mathbb{R}$, and the threshold t_ω. A point x of the query polygon is assumed to be contained in a part of interest if there are parts in the relevant polygons that, when matched to the query polygon, "cover" x. In other words, x is contained in a piece of the query polygon matched by a part of a relevant polygon. For such points, their weight is based on similarities of the parts in the relevant polygons that "covers" them. The larger the resemblance of a part in a relevant polygon with a piece of the query polygon containing x, the smaller the weight of x.

More precisely, let $R^1, \ldots, R^n$ be the retrieved polygons marked by the user as relevant, and let $R^k_1, \ldots, R^k_{m_k}$, denote the parts in the decomposition of R^k, $k = 1, \ldots, n$. The weight function ω corresponding to the set $\{R^1, \ldots, R^n\}$ of relevant results, is given by :

$$\omega(x) = \min_{\substack{k=1,\ldots,n;\ j=1,\ldots,m_k \\ \text{such that } x \in A_{\mathsf{s}}^{R^k_j}}} \mathsf{s}(A, R^k_j)$$

and $\omega(x) = \max_i \max_P \mathsf{S}_P[i]$, if $\forall\, k \in \{1, \ldots, n\}$, $\forall\, j \in \{1, \ldots, m_k\}\ x \notin A_{\mathsf{s}}^{R^k_j}$. So, the weight of each query point is given by the smallest similarity measure of a part in a relevant polygon that "covers" x in its optimal match against A. When there is no part of a relevant polygon to "cover" x, the weight of x is equal to the largest component of a partial similarity vector across the whole database.

A new search in the database is made with those connected polylines in A made of points with a weight smaller than a threshold t_ω. The weight of a point in A is determined by the similarities of the parts in the relevant polygons "covering" x in their optimal match against A. Some parts of the relevant polygons may not match well the query polygon, and thus the user may not be interested in points of A covered by such parts. For this reason the threshold value we consider is based only on the smallest similarities in the partial similarity vectors of the relevant polygons. The threshold t_ω is given by:

$$t_\omega = \max_{i=1,\ldots,n} \mathsf{S}_{R^i}[1] + \alpha(\max_{i=1,\ldots,n} \mathsf{S}_{R^i}[1] - \min_{i=1,\ldots,n} \mathsf{S}_{R^i}[1]).$$

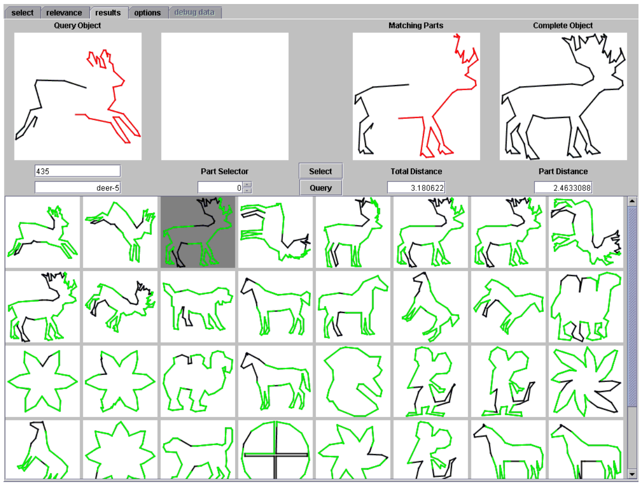
The parameter α in the threshold t_ω is strictly positive. As α increases, the length of the polylines of A made of points with a weight smaller than t_ω increases also.

After computing the set $A_1, \ldots, A_k$ of disjoint polylines of A consisting of points of weight smaller than the threshold t_ω, the system searches the database with the union of these parts. We use for the matching of a union of disjoint polylines with a polygon the similarity measure described in [9]. This matching is computationaly more complex than the straightforward matching that is done in the first iteration.

3. EXPERIMENTAL SETUP

We tested our relevance feedback mechanism on the MPEG7 Core Experiment "CE-Shape1" test set. This database consists of images of white objects on a black background. We computed an approximation of the outer closed contour of the object in each image, together with a boundary decomposition of the resulting simplified contour.

Figure 1 shows our retrieval application. The computed query parts are depicted in the upper left side of the screen.



435

deer-5

Part Selector: 0

Total Distance: 3.180622

Part Distance: 2.4633088

Fig. 1. The query parts computed from iteration 2 onwards and the results of searching the database with these parts are shown in the results screen.

The user can select one of the retrieved shapes (by clicking on it) and the selected shape appears in the upper right side, and its pieces matched by the query parts appear to its left.

In order to evaluate the performance of the above feedback mechanism, we made queries in the "CE-Shape1" test set. For each query polygon, there are many ways in which the user can guide a search in the database. Different selections of relevant contours and different threshold parameters lead to very different results. We chose, however, to query the database in a consistent way for all shapes in our test set, despite the fact that other modalities of searching may produce better results.

We considered the following experimental setup. Each contour in this small test set was submitted as a query. From the retrieved results, we counted the number of contours belonging to the same class that were ranked in the top 40 matches ("Bull's Eye" test).

For the purpose of experimental evaluation, the marking of relevant results after the first iteration is done as follows. If the number of contours belonging to the same class ranked in the top 40 is:

- only one, then we mark as relevant the first ten, after the query, in the top 80 matches;
- smaller than eleven, we mark all these (except the query) as relevant;
- larger than eleven, we mark the first ten ranked after the query.

We thus mark at most ten contours in each situation. Marking too many contours may lead to the whole query polygon boundary to be selected for researching the database, which means a query with the whole polygon and this is not what we are interested in.

We then re-iterate the search and compute the retrieval performance of the second iteration in the same way using the Bull's Eye test. The retrieval performance in any successive iteration is given by best performance from two queries:

- a query with the parts resulting from marking as relevant all contours (except the query) belonging to the same class and ranked in the top 20 matches;
- a query with the parts resulting from marking as relevant all contours belonging to the same class and ranked from the 10th to the 40th match.

This last way of querying was added to circumvent the situation that the first way leads to querying with parts that covers the whole contour. That would lead to a bad retrieval performance, since that amounts to global contour matching. For all query polygons and all iterations, the value of the threshold parameter α was set to 1.

4. RESULTS

We compared our part-based relevance feedback approach with a global shape matching technique, based on a curvature scale space representation (CSS) of the shape, which has one of the best reported retrieval rates on this test set, and was therefore selected to be part of the MPEG7 definition. We have selected a number of input images for which the CSS has a bad performance, to see if our relevance feedbakck provides additional retrieval power, see the table in figure 2. The table presents the results of the testing described in section 3 for three successive iterations, given by the Bull's Eye percentage. We found that from the fourth iteration onwards, evaluating the performance in a consistent way as in the previous iterations is not meaningful, because it often amounts to matching the whole contour.

With a few exceptions, the query results improve from one iteration to the other. When this is not the case, it is causes by two effects. One is that the query parts are very small, the other is that the query part is too big, for example the whole contour. These two effects are caused by the standard way of querying as described in section 3, and can be alleviated by fine tuning the selection of the relevant contours and/or the threshold parameter α. Indeed, this leads to better results in all those cases from figure 2.

image	It 1	It 2	It 3	CSS
beetle-10	40%	50%	45%	10%
beetle-20	30%	20%	20%	10%
butterfly-4	25%	45%	55%	15%
butterfly-11	30%	75%	75%	20%
bird-9	10%	35%	50%	20%
bird-11	25%	25%	20%	5%
bird-17	20%	40%	65%	20%
carriage-18	65%	95%	95%	70%
crown-13	10%	40%	30%	30%
deer-13	15%	40%	40%	20%
deer-15	10%	20%	50%	15%
dog-11	5%	30%	20%	10%
horse-3	10%	35%	45%	15%
horse-4	10%	35%	50%	25%
horse-17	10%	50%	85%	15%
ray-3	15%	50%	65%	15%
ray-17	10%	55%	70%	15%

Fig. 2. Experiment results. Bull's Eye percentages for the first three iterations, as well as for the Curavature Scale Space method.

References

[1] J. J. Rocchio, "Relevance feedback in information retrieval," in *G. Salton, editor The Smart System - Experiments in Automatic Document Processing*, 1971, pp. 313–323.

[2] T.-S. Chua, C.-X. Chu, and M. S. Kankanhalli, "Relevance feedback techniques for image retrieval using multiple attributes," in *Proceeding of the IEEE International Conference on Multimedia Computing and Systems, ICMCS 1999, 7-11 June, 1999, Florence, Italy, Proceedings, Volume I*, 1999, pp. 890–894.

[3] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool in interactive content-based image retrieval," *IEEE Transaction on Circuits and Systems for Video Technology, Special Issue on Segmentation, Description, and Retrieval of Video Content*, vol. 8, no. 5, pp. 644–655, 1998.

[4] S. Philipp-Foliguet J. Fournier, M. Cord, "RETIN: A content-based image indexing and retrieval system," *Pattern Analysis and Applications*, vol. 4, no. 2, pp. 153–173, 2001.

[5] S. Santini and R. Jain, "Beyond query by example," in *Proceedings of the 6th ACM International Conference on Multimedia '98, September 12-16, 1998, Bristol, England*, 1998, pp. 345–350.

[6] I. Cox, M. Miller, T. Minka, T. Papathomas, and P. Yianilos, "The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 20–37, January 2000.

[7] M. Tănase, *Shape Decomposition and Retrieval*, Ph.D. thesis, Utrecht University, Institution of Information and Computing Sciences, 2005.

[8] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S.B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209–215, March 1991.

[9] M. Tănase, R. C. Veltkamp, and Herman Haverkort, "Multiple polyline to polygon matching," Tech. Rep. UU-CS-2005-017, Utrecht University, Institution of Information and Computing Sciences, 2005.