

SHAPE MATCHING WITH OCCLUSION IN IMAGE DATABASES

Aristeidis Diplaros¹

Intelligent Sensory Information Systems, Faculty of Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
diplaros@science.uva.nl

Euripides G.M. Petrakis²

Department of Electronic and Computer Engineering, Technical University of Crete
GR-73100 Chania, Crete, Greece
petrakis@ced.tuc.gr

Evangelos Milios³

Department of Computer Science, Dalhousie University
Halifax, Nova Scotia, Canada B3H 1W5
eem@cs.dal.ca

Abstract. We propose an approach for matching deformed, occluded and unoccluded shapes using Dynamic Programming (DP). We distinguish among various cases of matching including cases where the shapes are scaled with respect to each other or cases where one shape matches the whole or only a part of the other shape. Our approach handles noise and shape distortions by allowing matching of merged sequences of consecutive small segments in a shape, with larger segments of another shape, while being invariant to translation, scale orientation and starting point selection. We demonstrate the superiority of our approach over traditional approaches to shape matching and retrieval based on Fourier descriptors and moments.

1 Introduction

The increasing amounts of image data in many application domains has generated additional interest for real-time image management and retrieval by shape content. Regarding shape matching (the core problem in shape retrieval) multiscale methods are considered the most promising. In [9], matching is performed through interval trees which are computed by tracking the *Curvature Scale Space* (CSS) representation from coarser to finer scales. In [8], the matching mechanism of the SQUID⁴ system, only the maxima of the CSS curves are used. In [11], it is demonstrated, that small shape changes may cause major structural changes in the interval tree and this may lead to matching errors. Multiscale methods have also been combined with Dynamic Programming (DP) [11, 7].

The methods referred to above do not treat occluded shapes. A promising method but untested in shape retrieval, defines a shape distance and association of shape parts on the basis of area features (shape skeleton) [10] by reducing shape matching to a largest subgraph isomorphism problem. A recent contribution [1], the same as our approach, it is motivated by an effort to avoid the high computational complexity of the true CSS-based approaches and also han-

¹This work is part of the author's student dissertation at the Technical University of Crete.

²This work was supported by project HIPER (BE97-5084) under programme BRIGHT-EURAM of the European Union (EU).

³This work was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

⁴<http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html>.

dles occlusion. However, this method, the same as [10], has never been tested on large image collection nor they are intrinsically invariant to scale and rotation.

We propose an approach for shape matching based on DP. We distinguish among various cases of matching including cases where one shape matches the whole or only a part of the second shape, cases where one or both shapes are occluded and cases where the two shapes are given at different scales (the matching algorithm determines the appropriate scale factor for matching). Matching is independent of shape translation, scaling, rotation and starting point selection. It operates implicitly at multiple scales by allowing the matching of merged sequences of consecutive segments in the shapes which are matched. This way our method maintains the advantages of previous methods (e.g., [11, 8]) utilizing smoothed versions of the shapes at various levels of detail, while avoiding the expensive computation of explicit scale-space representations. Our previous work on DP matching [7], handles only closed shapes and, the same as [11], it is non-optimal, that is, it may fail to compute the optimal match. We demonstrate the superiority of our approach over traditional methods for shape matching and retrieval such as Fourier descriptors [12, 4] and moments [3, 2, 5].

The basic notation, the cost functions and the outline of our shape matching algorithm are described in Section 2. Section 3 presents experimental results followed by conclusions in Section 4.

2 Methodology

In matching two shapes A and B , the algorithm builds a DP table, where rows and columns correspond to inflection points of A and B respectively. Starting at the lower left corner and proceeding upwards and to the right, the table is filled with the cost of the partial match containing the segments between the inflection points (rows and columns) swept so far. Because convex segments cannot match concave ones [11], only about half the cells are assigned cost values, in a checkerboard pattern. Merges, where a segment sequence of one shape matches a single segment of the other shape can occur. Merges introduce jumps in the traversal of the DP table. Reaching the top row implies a complete match, where all inflection points of shape A have been swept. Additional information is stored in each cell to allow the tracing of a path starting from that cell and working backwards. The tracing of a path reveals segment associations between the two shapes.

We distinguish between the following two cases of matching namely *global matching* and *local matching*. In global matching, the algorithm will find the best mapping between segments of A and B so that, no segments remain unassociated in either shape. In local matching the algorithm will find the best association of all segments of A to a subsequence of segments of B and vice versa. Because we cannot know in advance which shape is included within the other one, we run the algorithm twice (i.e., once for each possibility) and we take the matching with the minimum cost.

2.1 Dynamic Programming (DP) table

Let $A = a_1, a_2, \dots, a_M$ and $B = b_1, b_2, \dots, b_N$ be the sequence of M and N convex (C) and concave (V) segments of the two shapes to be matched, with a_i being the segment between inflection points p_i and p_{i+1} and b_j the segment between inflection points q_j and q_{j+1} . Henceforth, $a(i - m|i)$, $m \geq 0$, denotes the sequence of segments $a_{i-m}, a_{i-m+1}, \dots, a_i$; similarly for $b(j - n|j)$, $n \geq 0$. If shape A (or B) is closed, then $p_1 = p_{M+1}$ (or $q_1 = q_{N+1}$).

The DP table has \mathcal{M} rows and \mathcal{N} columns. If both shapes are open then $\mathcal{M} = M + 1$ and $\mathcal{N} = N + 1$; if shape A is open and shape B is closed, then $\mathcal{M} = M + 1$ and $\mathcal{N} = 2N$ (i.e.,

5			T		T		T	Termination Area
4		X		X		X		
3			X		X		X	
2		X		X		X		
1	S		S		S		S	Initialization Area
	1	2	3	4	5	6	7	

Figure 1: Example of a DP table with $\mathcal{M} = 5$ (shape A) and $\mathcal{N} = 7$ (shape B). S , X and T denote cells in the initialization, computation and termination areas respectively.

shape B is repeated twice to force the algorithm consider all possible starting points on B). If shape A is closed and B is open, we switch the roles of A and B . The case where both shapes are closed reduces to the previous one (see Section 2.3).

The rows of a DP table are indexed by i , $1 \leq i \leq \mathcal{M}$ and its columns are indexed by j , $1 \leq j \leq \mathcal{N}$ where, i, j are indices to inflection points of A and B respectively. If shape B is closed, its indices are taken modulo N . The cell at the intersection of rows i and column j is referred to as $cell(i, j)$. A link between cells (i_{w-1}, j_{w-1}) and (i_w, j_w) denotes the matching of the merged sequence of segments $a(i_{w-1}|i_w)$ with $b(j_{w-1}|j_w)$. $cell(i_{w-1}, j_{w-1})$ is called *parent* of $cell(i_w, j_w)$. A *path* is a linked sequence of cells $((i_0, j_0), (i_1, j_1), \dots, (i_t, j_t))$, not necessarily adjacent, indicating a partial match, where $i_0 < i_1 < \dots < i_t$ and $j_0 < j_1 < \dots < j_t$. This path begins at inflection point p_{i_0} of shape A and at inflection point q_{j_0} of shape B and tries to match sequences of segments $a(i_{w-1}|i_w)$ of A with sequences $b(j_{w-1}|j_w)$ of B for $w = 1, 2, \dots, t$.

Each $cell(i_w, j_w)$ contains the following values: $g(i_w, j_w)$, i_{w-1} , j_{w-1} , u_w , v_w and ρ_w where $g(i_w, j_w)$ is the partially accumulated match cost up to that cell, u_w and v_w denote number of unmatched segments of A and B respectively, i_{w-1} and j_{w-1} are the indices of the parent cell of $cell(i_w, j_w)$ and are used to trace back a complete path. The tracing of a path reveals segment associations between the two shapes. Finally, ρ_w denotes the scale factor corresponding to the parts of A and B which have been matched up to $cell(i_w, j_w)$ and is defined in Section 2.2.

Fig. 1 illustrates an example of a DP table. The DP table consists of three distinct areas:

Initialization area: It is the first row of the DP table. All paths start from cells in this area.

Matching starts always at the first segment a_1 of A ($i_0 = 1$). Matching may start at any segment b_{j_0} of B , where $1 \leq j_0 \leq N$. If a_1 and b_{j_0} have the same polarity, then $g(1, j_0)$, i_{w-1} , j_{w-1} , u_w , v_w , ρ_w are $0, 0, 0, M, N, 1$ respectively; otherwise we set $g(1, j_0) = \infty$.

Computation area: It is the area between the first and last row of the DP table. Cells in this area correspond to incomplete paths.

Termination area: It is the last row of the DP table. All complete paths end at cells in this area. The best match corresponds to the path with the least cost. Notice that about half of the cells of the above DP table are empty; this is because associations between opposite type segments (i.e., C and V) are not allowed [11]. By convention, the cost of matching

C with V segments is infinite. Matching always starts at the first inflection point of A ($i_0 = 1$) while any point of B is a candidate starting point.

2.2 Distance function

Fig. 2 illustrates geometric quantities (features) required in the definition of the cost functions below. These are

Rotation angle θ_i that is the angle traversed by the tangent to the segment from inflection point p_i to inflection point p_{i+1} and shows how strongly a segment is curved,

Length l_i that is the length of segment a_i and, finally,

Area s_i that is the area enclosed between the chord and the arc between the inflection points p_i and p_{i+1} .

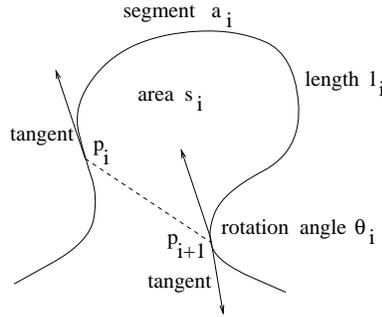


Figure 2: Geometric quantities for defining the importance of a segment

If one of the two shapes is scaled with respect to the other, then the length of one of the two shapes (i.e., shape B) has to be multiplied by an appropriate *scale factor*. If matching is global, shape A matches the whole shape B . The scale factor is computed as

$$\rho = \frac{\text{length of } A}{\text{length of } B}. \quad (1)$$

If matching is local, shape A may match either the whole or only a part of shape B . This case is more difficult to handle but, it is more general and includes the previous one (i.e., when matching the whole shape B yields the least cost). Although we know that the matched part of shape A will be the whole shape A , the length of the matched part of shape B is unknown before the algorithm is completed. To handle this problem, we compute a scale factor ρ_t for each partial path $((i_0, j_0), (i_1, j_1), \dots, (i_t, j_t))$, corresponding to matched parts so far:

$$\rho_t = \frac{\sum_{w=1}^t \sum_{i=i_{w-1}}^{i_w} l_i(A)}{\sum_{w=1}^t \sum_{j=j_{w-1}}^{j_w} l_j(B)}, \quad (2)$$

where $1 \leq t \leq W$ and $l_i(A)$ and $l_j(B)$ are the lengths of a_i and b_j respectively and W is total the number of matched segment sequences. This value is an approximation of the actual scale factor of a complete match. Notice that ρ_0 is undefined since the total matched length is 0 for both shapes. In this work ρ_0 is set to 1.

A *complete match* is a correspondence between sequences of segments in order, such that no segments are left unassociated in shape A and there are no crossovers or omissions. A complete

match is characterized by a *complete path* $((i_0, j_0), (i_1, j_1), \dots, (i_W, j_W))$, a path that starts at the initialization and ends at the termination area. The cost $D(A, B)$ of matching shape A with shape B is defined as the minimum complete match cost:

$$D(A, B) = \min_L D'(A, B), \quad (3)$$

where L is the number of complete paths found on the DP matching table and $D'(A, B)$ is the cost of a complete match. In turn, the cost of a complete match is defined as

$$D'(A, B) = \min_{i_w, j_w} \sum_{w=1}^W \psi(a(i_{w-1}|i_w), b(j_{w-1}|j_w)). \quad (4)$$

Function $\psi(a(i_{w-1}|i_w), b(j_{w-1}|j_w))$ represents the dissimilarity cost of its two arguments and is defined as

$$\psi(a(i_{w-1}|i_w), b(j_{w-1}|j_w)) = \lambda \text{MergingCost}(a(i_{w-1}|i_w)) + \lambda \text{MergingCost}(b(j_{w-1}|j_w)) + \text{DissimilarityCost}(a(i_{w-1}|i_w), b(j_{w-1}|j_w)). \quad (5)$$

The first two terms in Eq. 5 represent the cost of merging segments $a(i_{w-1}|i_w)$ in shape A and segments $b(j_{w-1}|j_w)$ in shape B respectively while the last term is the cost of associating the merged sequence $a(i_{w-1}|i_w)$ with the merged sequence $b(j_{w-1}|j_w)$. Constant λ represents the relative importance of the merging and dissimilarity costs. In this work λ was set to 1. Each allowable merging should be a recursive application of the grammar rules $CVC \Rightarrow C$ and $VCV \Rightarrow V$ [11]. This is enforced by the DP algorithm (see Fig. 3).

The dissimilarity cost of associating a group of segments from shape A with a group of segments from shape B is computed as

$$\text{DissimilarityCost} = U \max_{\text{all features } f} \{d_f\}. \quad (6)$$

The term d_f is the cost associated with the difference in feature f (i.e., length, area or angle). The intuition behind the use of *max* is that it tends to emphasize large differences on any feature. U is a weight term associated with the importance of this partial match. The proportion of the matched shape length with respect to total length is used to define U :

$$U = \max \left\{ \frac{\sum_{i=i_{w-1}}^{i_w} l_i(A)}{\text{length of } A}, \frac{\sum_{j=j_{w-1}}^{j_w} l_j(B)}{\text{length of } B} \right\}. \quad (7)$$

The term d_f is defined as

$$d_f = \frac{|F_A - S_w(f)F_B|}{F_A + S_w(f)F_B}, \quad (8)$$

where, $F_A = \sum_{i=i_{w-1}}^{i_w} |f_i|$, $F_B = \sum_{j=j_{w-1}}^{j_w} |f_j|$ and $S_w(f)$ is a parameter depending on the feature f . Specifically $S_w(f) = \rho_{w-1}$ for f being length and ρ_{w-1}^2 for f being area. For f being rotation angle, $S_w(f) = 1$, since angle measurements do not depend on the scale factor.

Let the types of the segments being merged be $CVC \dots C$ (the opposite case is obtained by switching C and V in the formulaes below). The merging cost is defined as follows:

$$\text{MergingCost} = \max_{\text{all features } f} \{U_f C_f\}, \quad (9)$$

where subscript f refers to a feature (length, area or rotation angle). For all features:

$$C_f = \frac{\sum_V \text{segs of group } |f|}{\sum_{\text{all segs of group}} |f|}. \quad (10)$$

```

Input: Shapes  $A = a_1, a_2, \dots, a_M, B = b_1, b_2, \dots, b_N$ ;
Output: Distance  $D(A, B)$  and correspondences between
similar sequences of segments;
// Initialization: Fill the first row
for  $j_0 = 1, 2, \dots, \mathcal{N}$  do
  if  $a_1$  and  $b_{j_0}$  are both  $C$  or  $V$  then  $cell(1, j_0) = (0, 0, 0, M, N, 1)$ ;
  otherwise  $cell(1, j_0) = (\infty, 0, 0, M, N, 1)$ ;
end for
// Fill from the 2nd to the  $\mathcal{M}$ -th row
for  $i_w = 2, 3, \dots, \mathcal{M}$  do
  for  $j_w = 2, 3, \dots, \mathcal{N}$  do
    if  $a_{i_w}$  and  $b_{j_w}$  are both  $C$  or  $V$  then
      compute  $g(i_w, j_w) = g(i_w - 1, j_w - 1) + \psi(a(i_w - 1 | i_w), b(j_w - 1 | j_w))$ ;
      compute  $\rho_w$  using Eq. 1 (global matching) or Eq. 2 (local matching);
    end for
  end for
// Select the least cost complete path
  select the least cost path from the  $\mathcal{M}$ -th row;
  retrace path using  $i_{w-1}, j_{w-1}$  cell values;

```

Figure 3: Outline of the algorithm.

The intuition behind these formulas is that they measure the importance of the absorbed segments (of type V) relative to the whole matched consecutive segments of the group. For f being any feature (length, area, rotation angle), U_f is defined as

$$U_f = \frac{\sum_V \text{segs of group } |f|}{\sum_V \text{segs of shape } |f|}. \quad (11)$$

where the sum of the numerator is over the absorbed concave segments, while the sum of the denominator is over all concave segments of the shape. The intuition behind this weight term is to measure the importance of the absorbed segments within the shape as a whole.

2.3 Algorithm

Let A and B be the two shapes to be matched. A is assumed to be open and B can be either open or closed. Fig. 3 outlines the matching algorithm. We handle the case with both shapes closed by pretending that A is open, repeating the algorithm for open and closed (global) shape matching M times (for each starting point on A), and by taking the least cost match as the cost of matching. We assume that the first segments of A and B have the same polarity (C or V); otherwise, matching starts at the second segment of B . Matching ends when all segments of A have been consumed. The last matched segments of A and B have the same polarity too. For global matching the algorithm consumes all segments from both shapes. Equivalently, the algorithm starts at the left-most cell and terminates at the right-most cell of the DP table. This cell contains the cost of matching. For local matching, any segment on shape B is a candidate starting segment for matching provided that it has the same polarity with the first segment of A . Equivalently, only half of the cells in the initialization area are candidate cells of starting path. The algorithm consumes all segments of A and may end at any segment of B having the same polarity with the last segment of A . Therefore, half of the cells at the termination area are

candidate termination cells of a complete match path. All these cells are searched to select the least cost match.

Fig. 4 illustrates segment correspondences (indicated by consecutive lines connecting the starting and ending points of the associated segments) obtained by matching fish silhouettes. Dots on shape curves denote inflection points. The left figure of the first row illustrates local matching between open shapes (i.e., part of the bigger shape has been left unmatched) and right figure of the same row illustrates global matching. The left figure of the second row illustrates local matching between an open and a closed curve while, the right figure corresponds to global matching between closed shapes.

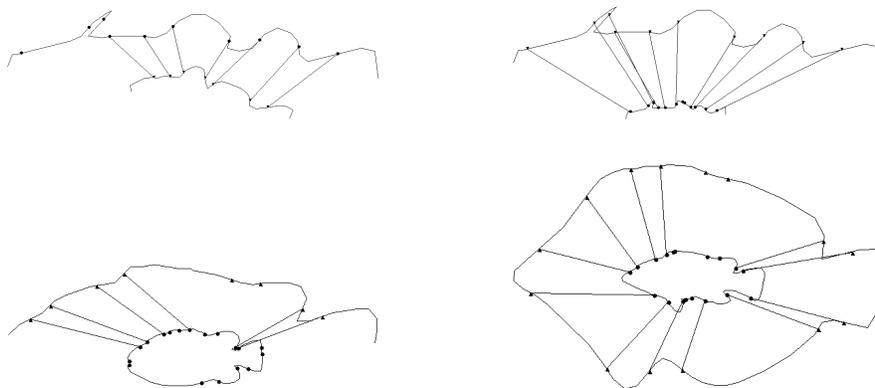


Figure 4: Segment associations reported by the matching algorithm.

3 Shape retrieval

In our experiments we used the following datasets:

CLOSED⁵: It is the dataset of SQUID and consists of 1,100 closed shapes.

OPEN⁶: Consists of 1.500 open shapes which have been generated from the CLOSED dataset by editing.

The experiments are designed to illustrate the superiority of our approach over traditional methods for shape matching and retrieval based on Fourier descriptors [12, 4] and moments [3, 2, 5], over our non-optimal DP shape matching method of [7] and SQUID. Notice that the last two methods work only of closed shapes. It is an interesting future project to have a thorough comparative experimental evaluation with shape matching methods like [1] and [10]. However, the underlying matching algorithms of these methods are quite complicated and their implementation is non-trivial. In [6] we present performance comparisons of our previous non-optimal DP-matching approach with the method of [11].

To evaluate our algorithm we created 18 closed and 20 open query shapes. The evaluations have been carried-out by 3 independent referees. Each open query on the OPEN dataset retrieved the its best 50 answers and each closed query on the CLOSED dataset retrieved its best 18 answers (the SQUID interface supports only 18 answers). Two shapes (open or closed) are

⁵<http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html>.

⁶<http://www.cs.yorku.ca/~eem/marineDB>.

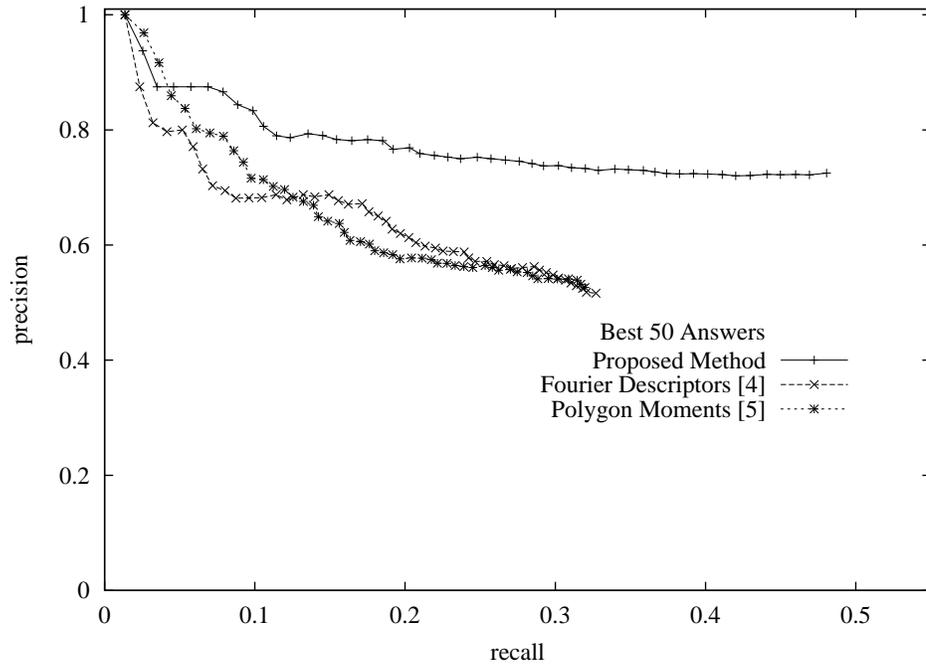


Figure 5: Precision-recall diagrams for the OPEN dataset.

considered similar if they represent the same figure. In particular, an open shape is considered similar to another open or closed shape if the former is similar (at least) to a part of the later.

We computed *precision* (i.e., percentage of similar shapes retrieved with respect to the number of retrieved shapes) and *recall* (i.e., percentage of qualifying shapes retrieved with respect to the total number of similar shapes in the database). We present a *precision-recall* plot for each method. The top-left point of a each plot corresponds to the precision/recall values for the best answer while, the bottom right point corresponds to the precision/recall values for the entire answer set. A method is better than another if achieves better precision and better recall.

Fig. 5 demonstrates that our method achieves at least 15% better precision and better recall for retrievals with the best 50 answers on the OPEN dataset. Fourier and moments perform about the same. For small answer sets containing up to 3 shapes (left-most 3 points of each curve) our method performs approximately the same with moments. Fig. 6 demonstrates that our method performs better than SQUID for large answer sets containing more than 4 answers, achieving up to 10% better precision and better recall. Notice that users typically retrieve more than 5-10 answers.

4 Conclusions

We proposed an approach for shape matching and shape similarity retrieval based on dynamic programming. Our algorithms handle occluded, noisy and deformed shapes and are independent of translation, scale, rotation, starting point selection. We tested our algorithms on two databases of marine life species with open and closed shapes respectively. The evaluation showed that our approach is very well suited for retrievals by shape content.

Acknowledgements

We are grateful to Z. Rao for valuable contributions to this work. We are also grateful to P. Kaklamanis and C. Genzis for their help in the experiments, R. Jeske and E.Voutsakis who

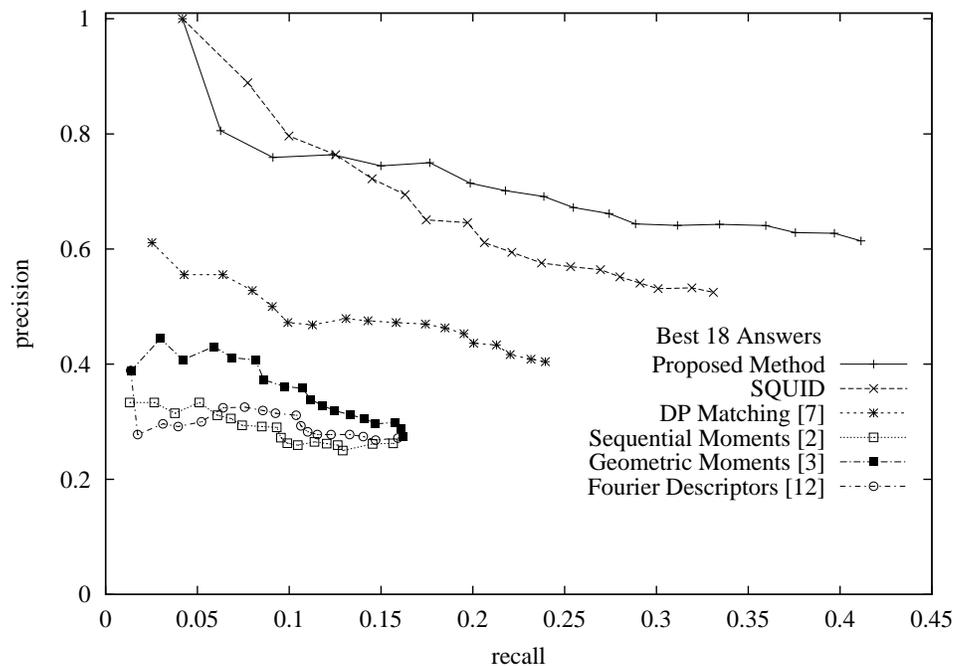


Figure 6: Precision-recall diagrams for the CLOSED dataset.

developed an interactive environment for the evaluation of visual queries, Prof. G. Bebis of the Dept. Comp. Science at the Univ. of Nevada for providing us the codes of the Fourier and moment methods for closed curves and to Prof. F. Mokhtarian of the Centre for Vision, Speech and Signal Processing laboratory at the Univ. of Surrey, UK, for providing us the marine dataset.

References

- [1] Gdalyahu Y and Weinshall D (1999) Flexible Syntactic Matching of Curves and its Application to Automatic Hierarchical Classification of Silhouettes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(12):1312–1328.
- [2] Gurta L and Srinath MD (1987) Contour Sequence Moments for the Classification of Closed Planar Shapes. *Pattern Recognition*, 20(3):267–271.
- [3] Hu MK (1962) Visual Pattern Recogn. by Moment Invariants. *IRE Trans. on Info. Theory*, IT-8:179–187.
- [4] Kalvin A, Schonberg E, Schwartz JT, and Sharir M (1986) Two-Dimensional, Model-Based, Boundary Matching Using Footprints. *The Intern. Journal of Robotics Research*, 5(4):38–55.
- [5] Koch MW and Kashyap RL (1989) Matching Polygon Fragments. *Pattern Recognition Letters*, 10(5):297–308.
- [6] Milios E and Petrakis EGM (1998) Efficient Shape Matching and Retrieval at Multiple Scales. TR CS-1998-11, Department of Computer Science, York University, Toronto. <http://www.cs.yorku.ca/techreports/1998/CS-1998-11.html>.
- [7] Milios E and Petrakis EGM (2000) Shape Retrieval Based on Dynamic Programming. *IEEE Trans. on Image Processing*, 9(1):141–147.

- [8] Mokhtarian F (1995) Silhouette-Based Object Recognition through Curvature Scale Space. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(5):539–544.
- [9] Mokhtarian F and Mackworth A (1986) Scale-Based Description of Planar Curves and Two-Dimensional Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):34–43.
- [10] Siddiqi K, Shokoufandeh A, Dickinson S, and Zucker S (1999) Shock Graphs and Shape Matching. *Intern. Journal of Computer Vision*, 35(1):13–32.
- [11] Ueda N and Suzuki S (1993) Learning Visual Models from Shape Contours Using Multi-scale Convex/Concave Structure Matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):337–352.
- [12] Wallace TP and Wintz PA (1980) An Efficient Three-Dimensional Aircraft Recognition Algorithm Using Normalized Fourier Descriptors. *Computer Graphics and Image Processing*, 13:99–126.