Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing

Stefano Berretti, Alberto Del Bimbo, *Member, IEEE,* and Pietro Pala, *Member, IEEE*

***Abstract*—An important problem in accessing and retrieving visual information is to provide efficient similarity matching in large databases. Though much work is being done on the investigation of suitable perceptual models and the automatic extraction of features, little attention is given to the combination of useful representations and similarity models with efficient index structures.**

In this paper we propose retrieval by shape similarity using local descriptors and effective indexing. Shapes are partitioned into tokens in correspondence with their protrusions, and each token is modeled according to a set of perceptually salient attributes. Shape indexing is obtained by arranging shape tokens into a suitably modified M-tree index structure. Two distinct distance functions model respectively, token and shape perceptual similarity. Examples from a prototype system and computational experiences are reported for both retrieval accuracy and indexing efficiency. Shape retrieval has been tested under shape scaling, orientation changes, and partial shape occlusions. A comparative analysis of different indexing structures, for shape retrieval is presented.

***Index Terms*—Image retrieval by shape, shape indexing, shape representation.**

I. Introduction

In RECENT years, visual information retrieval has become a major research area due to the ever increasing rate at which images are generated in many application fields. Visual information retrieval systems support retrieval by visual content by directly addressing image perceptual features such as color [1]–[3], shape [4]–[8], texture [9]–[11] and spatial relationship [12]–[14]. A major problem in visual information retrieval is the combination of useful representations and similarity models with index structures to provide efficient similarity matching in large databases. Commonly, perceptual facts are modeled as feature vectors, such that the computation of their similarity is reduced to evaluate the Euclidean distance between points in a multidimensional feature space. Point access methods (PAMs) [15]–[17] are used as index structures. The basic idea of PAMs is to cluster a point set into multidimensional regions. The distance between points can only be computed as an L_p norm such as the Euclidean (L_2) or the Manhattan (L_1), between their absolute positions. Each region is a node in the index tree. Effectiveness of the search is concerned with the minimization of the overlapping between two or more tree nodes. In order to adapt feature vector dimensionality (which is usually large) with PAMs requirements, vectors are mapped in a low-dimensional distance-preserving space, with appropriate transformations.

This combination is particularly ineffective in the case of retrieval by shape similarity. In fact, it has been verified that the feature vector model and Euclidean distance (more generally, metric distances) between feature points are not suited to model perceptual similarity between shapes [18].

Several systems have addressed the problem of modeling the correspondence between the system and the human similarity judgement, without considering the problem of effective shape indexing. Among them, the QVE system [19], the Photobook system [6], [20] and the system developed in [4].

The QVE system [19] uses edge pixels as features. Shape similarity is obtained by computing the correlation between the query sketch and database edge images. Small shifts and distortions between the database images and the sketch are taken into account by shifting the local correlation between the two blocks of the database edge image and the user's sketch. This solution does not allow indexing. To filter out non pertinent images, images in the database are aggregated according to a set of basic templates.

In the Photobook system [6], Sclaroff and Pentland have represented shapes as modal deformations of a prototype object. Mode amplitude vectors indicate the amount of energy required to align two objects. Shape similarity is computed by considering distances between mode vectors. The authors do not discuss indexing.

Del Bimbo and Pala [4], let the query sketch warp in order to adapt to shapes in the database images. Shape similarity is computed by minimizing a compound cost functional which accounts for the amount of deformation of the sketch and the degree of matching achieved between the deformed sketch and the database shape. Database shapes are analyzed sequentially.

Other systems have defined appropriate global features and used the feature vector model as an effective data model to perform indexing in large databases. Although they support effective indexing, none of them takes into account the effectiveness of retrieval with respect to human similarity judgement.

The QBIC [3] used a set of 22 global shape attributes including area, circularity, major axis orientation, and a set of algebraic moments. The 22-dimensional feature vectors are mapped into a lower dimensional feature space, through the distance-preserving Karhunen-Loéve transform. Indexing is performed in this low-dimensional space by using R^*-tree.

Jain and Vailaya [21], [22] used two sets of shape features to describe the global shape information: a 72 bins histogram of the shape edge direction and seven invariant moments. Shape similarity is calculated as a weighted sum of the Euclidean distance

Manuscript received August 17, 1999; revised October 10, 2000. The associate editor coordinating the review of this paper and approving it for publication was Prof. Wayne Wolf.

The authors are with the Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, 50139 Firenze, Italy (e-mail: berretti@dsi.unifi.it; delbimbo@dsi.unifi.it; pala@dsi.unifi.it).

Publisher Item Identifier S 1520-9210(00)11054-5.

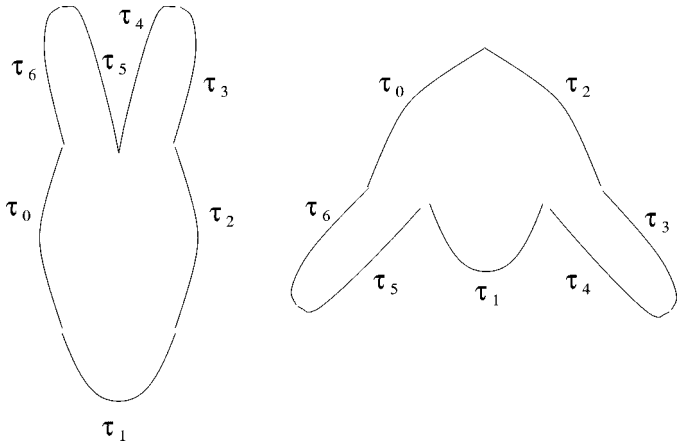

Fig. 1. Similar tokens arranged in a different order can lead to completely different perceived shapes.

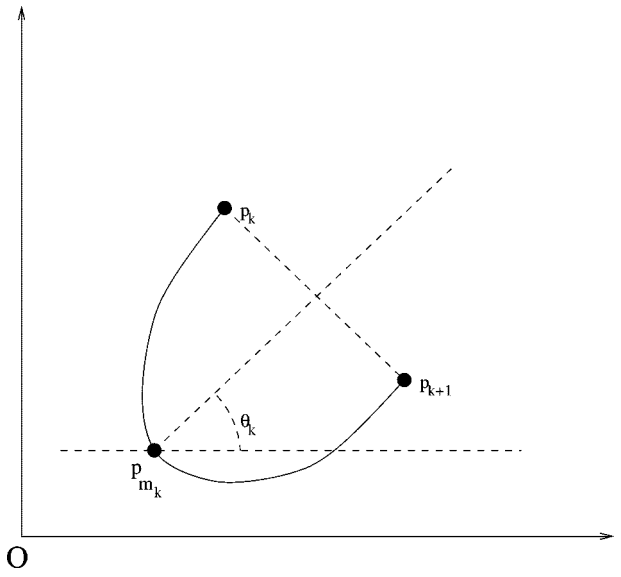

Fig. 2. The angle θ_k is used to measure the orientation of the token in the 2-D space.

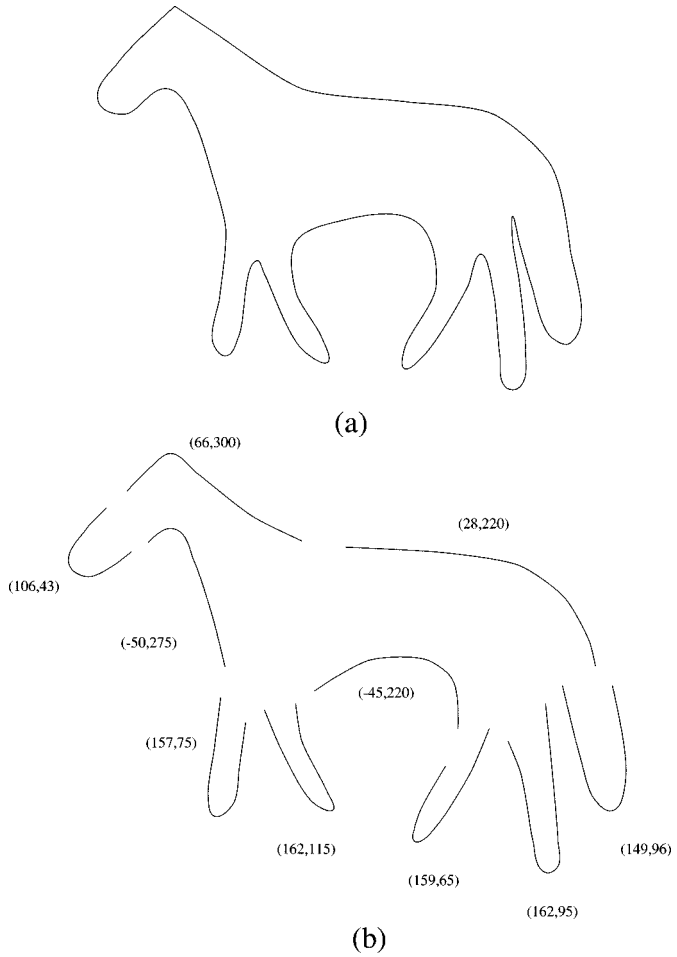

Fig. 3. (a) Shape of a horse and (b) the shape is partitioned at minima of the curvature function. Each token τ_k is described through the features (m_k, θ_k).

between edge histograms and the Euclidean distance between invariant moments. No indexing schema is proposed.

In [23], maxima of curvature-scale-space (CSS) image are used to represent two-dimensional (2-D) shapes at different levels of resolution. Every curvature scale-space contour represents a concavity or convexity of the original boundary. To compare two sets of maxima first any possible change in orientation is found, then two circular shifts to one of the two sets is applied. The summation of the Euclidean distances between the relevant pairs of maxima is then defined to be the matching score between the CSS images. This representation has proven to be robust under scaling, orientation and translation changes. A similar solution has been proposed by Daoudi *et al.* in [24].

The use of global attributes presents limitations in modeling perceptual aspects of shapes, and poor performance in the computation of similarity with partially occluded shapes. More effective solutions have employed local features, such as edges and corners of boundary segments. They are based on the partition of the shape boundary into perceptually significant tokens. This approach has been followed by Petrakis and Milios [25], [26], Grosky and Mehrotra in [27], [28], and Mehrotra and Gary in [29], [30].

Petrakis and Milios have approximated shapes as a sequence of convex/concave segments between two consecutive inflection points. A dynamic programming shape matching algorithm searches for segment correspondences at various levels of shape resolution. Matching of merging sequences of consecutive segments is allowed if this leads to the minimization of a cost function. Shapes are then mapped in a low-dimensional space, and points are indexed using an R-tree.

Grosky and Mehrotra have approximated shapes as polygonal curves: for each vertex, a local feature is defined by considering the internal angle at the vertex, the distance from the adjacent vertex, and the vertex coordinates. A fixed number of these local features is extracted from each shape. A shape is thus represented by an attributed string. Similarity between two shapes is computed as the *editing distance* between the two strings of the boundary features (i.e., the number of changes required to transform one string into another). A binary-tree is used for main memory access. Secondary memory access is accomplished through a m-*way* tree.

Following similar ideas, Mehrotra and Gary have developed a retrieval technique known as Feature Index-Based Similar-Shape Retrieval. Similarly to [27], they assume a polygonal approximation of the shape boundary. This is represented as an

```
procedure TokenDistance( Token τ_i, Token τ_j)
{
        d_curvature = |curvature(τ_i) − curvature(τ_j)|
        d_orientation = |orientation(τ_i) − orientation(τ_j)|
        d_ij = α d_curvature + (1 − α) d_orientation
}
```

Fig. 4. Procedure 1 used to compute the distance between two tokens; α determines the relative weight of curvature and orientation distance.

```
procedure ShapeDistance( Shape A, Shape B )
{
        if (number of tokens in A > number of tokens in B)
                exchange A and B

        for each token a_i in A
                for each token b_j in B
                        d_ij = TokenDistance(a_i, b_j)

        for each token a_i in A
                associate a_i to the nearest token b_j in B
                that is not yet assigned to any tokens in A.
        if (∀ i ∈ 1,...,n  d_i = d_ij ≤ δ_s)
        then
                d(A,B) = (Σ_{i=1}^{n} d_i) / n
}
```

Fig. 5. Procedure 2 used to compute the distance between two shapes; δ_s is the maximum threshold allowed for token distances.

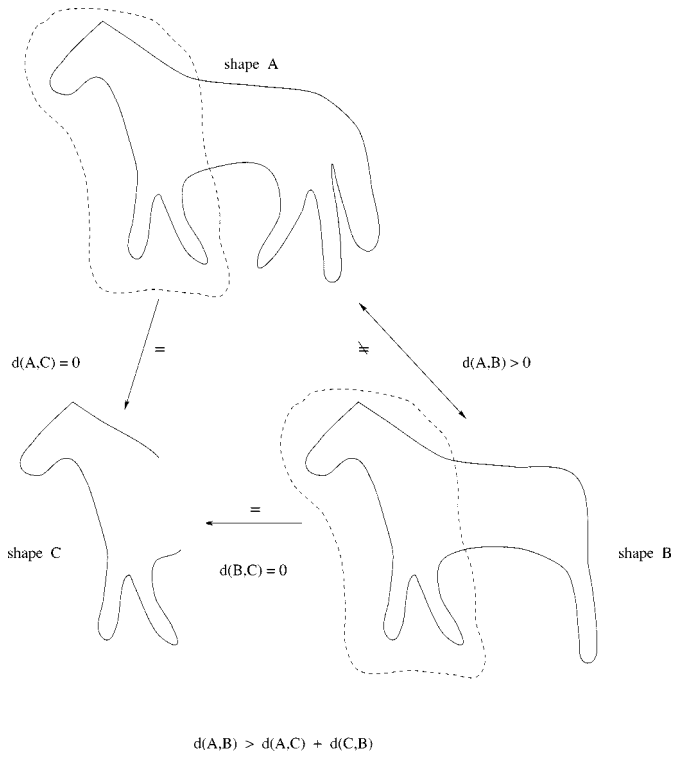

Fig. 6. Example of shape distance. Tokens of C are a subset of the tokens of A and B. It results that $d(A, C) = d(B, C) = 0$.

ordered finite collection of boundary features. Each feature collection represents one segment of the shape boundary. In order to have a low number of segments, segments are defined with a fixed number of adjacent vertices. Boundary feature vectors

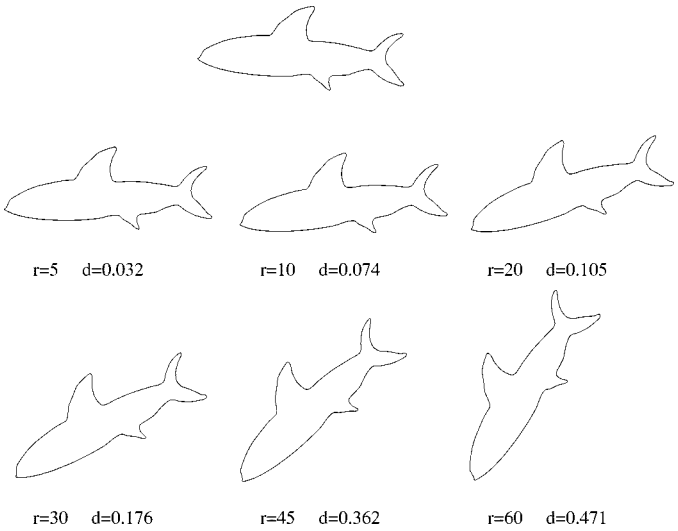

Fig. 7. Measures of the distance between the original shape of a shark and rotated shapes. The shape distance is normalized in $[0, 1]$.

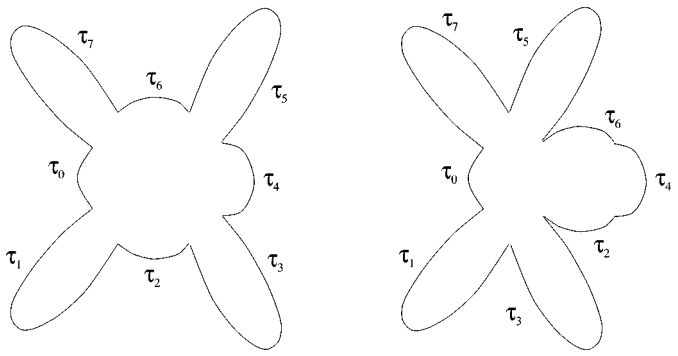

Fig. 8. Set of tokens arranged in the shape on the left are rearranged with a different ordering on the right.

are organized in a kdB-tree. The matching of one or more features doesn't guarantee full shape matching. Consequently, once shapes with similar features are retrieved, shape similarity is checked by overlaying each retrieved shape on the query shape and evaluating the amount of overlap between them.

Since they are based on local boundary features, both the approaches in [27] and [29] allow retrieval of partially occluded objects. They also allow a part of the object boundary to be used as the query example. Limitations are related to the fact that similarity based on polygonal approximation and Euclidean distance between boundary features has little relation with perceptual similarity and cannot be employed for generic shapes. Examples reported by the authors are in fact polygonal shapes of man-made objects. Moreover, in Grosky and Mehrotra's system, sensitivity to small deviations of feature values is critical to retrieval. If the features at the root of two subtrees placed on the same level of the index have a very small difference, and a slightly different version of one of them is compared with both, the wrong path might be chosen, thus leading to incorrect retrieval. To cope with this problem and make retrieval more robust, Grosky and Mehrotra allow ranges for each feature value in the index and consider the three sharpest internal angles of the polygonal approximation and the successive angles in ascending order. In Mehrotra and

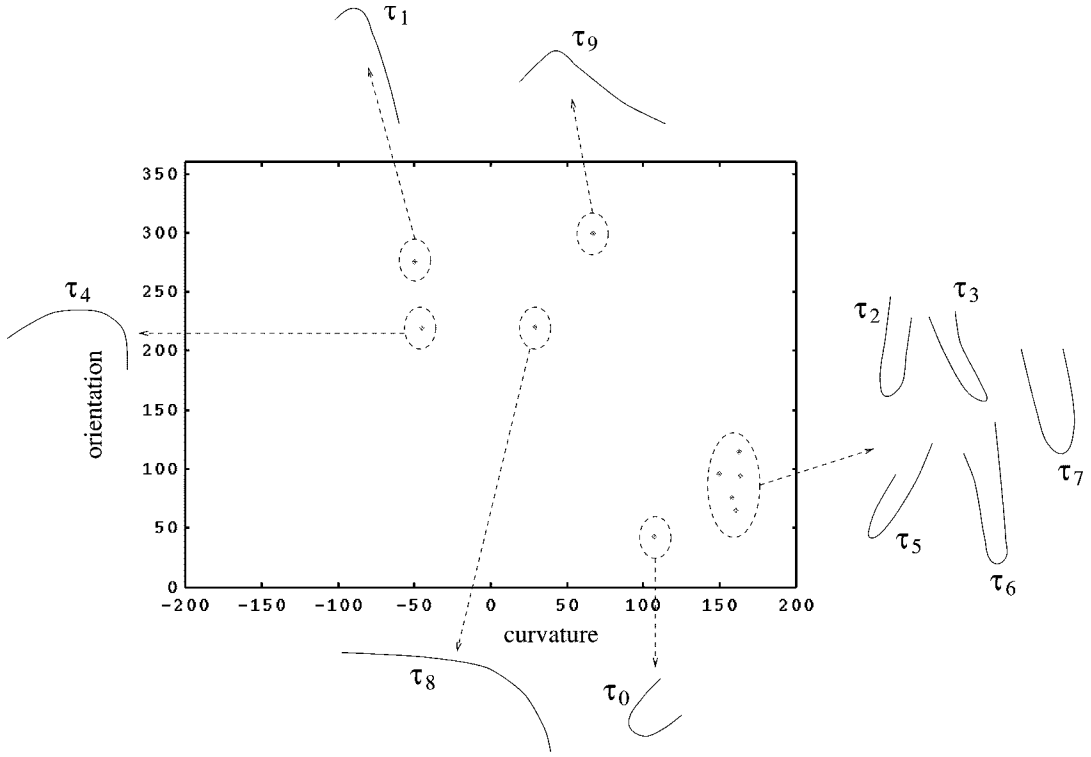

Fig. 9. Features of the horse shape of Fig. 3 are represented in the 2-D space of curvature and orientation. Feature clusters correspond to similar parts in the shape.

Gary's system, the effectiveness of the representation depends on the number of vertices that are used to create a boundary segment. However, a fixed-length chain of vertices almost never corresponds to a salient part of a shape.

In this paper we are proposing retrieval by shape similarity for generic shapes using local features and effective indexing using metric trees. Each shape is partitioned into tokens in correspondence with its protrusions and each token is modeled according to a set of perceptually salient attributes. Token attributes are organized into an M-tree [31], [32]. Differently from PAMs, metric trees deal with cases in which the relevant information is concerned only with relative distances between objects rather than with their absolute positions in a multidimensional space. Objects are organized according to a partition technique based on their distances from a reference object. In order to support retrieval based on perceptual shape similarity, shape similarity is measured according to a nonmetric distance defined as a combination of token distances.

This paper is organized as follows. In Section II, the local representation employed is described. Based on this representation, a shape distance measure is introduced in Section III. In Section IV, the proposed shape description is combined with an M-tree index structure. In Section V, several retrieval examples are shown and considerations about the effectiveness of the approach are expounded. A comparative analysis between the M-tree, the R^*-tree, and the SS-tree index structures for shape-based retrieval is also carried out.

II. Shape Boundary Segmentation

Many contributions have appeared pointing out the central role played in the partitioning process by those points where a shape bends more sharply [33]. Following this approach, it is assumed that partitions occur at points where the shape curvature is minimum. These points identify tokens that correspond to *protrusions* of the curve and can be used as signatures.

From a mathematical point of view, a planar continuous curve can be parameterized according to its arc-length t, and expressed as

$$c(t) = \{x(t), y(t)\}, \quad t \in [0, 1].$$

A Gaussian smoothing can be performed to obtain smooth boundary, and noise insensitivity. The curvature $\gamma(t)$ of $c(t)$ at the point $\{x(t), y(t)\}$ can be expressed as [23]

$$\gamma(t) = \frac{x_t(t)y_{tt}(t) - x_{tt}(t)y_t(t)}{(x_t^2(t) + y_t^2(t))^{3/2}}$$

where x_t, y_t and x_{tt}, y_{tt} are, respectively, the first and second derivatives of x and y with respect to t. Let $P = \{p_i\}_{i=1}^{N}$ be the set of minima of $\gamma(t)$. If we assume that curvature $\gamma(t)$ is continuous, between two consecutive minima p_k, p_{k+1} there is always a maximum of $\gamma(t)$, namely m_k, located at p_{m_k}. A feature that significantly affects the perception of a shape token is its width (narrow tokens are distinguished from wide tokens). We use the maximum value of token curvature m_k to represent the token width: narrow tokens are associated with high values of m_k; wide tokens with low values of m_k. Shape perception is also related to the way in which narrow and wide tokens are arranged: different arrangements result into completely different shapes (see Fig. 1).

Information about token arrangement can be preserved by retaining the orientation of each token in the 2-D space. Given a

token τ_k, its orientation θ_k is defined as the orientation, in polar coordinates, of the vector linking the median point of the segment $p_k - p_{k+1}$ with the point p_{m_k}, calculated with respect to an absolute reference system (see Fig. 2).

According to this, each token is represented by two features (m_k, θ_k) with m_k in $[-180\,180)$ and θ_k in $[0, 360)$. A generic closed curve c, with N partition points, is represented by the set

$$T(c) = \{(m_k, \theta_k)\}_{k=1}^{N}.$$

As an example, Fig. 3 shows the shape of a horse, its tokens, and their descriptions.

The case in which a shape is a perfect circle (i.e., no minima of the curvature function of the shape boundary are identified), is a singular case. In this case, the shape is represented by an unique token with a null curvature and a do not care orientation.

Two shapes are considered similar if they share tokens with similar curvature and orientation, according to an appropriate distance measure.

III. Use of Token and Shape Distances

A common result of many studies in the field of human perception (see [34]–[36] among the others) is that the triangular inequality does not hold in human similarity perception. As a result of this fact, the distance between two shapes, as perceived by humans, is not a metric distance. On the other hand, a metric distance is necessary to effectively organize database items through multidimensional index structures like PAMs [15], [16] and metric indices [31].

Two different contrasting requirements are therefore set for content-based retrieval by shape similarity. They are concerned respectively, with fitting human perception and with the efficiency of data organization and indexing. To cope with both these requirements two distinct distance measures have been defined: a *token distance* and a *shape distance*. The token distance is a metric distance and is used to provide a measure of the similarity between two tokens. The shape distance is a nonmetric distance, defined as a combination of token distances, and is used to derive a global measure of shape similarity which fits humans' perception. Indexing is performed at the token level, by exploiting the metric properties of token distance.

Given two generic tokens τ_1 and τ_2, their distance is computed according to Procedure 1 of Fig. 4.

Curvature and orientation distances are combined in a convex form. The parameter $\alpha \in [0, 1]$ weights the contribute of curvature and orientation in distance computation. Since the token distance fulfills the triangular inequality, the token space (m_k, θ_k) is a metric space.

Given two generic shapes A and B, with n and m tokens respectively (it is assumed that $n < m$; if this is not the case, A and B are exchanged), their distance is computed by combining the similarity of their tokens according to Procedure 2 of Fig. 5.

This procedure allows the computation of shape similarity even in the case of partial shape occlusions. In Fig. 6, an example of shape distance computation is shown. Three shapes A, B, and C are considered. A represents the contour of a horse. C is obtained from shape A by occluding its right portion. The left part of B corresponds to the left part of C and A; its right part is not present in both A and C.

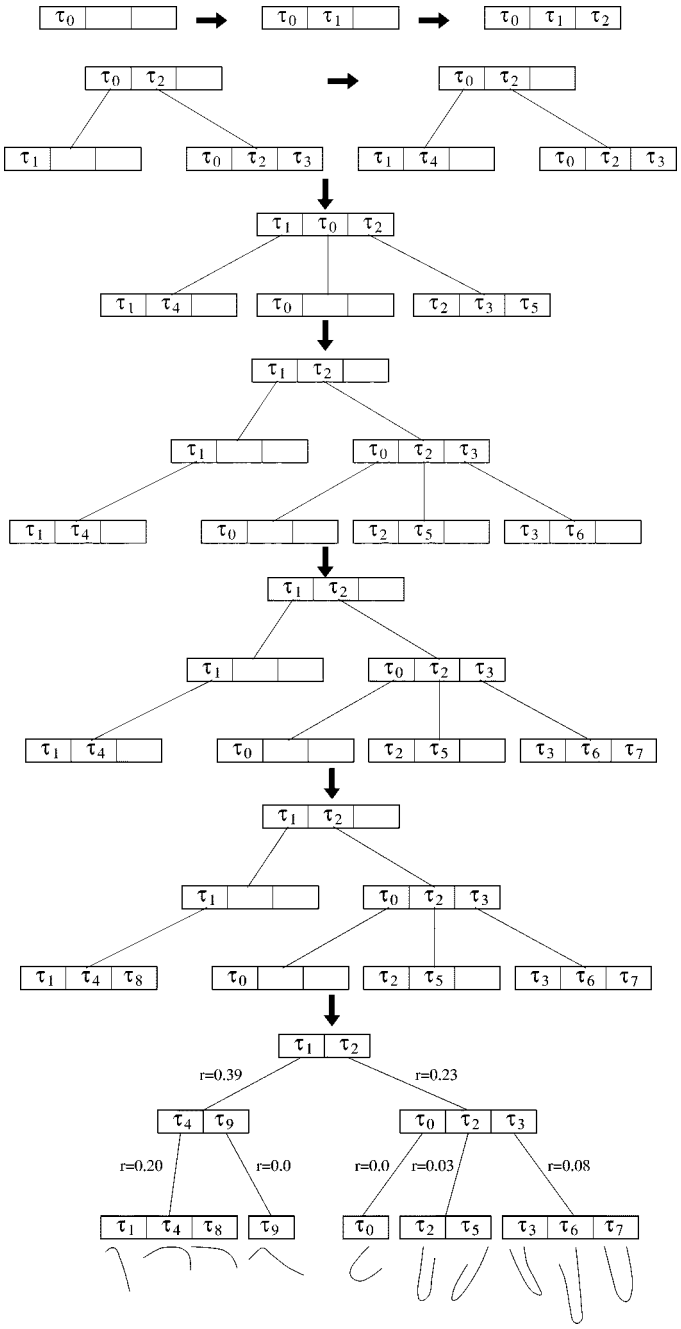

Fig. 10. Organization of tokens of Fig. 9 into the M-tree structure, using *mM-rad* split policy. The cluster size is assumed to be equal to three.

Fig. 6 shows that shape distance doesn't fulfill the triangular inequality. This can be easily proved: Assume that the triangle inequality holds. Then, given A, B, and C, with m, n and l tokens respectively $(m > n > l)$, it follows that

$$d(A, B) \leq d(A, C) + d(C, B). \tag{1}$$

Assume that tokens of C are a subset both of tokens of B and A, that is

$$T(C) \subset T(A) \quad \text{and} \quad T(C) \subset T(B)$$

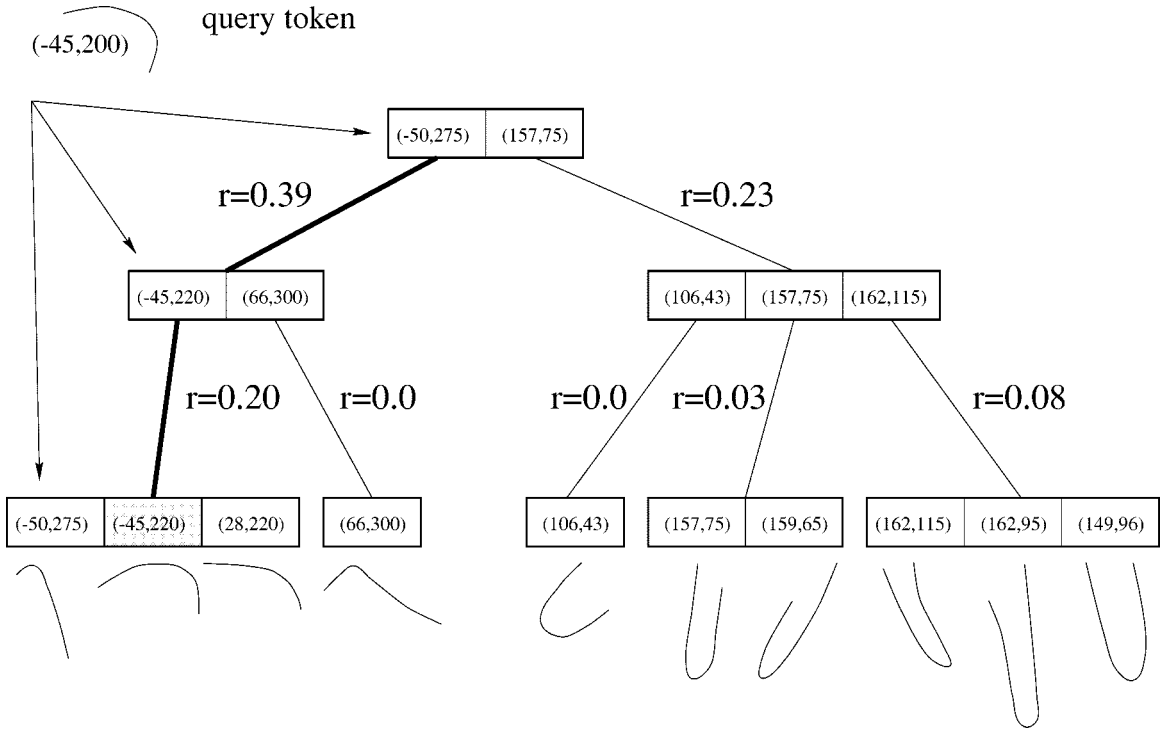

Fig. 11. Token of a query shape is used in the index of Fig. 10. Nodes with nonrelevant tokens are filtered out by evaluating the distance between the query token and the routing objects.

According to the definition of shape distance

$$d(A,C)=0 \quad \text{and} \quad d(C,B)=0. \tag{2}$$

Assume that at least one token in B is not present in A:

$$[T(A)\cap T(B)]\subset T(B)$$

then

$$d(A,B)>0. \tag{3}$$

By combining (2) and (3) it follows that

$$d(A,B)>d(A,C)+d(C,B)$$

which contradicts the initial hypothesis (1). Since there is at least one condition that makes triangular inequality invalid, the initial hypothesis is false in the general case.

Token curvature and orientation attributes are invariant with respect to the size of the shape. According to this, the shape representation is scale invariant.

Whether an absolute reference system (as used in the present implementation) or a local one is selected for token orientation, different properties are obtained for the shape representation. An absolute reference system results into nonrotation invariance. In fact, rotation of the shape affects the orientation of each token. However, similarity gracefully decreases as the rotation of the target shape is increased, as shown in Fig. 7.

On the other hand, a local reference system (the orientation of one of the tokens can be used as a reference, and the other token orientations are computed with respect to it) while results into rotation invariance, presents inefficiency in the case of retrieval with partially occluded shapes. If the reference token is occluded similarity cannot be computed.

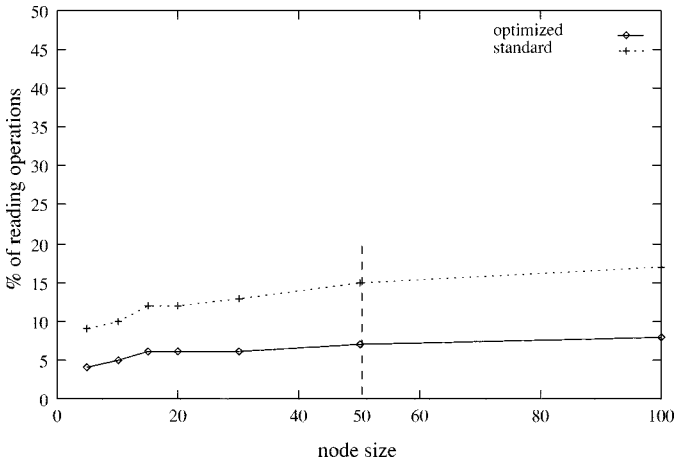

Fig. 12. Percentage of read operations (as a function of the node size) for the optimized and standard tree traversing procedures with respect to a linear scan of the entire shape database.

Shape distance does not depend on the ordering of tokens along the curve. This can determine retrieval of false positives (see Fig. 8). However, since token orientation is considered with respect to an absolute reference system, the degrees of freedom of token arrangements are highly reduced.

IV. Shape Indexing Using a Modified M-Tree

In that a generic token is modeled as a point in the two-dimensional (2-D) feature space of curvature and orientation, the representation of a generic shape results to be a set of points in this space, as shown in Fig. 9.

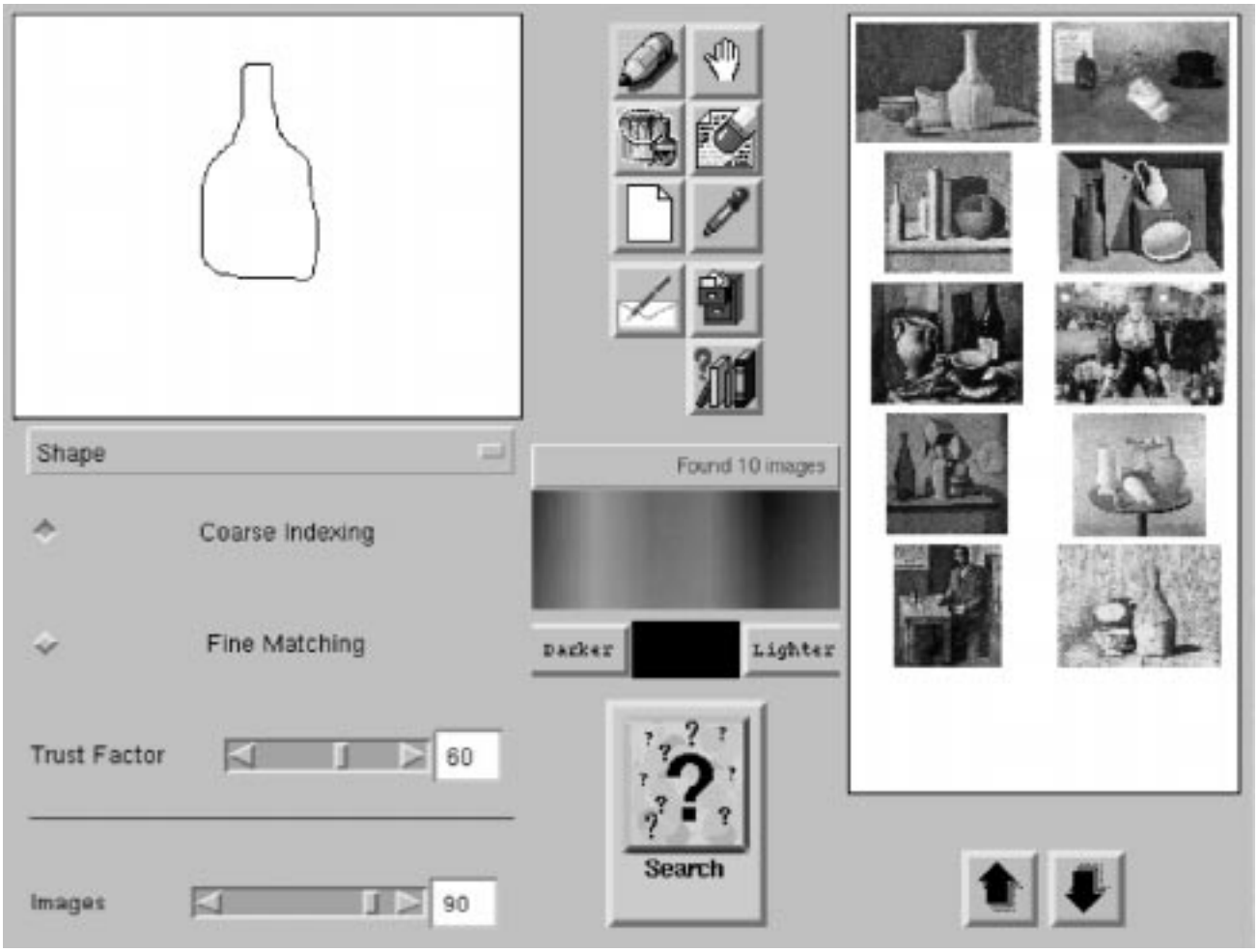

Fig. 13. Sketch representing a bottle and the corresponding output of the indexing process.

Shape tokens are stored in an M-tree index. M-tree organizes tokens as a hierarchical set of clusters, each of which is identified by a *routing object*, (i.e., the center of the cluster), and a *covering radius*, which determines the maximum distance between the routing object and each tokens included in the cluster. When the tree is built, the level at which each token has to be located is determined through the following process: each new entry is compared with the routing objects at each level; the tree traversing path is determined by selecting the routing object which requires the minimum increase of the covering radius to include the entry in the cluster. At the leaf level, if the leaf node selected is full (node overflow), a split of the cluster is performed. The split can propagate from the leaf to the upper nodes, up to the root.

Different split policies can be used. We considered: *mM-Rad* (*minimum of Maximum of Radii*), and *MLB-Dist* (*Maximum Lower Bound on Distance*). They differ in the way in which routing objects are promoted when a cluster split occurs: *mM-Rad*, promotes as routing objects the pair of entries which have the minimum of the maximum of covering radii (thus reducing the extension of covering regions); *MLB-Dist*, promotes the two entries that are at the greatest distance (thus reducing the overlapped area). Fig. 10 shows the organization of the tokens of Fig. 9 into the M-tree structure, using the *mM-Rad* policy. Cluster size is assumed to be equal to 3.

Each node entry has the following format:

$$entry(\tau_i) = [\tau_i, ptr(T(\tau_i)), r(\tau_i), d(\tau_i, P(\tau_i))]$$

where τ_i is the feature vector of the indexed token if the node is a leaf, the token identifier otherwise; $ptr(T(\tau_i))$ is a pointer to the root of the sub-tree $T(\tau_i)$, which includes all τ_j such that their distance from τ_i is less than the covering radius $r(\tau_i)$, that is

$$d(\tau_i, \tau_j) \leq r(\tau_i) \quad \forall \tau_j \text{ in } T(\tau_i);$$

in the leaf nodes, ptr$(\cdot)$ is replaced with the identifier of the shape to which the token belongs; and $d(\tau_i, P(\tau_i))$ represents the distance between the node entry and its parent token $P(\tau_i)$; distance d is a metric distance which is calculated according to Procedure 1 of Fig. 4.

Distances $d(\tau_i, P(\tau_i))$ and $r(\tau_i)$ are precomputed so that the number of accessed nodes and the number of distance computations are reduced at search time.

Given a query token τ_q and a range r, a range query selects all the database tokens τ_i such that

$$d(\tau_i, \tau_q) \leq r.$$

Triangle inequality is used to prune a sub-tree from a search

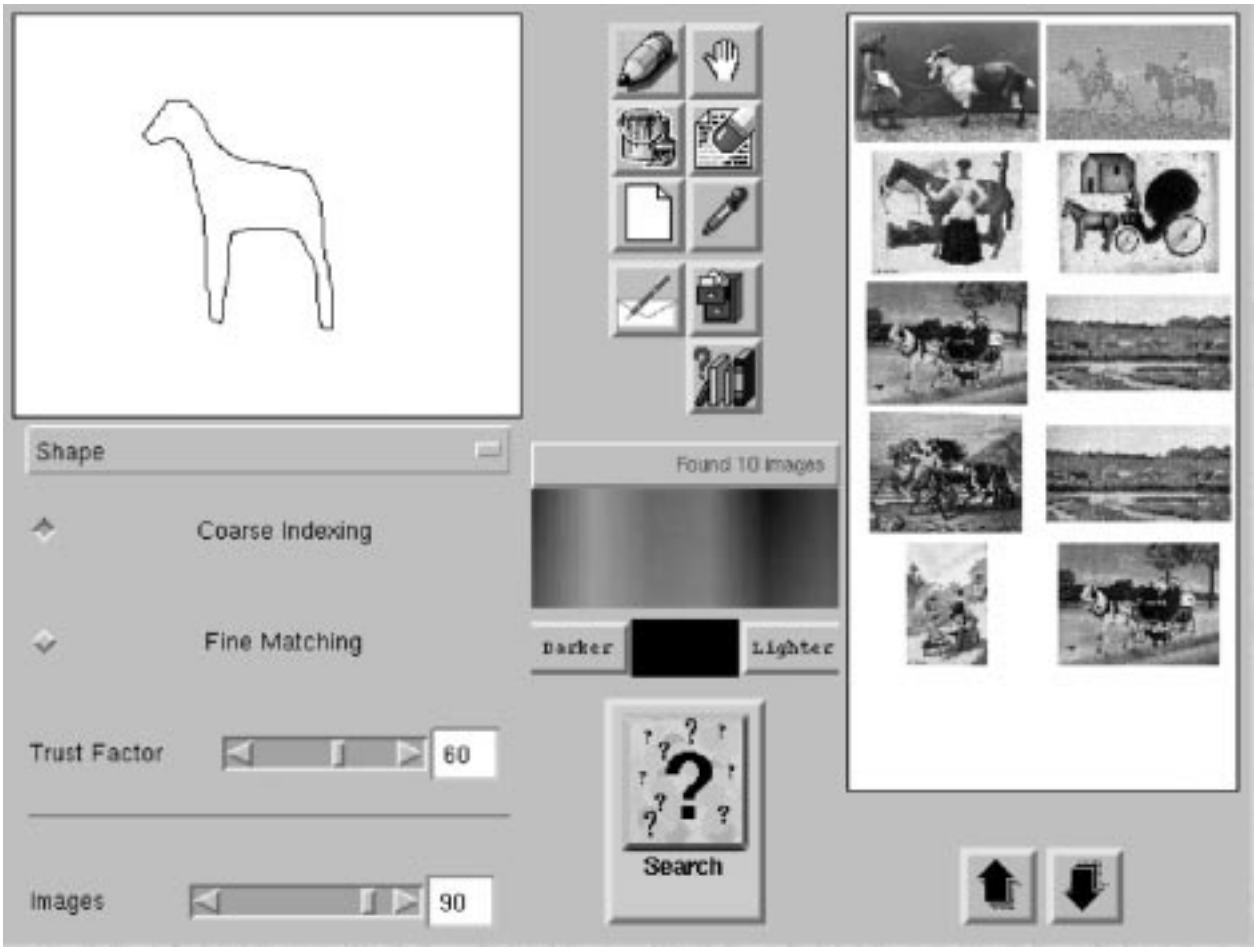

Fig. 14. Sketch representing a horse and the corresponding output of the indexing process. If two different shapes in the same image match the query shape, then the image appears twice in the result set (fifth and tenth ranked images, sixth and eighth ranked images, match the query sketch with two different shapes).

path. In fact, according to triangle inequality

$$\begin{aligned}&\text{if } d(\tau_i,\tau_q) > r + r(\tau_i),\\ &\text{then } d(\tau_j,\tau_q) > r \quad \forall \tau_j \text{ in } T(\tau_i)\\ &\text{if } |d(P(\tau_i),\tau_q) - d(\tau_i,P(\tau_i))| > r + r(\tau_i),\\ &\text{then } d(\tau_i,\tau_q) > r + r(\tau_i) \text{ is true}\end{aligned}$$

and computation of distance $d(\tau_i,\tau_q)$ can be avoided.

A generic query, searching for shapes similar to a shape example, is handled as a multiple token query. Tokens of the query shape are presented to the index (see Fig. 11). Shape identifiers at the leaf level are used for shape integrity. Once similar tokens are retrieved, shape distances are computed separately (using Procedure 2) for each set of tokens with the same shape identifier.

The standard tree traversing algorithm has been optimized to reduce the number of read operations. The comparison between the tokens of the query and the tokens stored in the tree node at a generic level l can eventually produce the same search path at level $l+1$, for several query tokens. To avoid that the same node is loaded from secondary memory several times, only one read is performed for multiple requests.

Fig. 12 shows the percentage of read operations that are saved using optimized and standard tree traversing. Plots are compared with respect to the number of read in the case of a linear scan. Curves are plotted as a function of the node size. As an example, if the node size is 50, with a standard tree traversingprocedure 85% $(100\% - 15\%)$ of read operations are saved with respect to a linear scan of the database. The use of the optimized traversing procedure allows the system to save 93% $(100\% - 7\%)$ of read operations.

V. Experimental Results

A. Retrieval Examples

Figs. 13–15 present examples of retrieval by shape similarity, using a prototype system which implements the proposed solution. The test database contains 1637 shapes of objects extracted from 20th century paintings. Each shape has been sampled at 100 equally spaced points. Descriptions have been organized into the M-tree index structure, according to the procedure discussed in Section IV.

The user can make the search more or less selective, by changing the *trust factor*. The trust factor defines the minimum similarity threshold of token distance. High values restrict the maximum distance admitted between tokens; this also affects the number of similar tokens retrieved. The user can also set the maximum number of images retrieved through the slide bar *images*.

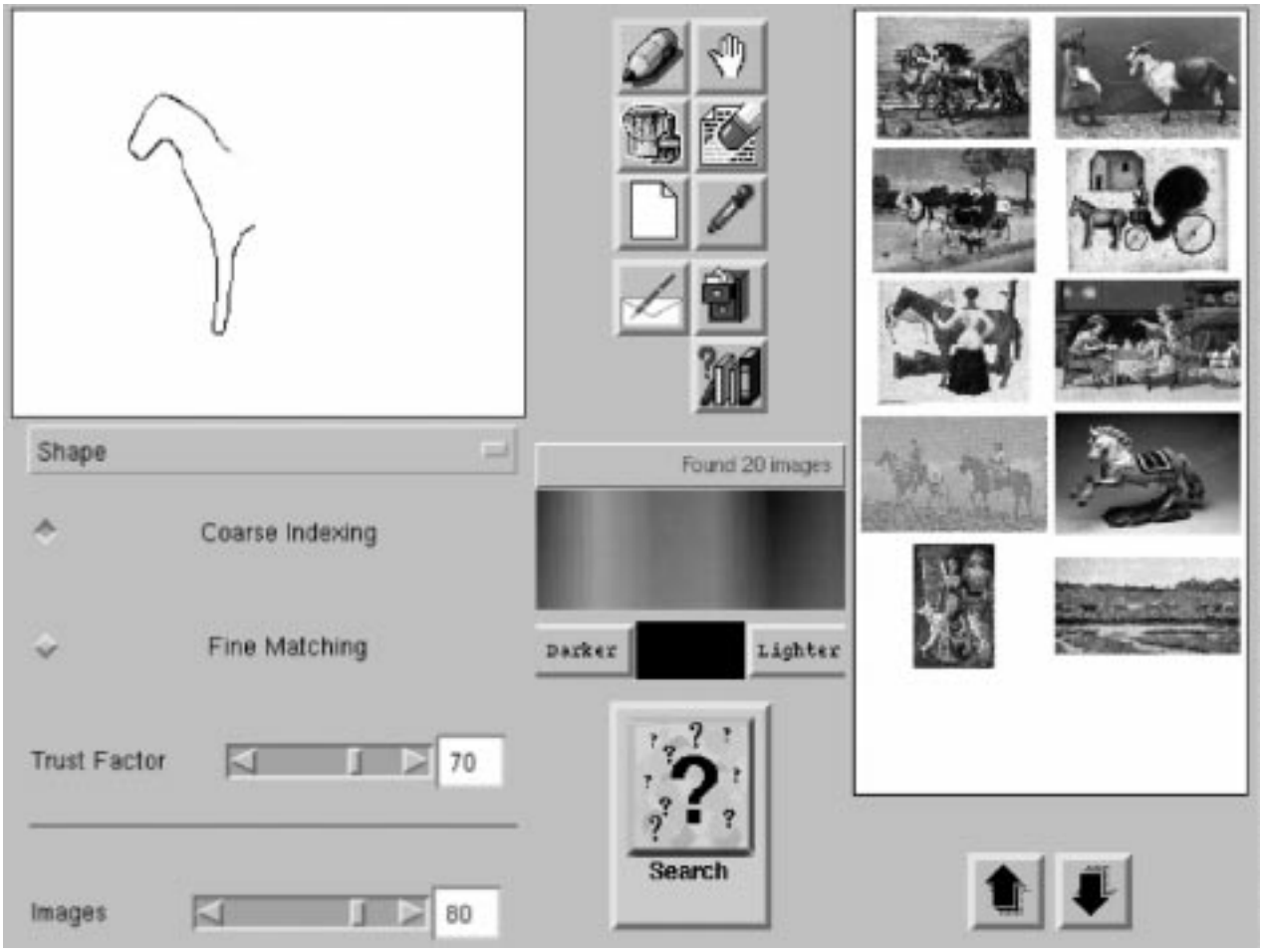

Fig. 15. Sketch representing a part of a horse shape and the corresponding output of the indexing process.

In Fig. 13, it can be observed that the retrieval is insensitive to slight shape differences (e.g., artifacts in the neck or in the body of the bottle).

Fig. 14 shows retrieval results for a more complex query shape. The best ranked images include shapes of similar objects with different postures; neither the difference in the number of legs of the animals, nor partial occlusions affect the quality of the retrieval.

Fig. 15 shows an example of retrieval in the presence of shape occlusions. Although the query shape represents only part of the horse's shape, all best ranked images include shapes similar to the query shape.

B. Performance Analysis

Performance has been analyzed in terms of *retrieval accuracy* and *indexing efficiency*.

Retrieval accuracy is concerned with the effectiveness of shape retrieval at both *quantitative* and *qualitative* level. At quantitative level, precision and recall are computed. At qualitative level, we measure to what extent the system agrees with human perception. Indexing efficiency is evaluated by comparing the modified M-tree index with respect to the R^*-tree [15] and SS-tree [37].

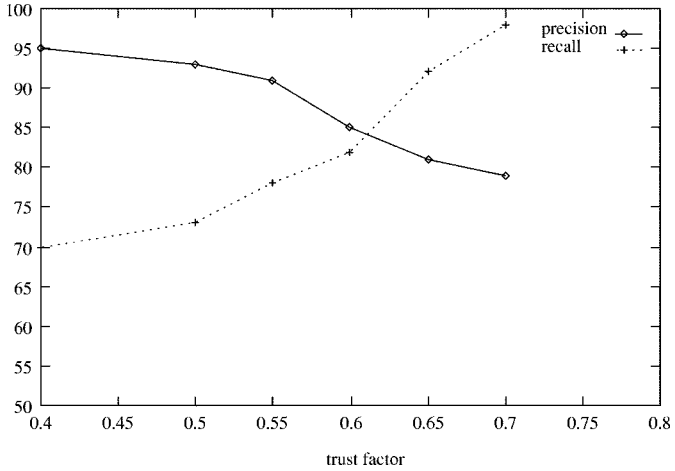

Fig. 16. Recall and precision as a function of the trust factor.

1) Retrieval Accuracy:

a) Quantitative analysis: For a given query, let T be the total number of relevant items available, R_r the number of relevant items retrieved, and T_r the total number of retrieved items. *Precision* $\mathcal{P}$ is defined as R_r/T_r; *recall* $\mathcal{R}$ as R_r/T. Shapes in the test database were clustered by similarity, by a human operator. Therefore, for each shape belonging to one of these clusters, the exact number T of similar shapes in the database is

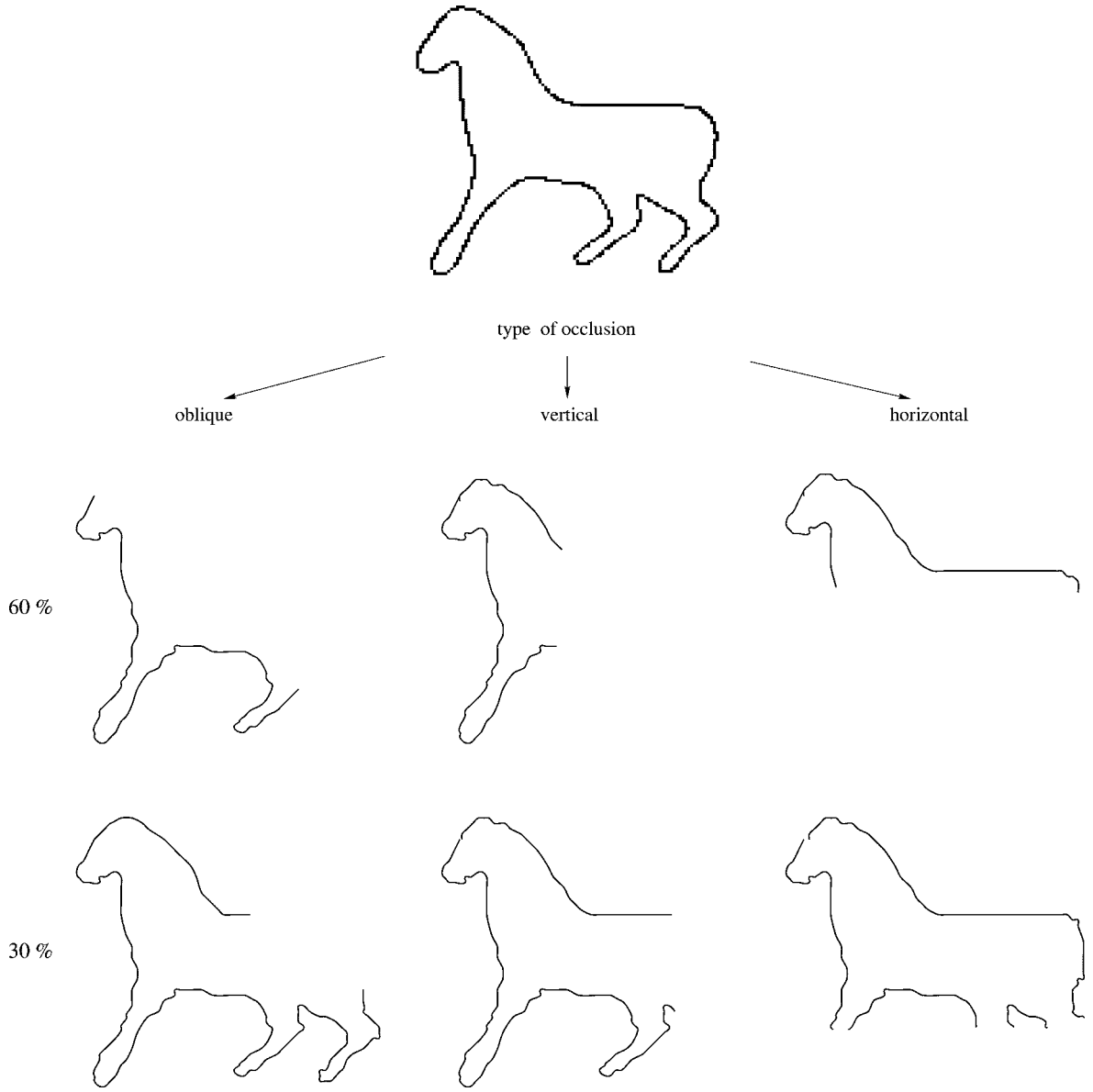

Fig. 17. Occluded shapes used in the partial match experiments. From the original horse, the shape boundary is occluded for 60% and 30% of its length. Occlusions are: oblique, vertical, and horizontal.

known. Using these shapes as query sketches, values of $\mathcal{R}$ and $\mathcal{P}$ as a function of the trust factor are shown in Fig. 16.

To analyze the extent to which the system can cope with occlusions, experiments were carried out, using 20 test shapes. For each shape three distinct degrees of occlusion corresponding to 0% (the original shape), 30%, and 60% of the shape contour were defined. For each of them, three different directions of occlusion were considered: oblique, vertical, and horizontal (see Fig. 17).

Fig. 18 shows average $\mathcal{R}$ and $\mathcal{P}$ as a function of shape occlusion.

b) Qualitative analysis: For this analysis, 22 sample images representing bottles and three reference bottle sketches were considered. The sample images, and the query sketches, are the same as those used in [4]. They are shown respectively, in Figs. 19 and 20.

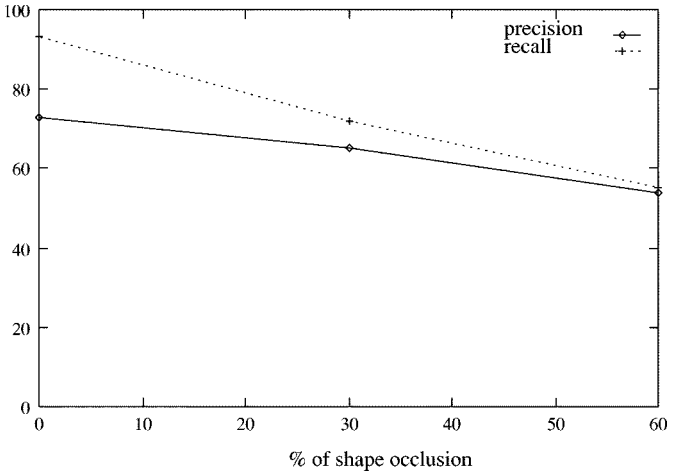

Fig. 18. Recall and precision at different degrees of occlusion.

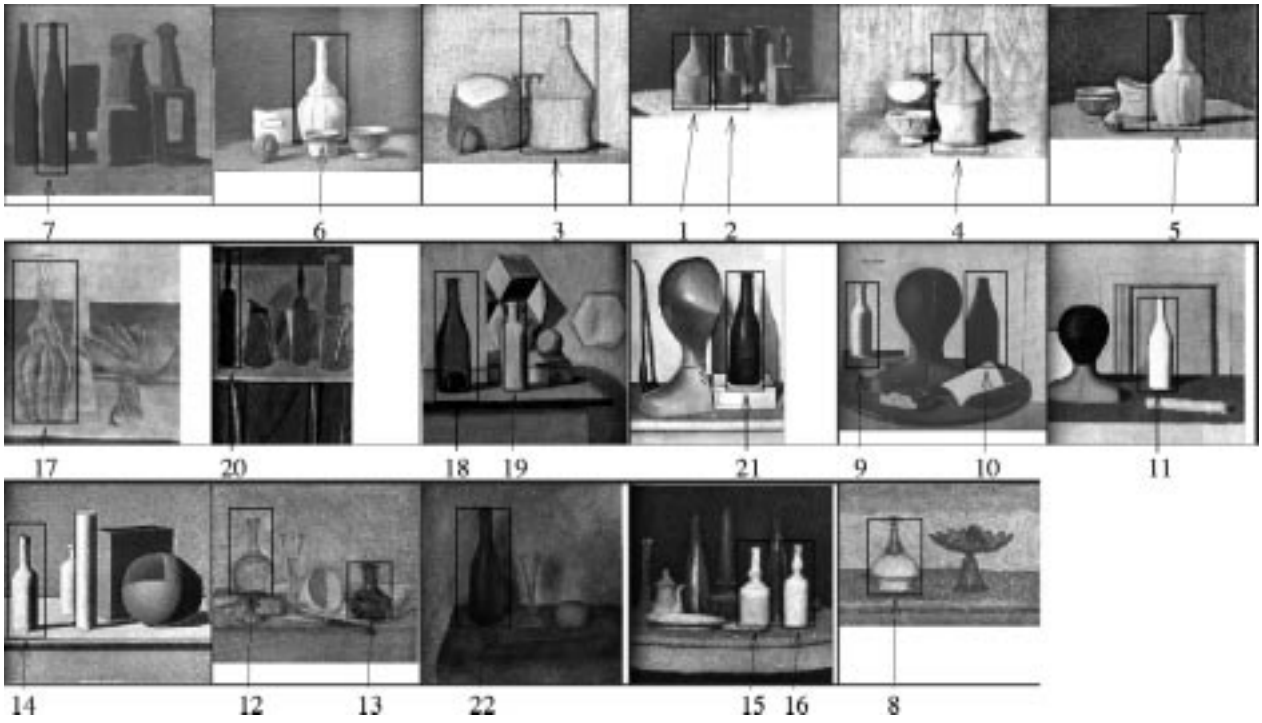

Fig. 19. Test set of 22 images from Morandi's paintings [4].

We collected answers from 42 people, all with a university education. They are workers and students in the art field (23), engineering (12) and literature (7). For each reference sketch, people were asked to assign to each image a score in $[0,1]$ representing the perceived similarity between the query sketch and the shape represented in the image. For each of the 22 sample images three statistical functions $\{p_j(i)\}_{j=1}^{3}$ were derived, representing the ranking of the i-th image with reference to the sketch j. For each function $p_j(i)$ an average value $\bar{p}_j(i)$ and a standard deviation $\sigma_j(i)$ were computed. These represent the average ranking of the ith image for the jth sketch and the agreement about a ranking close to the $\bar{p}_j(i)$-th rank, respectively. Finally, for each image i and rank k, a function $Q_j(i,k)$ with values in $[0,100]$ was defined, representing the percentage of people that ranked the ith image in the kth position with reference to the sketch j.

We considered, for each sample image i and reference sketch j, a window of width $\sigma_j(i)$ centered in the similarity rank $P_j(i)$ given by the token similarity algorithm. The measure of the agreement $S_j(i)$, between the system and human similarity ranking for a reference sketch j and a sample image i, was represented by the sum of the percentage of people who ranked the i-th image in a position between $P_j(i) - \lceil \sigma_j(i)/2 \rceil$ and $P_j(i) + \lceil \sigma_j(i)/2 \rceil$. Therefore, $S_j(i)$ is defined as

$$S_j(i) = \sum_{k=P_j(i)-\lceil \frac{\sigma_j(i)}{2} \rceil}^{P_j(i)+\lceil \frac{\sigma_j(i)}{2} \rceil} \cdot Q_j(i,k).$$

In Fig. 21, plots of $S_j(i)$ as a function of $P_j(i)$ are presented for the proposed approach (TOKEN), the QBIC system (version described in [3]), the QVE [19] system, and the elastic matching approach (ETM) [4]. Only ranks from one to six are shown representing the agreement on the most similar bottles. TOKEN reveals an average agreement of 80% for all three shapes, which shows itself to be better than QBIC (66%) and QVE (56%). It is also close to ETM (88%) (though, this method does not support indexing).

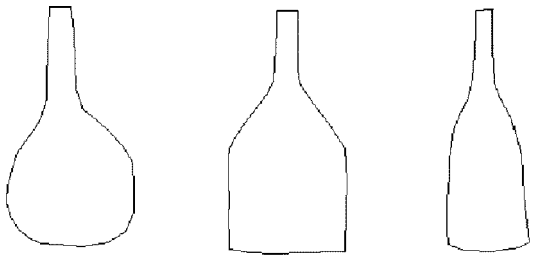

Fig. 20. Sketched templates used in the test.

All four solutions show a minimum of the average agreement function, for the sketch T_3. This is due to the fact that only few elongate bottles were in the test set, which are ranked in the highest positions.

With respect to QBIC and QVE, TOKEN shows the highest values for the absolute minimum and maximum of the agreement function $S(\cdot)$ for all tested cases.

2) Shape Indexing Efficiency: Performance of indexing is measured in reference to two different data sets: *i*) the *real image* data set, which includes about 15 000 tokens from 1637 shapes; *ii*) the *synthetic image* data set, which includes shapes obtained with an automatic generation process.

a) Real image data: We compared the M-tree, R^*-tree, and SS-tree with respect to I/O and CPU requirements to build the index and average searching. Experiments were performed with different node sizes. This allowed the evaluation of per-

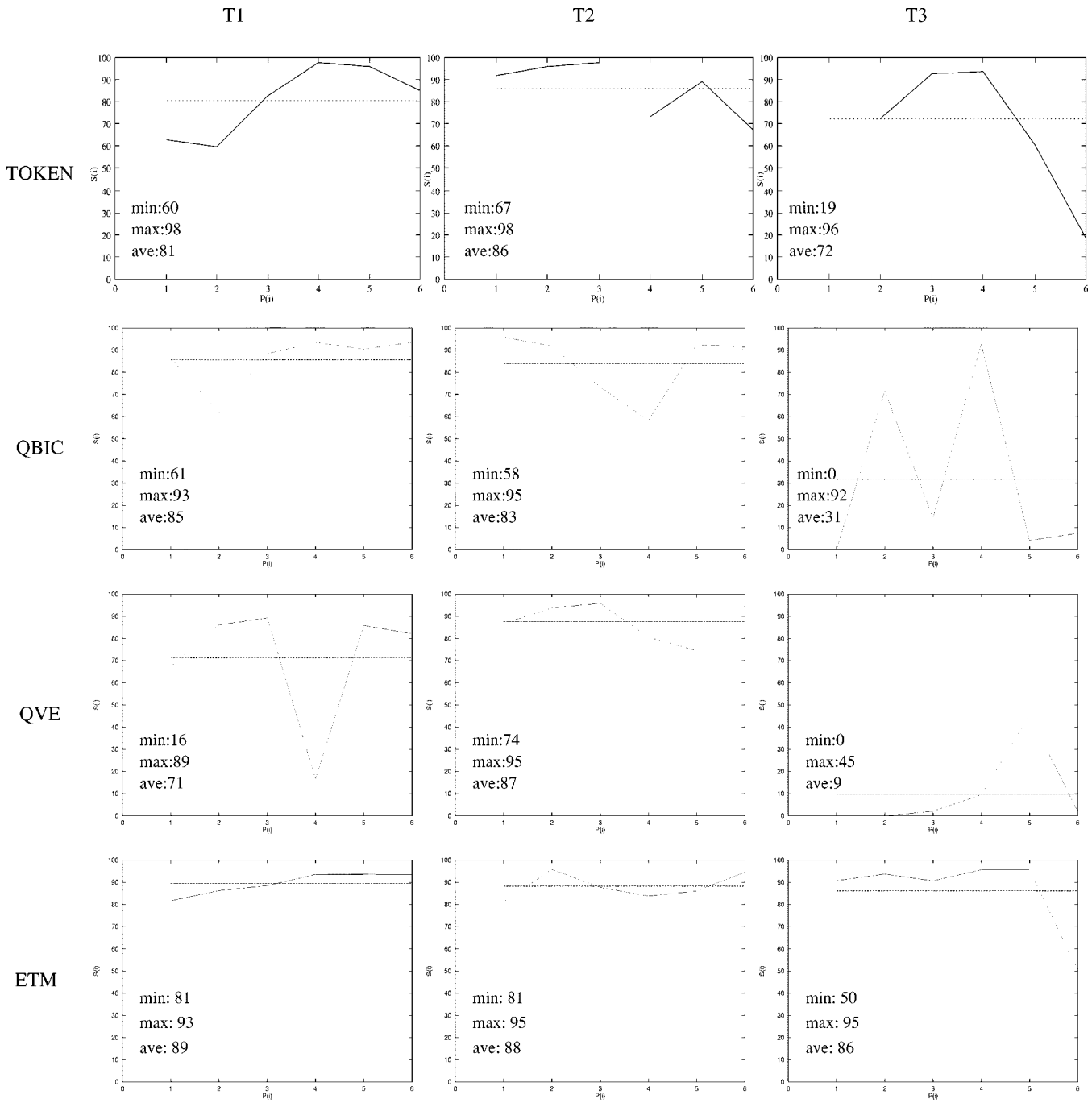

Fig. 21. Comparative results for retrieval effectiveness for the TOKEN based shape description, QBIC, QVE, and ETM. Plots report the agreement of the system in ranking shapes in accordance with the human perception of similarity. QBIC, QVE, and ETM results are those reported in [4].

formance change with the number of entries in internal and leaf nodes, as well as the extent to which different block sizes in secondary memory affect system response.

The M-tree is tested under two node-split policies: *mM-Rad*, and *MLB-Dist*. Plots of Fig. 22, indicate the number of distance computations [Fig. 22(a)] and the number of I/O operations [Fig. 22(b)] as a function of node size in the tree building phase.

As the number of node entries increases, the number of distance computations increases and the number of I/O operations decreases. Fig. 22 shows that M-tree and R^*-tree require a similar number of I/O operations, whereas M-tree outperforms R^*-tree as to the number of distance computations. A small node size reduces the number of distance computations, while the opposite results for I/O operations.

In Fig. 23, plots indicate growth of the index (in terms of number of nodes (a) and tree height (b)), as a function of the node size. The largest tree is the M-tree with the *mM-Rad* split policy while the smallest is the SS-tree.

Average searching is tested for a set of range queries. Range queries are performed using tokens randomly selected from the real data set. Fig. 24 shows the number of leaves that are read for range queries with different search radii. Each plot is obtained by averaging 100 trials. The *mM-Rad* split policy

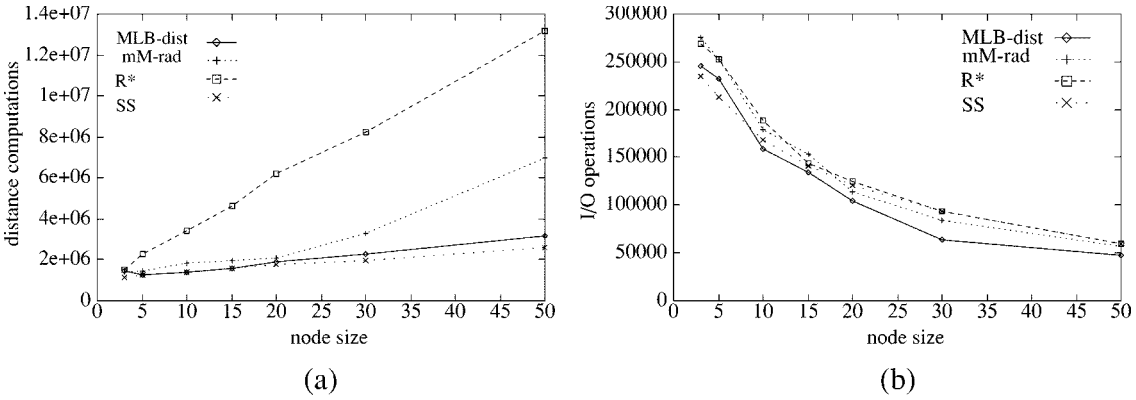

Fig. 22. (a) Number of distance computations (as a function of the node size) to build M-tree, R^*-tree, and SS-tree. (b) Number of I/O operations to build M-tree, R^*-tree, and SS-tree.

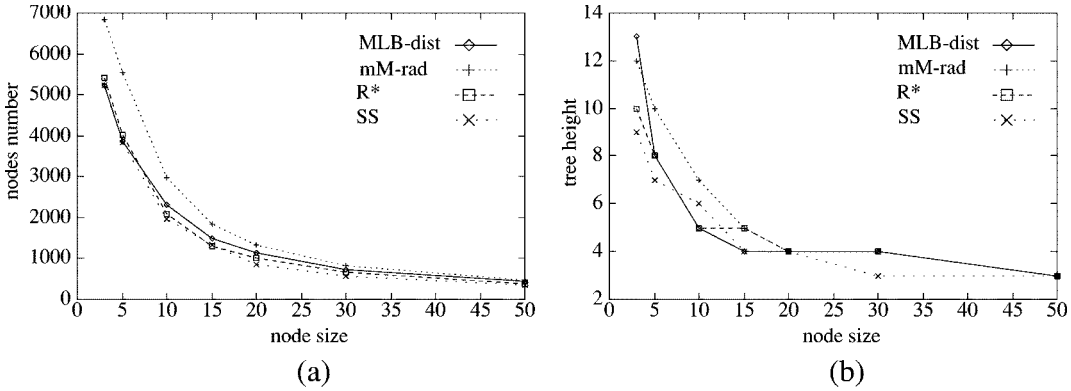

Fig. 23. Number of nodes (a) and height (b) for M-tree, R^*-tree, and SS-tree as a function of the node size.

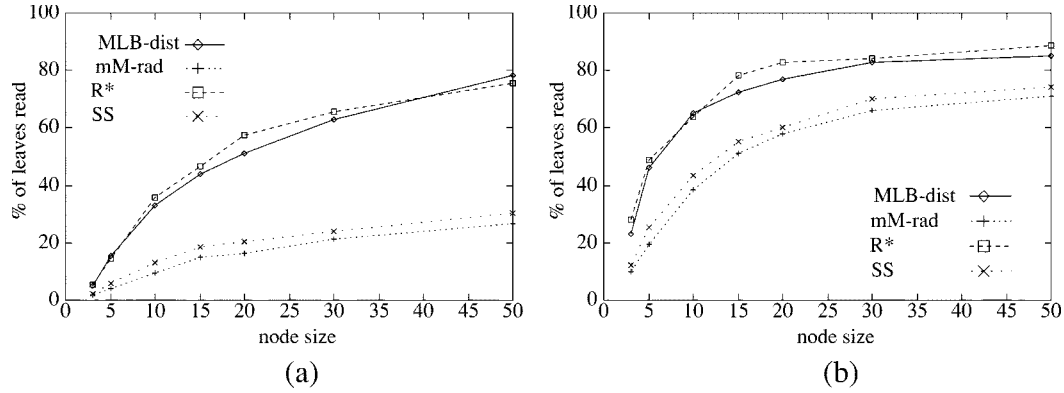

Fig. 24. Number of leaves read for range queries with different search radius. Values are expressed as percentage of the total number of leaves for two different search radius: (a) 0.1; (b) 0.3.

produces larger trees than those generated by *MLB-Dist* but presents the best performance of response time.

The percentage of nodes read with M-tree indexing has been compared to a linear scan of the shape database. From 85% to 90% of blocks read in the secondary memory are saved in the average for each query.

b) Synthetic image data: Tokens in the synthetic image data set are automatically generated using a clustering algorithm, that produces sets of different sizes and data distributions. Two different conditions of uniformly distributed and clustered data are considered. In all these experiments the maximum number of node entries has been set equal to 50.

Fig. 25 shows plots of the tree building cost as a function of data set size. The number of I/O operations and distance computations indicated in (a) and (b), are expressed as average values for each token inserted.

Performance of range queries is analyzed under the assumption that the data distribution is uniform. This is the worst case condition: the large overlapping area between adjacent nodes requires that a large number of internal nodes and leaves is visited. The average number of I/O operations and distance computations, for different data set sizes, is plotted respectively in Fig. 26(a) and (b). Plots display a similar trend for the three index structures. The *mM-Rad* exhibits the best performance.

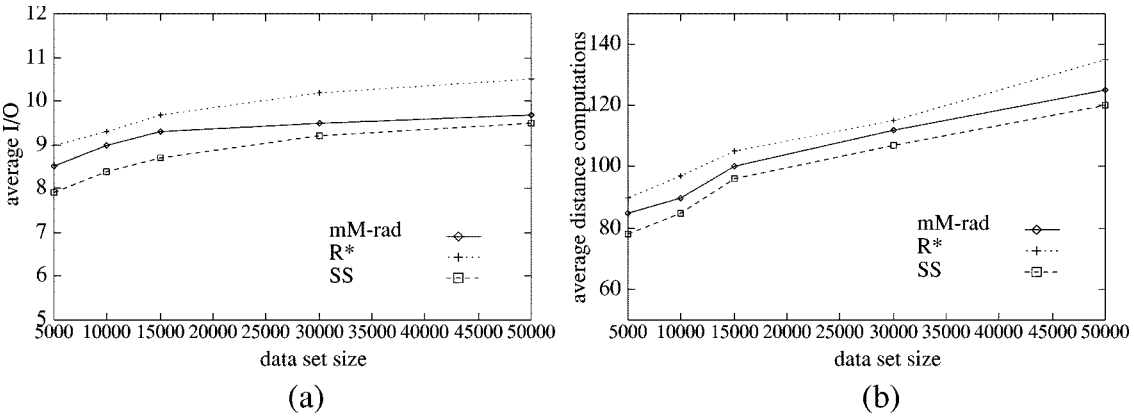

Fig. 25. Tree building cost as a function of the data set size: (a) average number of I/O operations; (b) average number of distance computations.

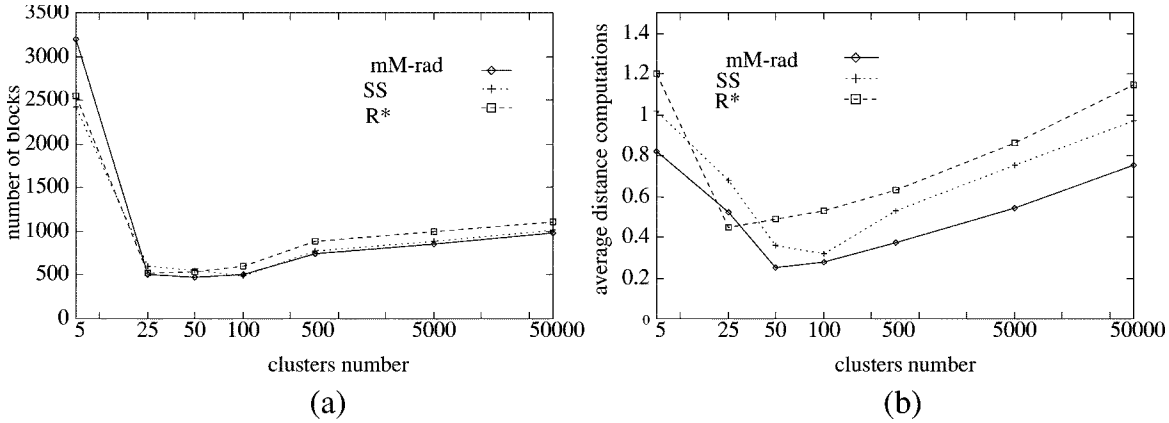

Fig. 26. Cost for a range-query as a function of the data set size: (a) average number of read operations; (b) average number of distance computations.

Seven different data sets of 50 000 tokens are considered to show the extent to which the range query cost depends on data distribution. In the first data set, the 50 000 tokens are clustered into five clusters. This corresponds to an almost uniform data distribution. In the other data sets 25, 50, 100, 500, 5000, and 50 000 clusters (one token for each cluster) are employed. Best performance is obtained for all the indexes when the number of clusters is low. Fig. 26 shows that the M-tree presents the best performance even when the data distribution is changed and the data set size increases.

VI. Conclusion

In this paper, we proposed retrieval by shape similarity by combining correspondence with human perception and effective indexing.

Two distance measures are defined which model respectively token similarity (according to a metric distance which is combined with the M-tree index) and shape similarity (according to a nonmetric combination of token distances).

Experimental results with a prototype retrieval system have been presented showing retrieval effectiveness and indexing efficiency. Shape retrieval based on token representation has shown to be robust in the presence of partially occluded objects, scaling and rotation. Test results show that the use of the M-tree index permits to outperform traditional indexing based on the R^*-tree and SS-tree structures.

References

[1] E. Binaghi, I. Gagliardi, and R. Schettini, "Image retrieval using fuzzy evaluation of color similarity," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 8, no. 4, pp. 945–968, May 1994.

[2] A. Del Bimbo, M. Mugnaini, P. Pala, and F. Turco, "Visual querying by color perceptive regions," *Pattern Recognit.*, vol. 31, pp. 1241–1253, 1998.

[3] C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz, and R. Barber, "The Qbic Project: Efficient and Effective Querying by Image Content," IBM Res. Div. Almaden Res. Center, Res. Rep. 9453, Aug. 1993.

[4] A. Del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 121–132, Feb. 1997.

[5] I.-J. Lin, A. Vetro, H. Sun, and S.-Y. Kung, "Shape representation and comparison using voronoi order skeletons and dag-ordered trees," presented at the Int. Workshop on Very Low Bit-Rate Video Coding, Oct. 1999.

[6] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image databases," presented at the SPIE Storage and Retrieval for Image and Video Database, Feb. 1994.

[7] L. G. Shapiro and R. M. Haralick, "Decomposition of two-dimensional shapes by graph theoretic clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 10–20, 1979.

[8] ——, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 5, pp. 504–519, Oct. 1981.

[9] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness—Wold features for image modeling and retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 722–733, July 1996.

[10] R. Picard, A society of models for video and image libraries, in MIT Media Lab Perceptual Computing Section Tech. Rep., Cambridge, MA, no. 360, 1995.

[11] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 460–473, Apr. 1976.

[12] S. K. Chang, Q. Y. Shi, and C. W. Yan, "Iconic indexing by 2-d strings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 413–427, July 1987.
[13] M. J. Egenhofer and R. Franzosa, "Point-set topological spatial relations," *Int. J. Geogr. Inform. Syst.*, vol. 9, no. 2, 1992.
[14] H. Samet, "Hierarchical representations of collections of small rectangles," *ACM Comput. Surv.*, vol. 20, no. 4, pp. 271–309, Dec. 1988.
[15] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r^*-tree: An efficient and robust access method for points and rectangles," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, May 1990, pp. 322–331.
[16] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. 1984 ACM SIGMOD Conf. Management of Data*, June 1984, pp. 47–57.
[17] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The r^+-tree: A dynamic index for multi-dimensional objects," in *Proc. 13th VLDB Int. Conf.*, Sept. 1987, pp. 507–518.
[18] S. Santini and R. Jain, "Similarity measures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 871–883, Sept. 1999.
[19] K. Hirata and T. Kato, "Query by visual example: Content-based image retrieval," *Advances in Database Technology—EDBT'92*, vol. 580, 1992.
[20] S. Sclaroff, "Deformable prototypes for encoding shape categories in image databases," *Pattern Recognit.*, vol. 30, no. 4, pp. 627–642, Apr. 1997.
[21] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognit.*, vol. 29, no. 8, pp. 1233–1244, Aug. 1996.
[22] ——, "Shape-based retrieval: A case study with trademark image database," *Pattern Recognit.*, vol. 31, no. 9, pp. 1369–1390, Sept. 1998.
[23] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," presented at the Int. Workshop on Image Databases and Multi-Media Search, Aug. 1996, pp. 35–42.
[24] S. Matusiak, M. Daoudi, T. Blu, and O. Avaro, "Sketch-based images database retrieval," in *Proc. Workshop on Multimedia Information Systems*, Sept. 1998, pp. 185–191.
[25] E. Petrakis and E. Milios, "Shape retrieval based on dynamic programming," *IEEE Trans. Image Processing*, vol. 9, pp. 141–147, Jan. 2000.
[26] ——, "Efficient retrieval by shape content," *IEEE Int. Conf. Multimedia and Computing Systems*, pp. 616–621, June 1999.
[27] W. I. Grosky and R. Mehrotra, "Index-based object recognition in pictorial data management," *Comput. Vis. Graph. Image Process.*, vol. 52, pp. 416–436, 1990.
[28] W. I. Grosky, P. Neo, and R. Mehrotra, "A pictorial index mechanism for model-based matching," *Data Knowl. Eng.*, vol. 8, pp. 309–327, 1992.
[29] R. Mehrotra and J. E. Gary, "Similar-shape retrieval in shape data management," *IEEE Computer*, pp. 57–62, Sept. 1995.
[30] ——, "Shape matching utilizing indexed hypothesis generation and testing," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 70–77, Jan. 1989.
[31] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proc. Int. Conf. VLDB*, 1997, pp. 522–525.
[32] P. Ciaccia, M. Patella, F. Rabitti, and P. Zezula, "Indexing metric spaces with m-tree," in *SEBD'97*, 1997, pp. 67–86.
[33] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, pp. 183–193, 1954.
[34] F. G. Ashby and N. A. Perrin, "Toward a unified theory of similarity and recognition," *Psychol. Rev.*, vol. 95, no. 1, pp. 124–150, 1988.
[35] A. S. Householder and H. D. Landahl, "M," in *Mathematical Biophysics of the Central Nervous System*. New York: Principia, 1945.
[36] A. Twersky and I. Gati, "Similarity, separability, and the triangle inequality," *Psychol. Rev.*, vol. 89, pp. 123–154, 1982.
[37] D. A. White and R. Jain, "Similarity indexing with the ss-tree," in *Proc. 12th Int. Conf. Data Engineering*, Feb. 1996, pp. 516–523.

Stefano Berretti received the doctoral degree in electronics engineering in 1997 from the Università degli Studi di Firenze, Italy, where he is currently pursuing the Ph.D. degree in information technology and communications. His current research interest is mainly focused on content modeling and retrieval for image and video databases.

Alberto Del Bimbo (M'90) is Full Professor and Director of the Department of Sistemi e Informatica at the Università degli Studi di Firenze, Italy. He is also the Director of the Master in Multimedia at the same University. His scientific interests and activity have addressed the subject of image technology and multimedia, with particular reference to object recognition and image sequence analysis, content-based retrieval for image and video databases, visual languages and advanced man-machine interaction. He Del Bimbo is the author of over 150 publications, having appeared in the most distinguished international journals and conference proceedings and is the author of the monograph *Visual Information Retrieval* (San Mateo, CA: Morgan Kaufman 1999). He has also been the Guest Editor of several special issues of international journals and the chairman of several conferences in the field of image processing, image databases and multimedia. He is an associate editor of *Pattern Recognition, Journal of Visual Languages and Computing*, and *Multimedia Tools and Applications Journal.*

Prof. Del Bimbo is an IAPR fellow and presently a member of the steering committee of IEEE ICME International Conference on Multimedia and Expo and of the VISUAL conference series. From 1996 to 2000, he was the President of the Italian Chapter of IAPR, the International Association for Pattern Recognition. Since 1999, he has been a member of the IEEE Publications Board. He presently serves as Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Pietro Pala (S'93–M'97) graduated in electronic engineering from the Università di Firenze, Italy, in 1994, where he received the Ph.D. in information science in 1998.

Currently, he is Assistant Professor in the Dipartimento di Sistemi e Informatica, University of Florence. His current research interests include pattern recognition, image and video retrieval by content, and related applications.