Learning Chance Probability Functions for Shape Retrieval or Classification

Boaz J. Super
Computer Science Department, University of Illinois at Chicago
super@cs.uic.edu

Abstract

Several example-based systems for shape retrieval and shape classification directly match input shapes to stored shapes, without using class membership information to perform the matching. We propose a method for improving the accuracy of this type of system. First, the system learns a set of chance probability functions (CPFs). The CPFs estimate the probabilities of obtaining a query shape with particular distances from each training example by chance. The learned CPFs are used at runtime to rapidly estimate the chance probabilities of the observed distances between the actual query shape and the database shapes. These estimated probabilities are then used as a dissimilarity measure for shape retrieval and/or nearest-neighbor classification. The CPF learning method is parameter-free. Experimental evaluation demonstrates that: (1) chance probabilities yield higher accuracy than Euclidean distances; (2) the learned CPFs support fast matching; and (3) the CPF-based system outperforms prior systems on a standard benchmark test of retrieval accuracy.

1. Introduction

The focus of this paper is many-to-one shape matching using example-based methods. These are methods that do not use geometric or probabilistic models of shape classes, but instead match input shapes directly to stored examples. They are frequently used for *shape retrieval*, in which a set of shapes similar to a query shape are found in a database [1,8,9,14,19,21]. Shape retrieval does not require class labels. However, if class labels are available, then shape retrieval can be used to perform nearest-neighbor *classification*, by retrieving the most similar stored shape and outputting its label (e.g., [1,9,19,20]). This paper is concerned with how to improve the accuracy of retrieval and classification under the assumption that class membership information is not used in the matching step.

The distribution of shapes in feature spaces is generally not uniform. Equal distances from the input shape to different stored shapes do not have the same significance in high-density and low-density regions of the space. This suggests that replacing the feature-space distance measure with a probability measure based on the distribution of the shapes in the space would lead to improved accuracy. One possible approach is kernel-based estimation of the density of shapes in shape space. However, kernel-based methods typically require the specification of a kernel scale parameter and are thus not parameter-free [6].

This paper presents a parameter-free approach for implicitly using the density of shape space to judge the similarity of the input shape to the stored shapes. It is based on the probability of an input or query shape being close to a stored shape by chance. The less likely it is that an observed dissimilarity between a query shape q and a stored shape s occurred by chance, the more similar q and s are considered to be.

During the training phase, a set of *chance probability functions* (CPFs) are learned from the training data, one CPF per shape. During retrieval or classification, the learned CPFs are used to rapidly estimate the chance probabilities of the specific query shape. It is the learning that makes the method feasible: since each CPF depends on all of the shapes in the database, it is too expensive to compute the CPFs during run-time. CPFs are also storage-efficient since the properties of nearest-neighbor methods can be exploited to compress the representation of each CPF into a relatively small lookup table. On a 1,400-shape MPEG-7 benchmark database [14], training takes about 10 minutes and yields pairwise shape matching times of 2.5 ms at runtime.

The experimental results presented in this paper demonstrate a significant increase in accuracy when chance probabilities are substituted for normalized squared Euclidean distances in an example-based retrieval and classification system.

This paper also compares the system's retrieval accuracy to the published retrieval accuracy scores of 11 prior systems on the MPEG-7 benchmark test. Both versions of our system outperform the prior published systems. The version using chance probabilities has significantly higher accuracy than prior systems, and is also significantly faster than the systems based on alternative approaches.

2. Related work

2.1 Models of statistical inference

Bayesian methods for estimating posterior probability densities provide optimal accuracy when class membership information and sufficient training examples are available [3,23]. When class information is not available, as in unlabeled shape retrieval, or not used to generate class models, as in nearest-neighbor shape classification, then a class-free (non-Bayesian) framework is needed.

Our approach is related to classical statistical hypothesis testing, in that a low probability of an event occurring by chance is used to infer that the event did not occur by chance. However, instead of making decisions based on thresholds (confidence levels), the chance probabilities are used to rank the database shapes relative to the query shape.

Our approach is also related conceptually to the non-accidentalness principle, in which a low probability of an image property occurring as a result of an accidental viewpoint or configuration is used to infer that the property exists in the scene [22,15]. Our approach also has similarities with a method for calculating shape matching thresholds using chance probabilities [17]. Our work uses chance probabilities to rank shape matches for retrieval and classification instead of for setting a threshold. Our method also differs from [17] in that it directly estimates the chance probabilities, does not assume independent features, and can be used on high-dimensional feature spaces.

2.2 Shape matching

Many different methods have been proposed for matching shapes. Global shape feature methods match vectors of measurements such as circularity, eccentricity, and moments [7]. These methods are fast but have limited accuracy on larger shape databases. Some transform-based methods, which represent shapes in alternative (non-spatial) domains such as wavelet coefficients [5] or curvature scale space [16], are relatively accurate and efficient. Structural methods represent shapes as parts and relationships, using graphs, trees, or strings [4,8,12,24]. They can handle large shape dissimilarities, but are computationally expensive.

A broad class of approaches match shapes directly as spatial data [1,2,10]. Spatial matching methods register two shapes and measure the residual difference between them. They differ in how correspondences are found and in the family of transformations considered. In general, methods that solve the correspondence problem are too slow to be of practical use in example-based retrieval or classification on large shape databases. One solution is to use *alignment* methods that use minimal-size sets of points (frame groups) to align the two shapes [10]. Because an alignment is based on few point correspondences (usually two to four), multiple alignments can be tested, yielding redundancy that enables similar shapes to be matched even if they differ in part. The cost of testing many alignments is potentially high, so methods that reduce the number of candidate alignments and/or reduce the cost of verifying alignments are used (e.g., [11,18]). The current work uses a very efficient alignment matcher based on [20] as the testbed for evaluating the use of chance probabilities.

3. Learning chance probability functions

Let S denote the shape space (feature space) and let $p(s)$ denote the probability density within this space. Let $D(s,s')$ denote the Euclidean distance between shapes s and s'. Finally, let $P_s(d)$ denote the probability of a shape occurring within distance d of s:

$$P_s(d) = \text{Prob}\{\, s' \in S : D(s,s') \leq d \,\}.$$

One approach for estimating $P_s(d)$ is to first estimate $p(s)$ using kernel-based methods. This requires selecting the value of a window size parameter or parameters, and is costly in high-dimensional spaces when there are a large number of training examples.

Instead, we treat $p(s)$ as a discrete density defined by the n training (database) shapes, and estimate

$$P_s(d) = |\, \{\, s' \in S : D(s,s') \leq d \,\} \,| \,/\, |\, \mathrm{S} \,| ,$$

where | . | denotes set cardinality.

Suppose that the shapes in S are ranked by distance from s; i.e., $s_1,s_2,\ldots,s_n$, where $s_1 = s$. Then $P_s[D(s,s_i)] = i/n$ for $i = 1,\ldots,n$. These probabilities are simply the rank orderings of the training shapes by distance from s, normalized by n. The $P_s(d)$ function for each s is a cumulative probability distribution. An example is shown in Figure 1 (the stair function). It is straightforward to store these cumulative probability functions as lookup tables indexed by d. A complete lookup table requires n entries; however, since in example-based retrieval and classification methods the results are usually determined by the closest examples, it is only necessary to keep the first few entries $1,\ldots,n_1$ of the table plus a small number (n_2) of the remaining entries, which can be interpolated if necessary.

Now let q represent an unknown shape. The *chance probability of q relative to s* is defined as

$$P_s[D(s,q)] = \text{Prob}\{\, s' \in S : D(s,s') \leq D(s,q) \,\}.$$

This is the probability of getting a shape as close or closer to s than q by chance, as estimated by the actual distribution of the database shapes relative to s. The *smaller* the value of $P_s[D(s,q)]$, the *less* likely it is that q is near to s by chance. Therefore, when $P_s[D(s,q)]$ is small, we infer that q is similar to s.

Experiments have shown that better results are obtained when $P_s(d)$ is linearly interpolated between the steps, as shown in Figure 1, since otherwise ties are common. In the rest of the paper $P_s(d)$ will refer to this modified form which we call the *chance probability function* (CPF) of s. Since the functions $P_s(d)$ are stored as lookup tables, the interpolation is performed during retrieval or classification rather than during learning.

The chance probabilities can be used by a shape retrieval or nearest-neighbor classification system, as follows. Given an unknown shape q, $D(s,q)$ is computed for every training example s as in regular nearest-neighbor retrieval. Then, for each s, the chance probability $P_s[D(s,q)]$ is found using the lookup table for the CPF of s. From that point forward, the system proceeds unchanged except that the chance probabilities $P_s[D(s,q)]$ are substituted for the distances $D(s,q)$.

It should be noted although the chance probabilities are used as a dissimilarity measure they are not metric distances: neither symmetry nor the triangle inequality are satisfied. In the experimental results below we use a symmetric version $P(s,q) = \max[P_s(D(s,q)), P_q(D(s,q))]$. The second term, $P_q(D(s,q))$, is the probability that a database shape will be as close or closer to q than s by chance, under the assumption that the training set is representative of the population of query shapes. The max operator enforces a conservative inference strategy, since the larger the chance probability, the less likely the two shapes are to be similar in reality.

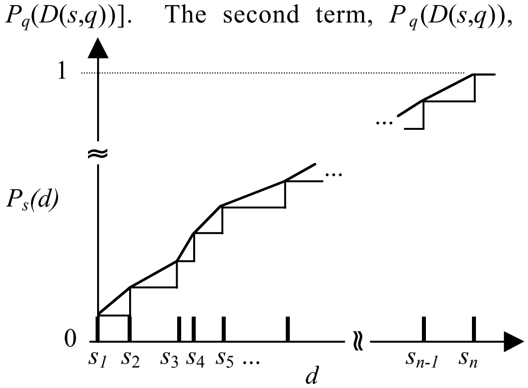

Figure 1. Chance probability function for one shape. The piecewise linearly interpolated version is used.

The use of chance probabilities differs fundamentally from the use of class-conditional probabilities $p(q|s)$ or posterior probabilities $p(s|q)$. In these approaches, high similarity is inferred from *high* evidence for the *class* hypothesis. In our approach, high similarity is inferred from *low* evidence for the *chance* hypothesis. A principal advantage of the chance probability approach is that it can be used when class information is not available.

4. Shape matcher

The shape retrieval and classification system used for this evaluation is a simplified version of an existing fixed-correspondence alignment matcher. It is summarized here; the details are given in [20,21].

Each shape is represented by a closed planar contour, such as a silhouette of an object. A set of key points corresponding to significant positive local maxima and negative local minima of curvature are extracted. Figure 2a shows the key points of a typical shape.

For each key point **x**, let **y** be the contour point furthest from **x** in straight-line distance, and let **z** be the contour point furthest from the line defined by **x** and **y**. Whereas **x** is a key point, **y** and **z** may be general contour points, not just key points. For each key point **x**, the points **x**, **y**, and **z** form a frame group used to define a similarity transformation, $T_{\mathbf{x}}$, for mapping the contour to a canonical reference frame. Specifically, $T_{\mathbf{x}}(\mathbf{x}) = (0,0)$, $T_{\mathbf{x}}(\mathbf{y}) = (1,0)$, and if necessary, the contour is reflected so that $T_{\mathbf{x}}(\mathbf{z})$ is above the x-axis. (See Figure 2b-d.) The representation of the contour in the canonical reference frame is invariant to position, orientation, scale, and reflection. Since for a given shape, **y** and **z** are determined by the choice of **x**, there is one such invariant representation for each key point **x**. For brevity, we refer to each such representation as a *pose* of the contour.[1] Figure 2b-d shows three of the 15 poses of the example shape in Figure 2a.

Two shapes are compared by comparing all pairs of poses of the two shapes within the canonical reference frame and selecting the best-matching pair of poses. (The pose matching will be described later.) This method is thus an alignment method, where a

[1] Usually 'pose' refers to the transformation rather than to the transformed contour.

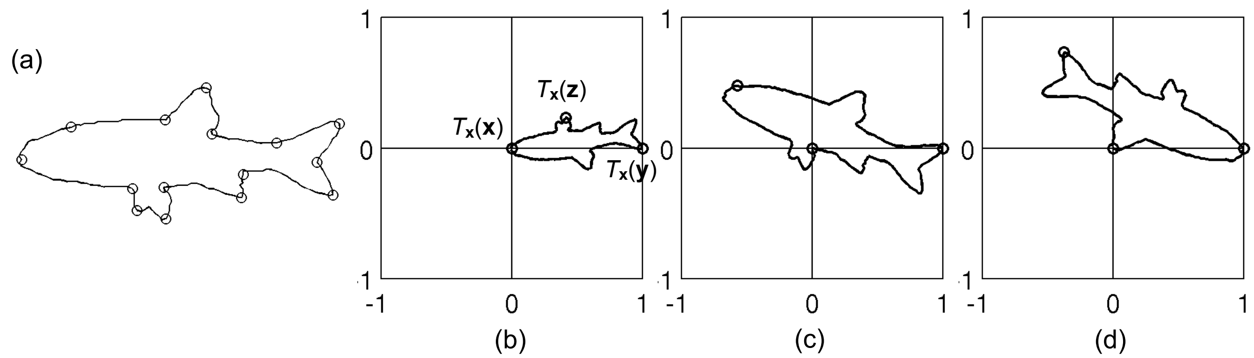

Figure 2. (a) Example shape with key points. Key points often lie on significant projections or concavities. (b)-(d) Three of the 15 normalized contours, or poses, of the shape shown in (a). There is one pose per key point. The transformed positions of **x**, **y**, and **z** are indicated by dots.

geometric construction (the triple **x**, **y**, **z**) has been used to ensure that the number of poses per shape is the number of key points in the shape, rather than the square or cube of the number of key points (as in e.g., [10,11,18]). This limits the number of possible alignments (pose-to-pose matches) to quadratic complexity.

The most computationally expensive part of many direct-matching alignment methods is finding a good flexible correspondence between the shapes after they have been aligned. To obtain sufficient speed, we use a fixed correspondence instead. Every pose of every shape in the training set is resampled with an identical n-point sampling pattern starting at the origin (0,0) = $T_{\mathbf{x}}(\mathbf{x})$. Let $(x_1,y_1),\ldots,(x_n,y_n)$ be the coordinates of the new sample points. Each pose can thus be represented as a single point in a $2n$-dimensional feature space by the vector $(x_1,\ldots,x_n,y_1,\ldots,y_n)$. Two poses are matched by computing the sum of squared distances between correspondingly numbered points, which is the squared Euclidean distance in the feature space (Figure 3).

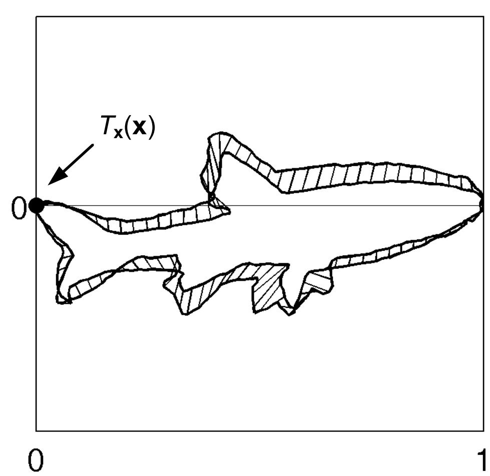

Figure 3. Pose-to-pose matching in the canonical reference frame.

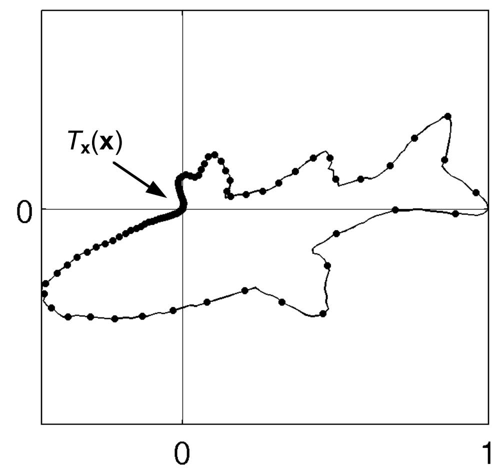

Figure 4. Logarithmic sampling.

Normalization of the squared distance by the arc lengths of the poses produces better results [20], so we will use the normalized squared distance (NSD) to provide a stronger test of the chance probability method.

The fixed correspondence is an approximation to flexible correspondence. The approximation is exact at $T_{\mathbf{x}}(\mathbf{x})$, where all poses of all shapes correspond, and tends to degrade with distance along the contour from $T_{\mathbf{x}}(\mathbf{x})$. For this reason, we use a non-uniform logarithmic sampling pattern that samples the shape more densely nearer to the key point $T_{\mathbf{x}}(\mathbf{x})$ (Figure 4).

The fixed-correspondence approximation to a flexible correspondence is most accurate for shapes that relatively similar. Since in nearest-neighbor approaches the system's output is dominated by the relatively similar matches, the use of fixed correspondence does not significantly degrade the system's accuracy even as it greatly increases the speed.

To perform retrieval, the query shape is compared to the training shapes using the matcher, and the N

Table 1. Retrieval accuracy on 1,400 test objects (MPEG-7 test). The scores for methods 8-13 are from [14]. Shape matching time includes all pose-to-pose matches between two shapes.

Method	Retrieval accuracy	Avg shape matching time (milliseconds)
1. Chance probabilities (CPFs)	**82.69%**	**2.5**
2. Normalized squared distances (NSDs)	79.36%	0.3
3. RACER (Super) [20]	79.09%	0.4
4. Distance Sets (Grigorescu & Petkov) [9]	78.38%	700
5. Curve edit distance (Sebastian et al.) [19]	78.17%	>1000
6. Eigenshapes (Super) [21]	76.92%	0.6
7. Shape contexts (Belongie et al.) [1]	76.51%	200
8. Part correspondence (Latecki & Lakamper) [12,13]	76.45%	50
9. Curvature scale-space (Mokhtarian et al.) [16]	75.44%	n/a
10. Multilayer eigenvectors (Hyundai)	70.33%	
11. Zernike moments (Khotanzan & Hong)	70.22%	
12. Wavelet (Chuang & Kuo) [5]	67.76%	
13. Skeleton DAG (Lin & Kung)	60%	

best-matching shapes are returned. To perform classification, the label of the single best-matching shape is returned.

We will test the method with two dissimilarity measures: NSD [20] and chance probabilities. The chance probabilities are computed using NSD as the measure D in Section 3. Note that during the training stage, the CPFs are computed from the set of all poses of all database shapes.

5. Experimental results

We used an MPEG-7 1,400-shape test database and retrieval accuracy measure [14] to compare chance probabilities versus normalized squared distances. This benchmark test is the one used most often in the literature.

The MPEG-7 test database consists of 70 classes with 20 examples per class, for a total of 1,400 shapes. The shapes in the database come from a variety of sources and represent both natural and artificial objects. The database is challenging due to the presence of examples that are visually dissimilar from other members of their class, and examples that are highly similar to members of other classes [14].

We used the standard test procedure from [14]. Each database shape is used as a test query. A retrieval is counted as correct if it is in the same class as the query. The number of correct retrievals in the top 40 ranks is counted, including the self-match. Each system's accuracy is reported as a percentage of the maximum possible number of correct retrievals, which is 28,000 (1,400 shapes × 20 correct retrievals per shape). This percentage score will be referred to as retrieval accuracy or the "bullseye" score (following [14]).

Preprocessing of the 1,400 shapes resulted in 20,035 poses (an average of 14 poses/shape). The number of sample points was set to n = 100, so the dimensionality of the shape space was 200. During the training stage, the CPFs were computed using the entire set of 20,035 poses. There were no parameters to select for the learning. Each CPF lookup table was compressed 100 to 1, from 20,035 to 200 (n_1 = 100, n_2 = 100).

Table 1 shows the bullseye score of the current system using both chance probabilities and normalized squared distances, and also the published accuracies of 11 prior methods. The performance of each system was measured by the creators of that system [14,1,9,19-21]. The retrieval accuracy of the current method is 82.69% when CPFs are used and 79.36% when NSDs are used. This is a relatively large increment in performance compared to the incremental gains of the several most recent methods in Table 1. It represents an additional 933 shapes correctly retrieved. To our knowledge, the CPF version of our system has a higher retrieval accuracy than any other published approach.

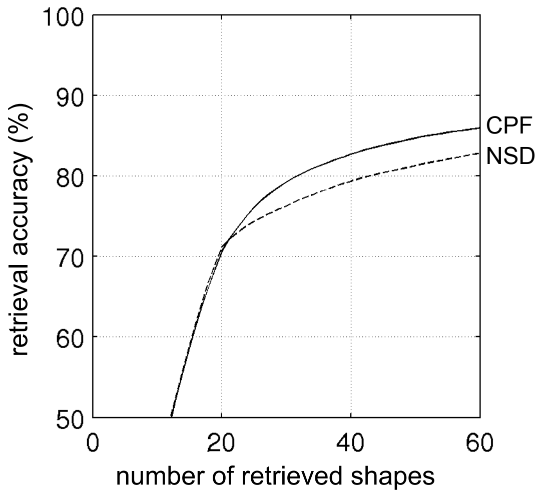

Figure 5. Retrieval accuracy as a function of the number of retrieved shapes.

Figure 5 displays retrieval accuracy for each method as a function of the number of retrieved items, N. Since there are 20 examples per class in the database, achievable recall is limited by N when $N < 20$. The CPF method provides a significant advantage over NSD when $N > 20$.

We also measured classification accuracy on the same database. A 1-NN classifier was evaluated using a leave-one-out procedure [6]. The results are shown in Table 2. A direct comparison of classification performance with most of the other methods in Table 1 is not possible, since most of the papers reported only retrieval accuracy and not classification accuracy. The CPF method results in fewer errors than the NSD method. However, all three methods in Table 2 have a very low number of errors, and the differences are not significant in a practical sense.

Table 2. 1-NN classification accuracy on 1,400 test shapes (MPEG-7 database). Leave-one-out evaluation was used.

Method	Successes	Errors
chance probabilities	1,360 (97.1%)	40 (2.9%)
normalized squared distances	1,357 (96.9%)	43 (3.1%)
RACER [20]	1,355 (96.8%)	45 (3.2%)

Learning the CPFs took 10 minutes for 20,035 poses in Matlab on a 1.8 Ghz Pentium-4 PC. Using these CPFs, each pairwise shape match, which includes all pose-to-pose matches between the two shapes, took 2.5 ms on average. The CPF table lookups account for the extra time required over the NSD version. However, even with this extra processing, the current approach is still significantly faster than the alternative approaches in Table 1, with the exception of earlier versions of the current approach.

6. Conclusion

We used chance probability functions to improve example-based shape retrieval and shape classification by implicitly using the density of the shape space. The learning method is parameter-free and makes no use of class membership information. Most of the work is performed during the learning stage, enabling rapid retrieval and classification. A direct comparison demonstrated that using chance probabilities instead of normalized squared distances yields higher retrieval and classification accuracies. Furthermore, the system's accuracy on the MPEG-7 1,400-shape database is the highest published to date. We are currently applying the methods presented here to other retrieval and classification problems, such as partial-shape matching. Finally, we note that the CPF method is general and can be applied to problems involving retrieval of feature vectors in domains other than shape.

Acknowledgements

The author thanks Drs. Mokhtarian and Latecki for generously making contour data files available. Figures 1-5 are used by permission of CVRL; versions of Figures 2-4 have also appeared in [20,21]. This research was supported in part by NSF 9802090. The author thanks the anonymous reviewers for helpful comments.

References

[1] S. Belongie, J. Malik, and J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence* **24**(4), 509-522, 2002.

[2] P. Besl and N. D. McKay, A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence* **14**(2), 239-256, 1992.

[3] J.-L. Chen and G. C. Stockman, 3D Free-Form Object Recognition Using Indexing by Contour Features, *Computer Vision and Image Understanding* **71**(3) 334-355, 1998.

[4] S. W. Chen, S. T. Tung, C. Y. Fang, S. Cherng, A. K. Jain, Extended Attributed String Matching for Shape Recognition. *Computer Vision and Image Understanding* **70**(1), 36-50, 1998.

[5] G. C.-H. Chuang and C.-C. J. Kuo, Wavelet Descriptor of Planar Curves: Theory and Applications. *IEEE Transactions on Image Processing* **5**(1), 56-70, 1996.

[6] R. Duda, P. Hart, and D. Stork, *Pattern Classification* (2nd ed.). Wiley, New York, 2001.

[7] M. Flickner et al., Query by Image and Video Content: The QBIC System. *IEEE Computer* **28**, 23-32, 1995.

[8] Y. Gdalyahu and D. Weinshall, Flexible Syntactic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes, *IEEE Trans. Pattern Analysis and Machine Intelligence* **21**(12), 1312-1328, 1999.

[9] C. Grigorescu and N. Petkov, Distance Sets for Shape Filters and Shape Recognition, *IEEE Trans. Image Processing* **12**(10) 1274-1286, 2003.

[10] D. P. Huttenlocher and S. Ullman, Recognizing Solid Objects by Alignment with an Image. *International Journal of Computer Vision* **5**(2) 195-212, 1990.

[11] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, Object Recognition by Affine Invariant Matching. *Proc. Conf. Computer Vision Pattern Recognition*, 335-344, 1998.

[12] L. Latecki and R. Lakämper, Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **22**(10), 1185-1190, 2000.

[13] L. Latecki and R. Lakämper, Application of planar shape comparison to object retrieval in image databases, *Pattern Recognition* **35**, 15–29, 2002.

[14] L. Latecki, R. Lakämper, and U. Eckhardt, Shape Descriptors for Non-rigid Shapes with a Single Closed Contour. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000.

[15] D. G. Lowe, Three-Dimensional Object Recognition from Single Two-Dimensional Images, *Artificial Intelligence* **31**(3) 355-395, 1987.

[16] F. Mokhtarian, S. Abbasi, and J. Kittler, Efficient and Robust Retrieval by Shape Content through Curvature Scale Space, in A. Smeulders and R. Jain (Eds.), *Image Databases and Multi-Media Search*, World Scientific, New Jersey, pp. 51-58, 1997.

[17] P. Muse, F. Sur, F. Cao, and Y. Gousseau, Unsupervised Thresholds for Shape Matching, *Proc. International Conference on Image Processing,* Vol. II, 647-650, 2003.

[18] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy, Planar Object Recognition using Projective Shape Representation. *Int. J. Computer Vision* **16**, 57-99, 1995.

[19] T. Sebastian, P. Klein, and B. Kimia, On Aligning Curves. *IEEE Trans. Pattern Analysis and Machine Intelligence* **25**(1), 116-125, 2003.

[20] B. J. Super, Improving Object Recognition Accuracy and Speed through Non-Uniform Sampling, *Proc. SPIE Conference Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision,* SPIE Vol. 5267, pp. 228-239, Providence, RI, 2003.

[21] B. J. Super, Fast Correspondence-based System for Shape Retrieval, *Pattern Recognition Letters* **25**(2) 217-225, 2004.

[22] A. P. Witkin and J. M. Tenenbaum, On the Role of Structure in Vision, in Beck, J., Hope, B., and Rosenfeld, A., (Eds.), *Human and Machine Vision*, New York: Academic Press, pp. 481-543, 1983.

[23] J. H. Yi and D. M. Chelberg, Model-Based 3D Object Recognition Using Bayesian Indexing, *Computer Vision and Image Understanding* **69**(1) 87–105, 1998

[24] S. C. Zhu and A. L. Yuille, 1996. FORMS: A Flexible Object Recognition and Modelling System. *International Journal of Computer Vision* **20**(3) 187-212.