Invariant-Based Shape Retrieval in Pictorial Databases

Michael Kliot and Ehud Rivlin

Computer Science Department, Technion - Israel Institute of Technology, Haifa 32000, Israel. Tel.: 972-4-8294304. E-Mail: {michk,ehudr}@cs.technion.ac.il

Abstract. One of the strongest cues for retrieval of content information from images is shape. However, due to the wide range of transformations that an object might undergo, this is also the most difficult one to handle. It seems that shape retrieval is one of the major barriers nowadays on the way of image databases to become commonly used. Common approaches use global attributes (Faloutsos et al. [1]), feature points (Pentland et al. [2]), histograms (Jain and Vailaya [3]), or physical models of deformations (Del Bimbo and Pala [4]). We present an approach for shape retrieval from pictorial databases which is based on invariant features of the image. In particular we use a combination of semi-local multi-valued invariant signatures and global features. Spatial relations and global properties are used to eliminate non-relevant images before similarity is computed. Common approaches usually don't handle viewpoint transformations more complex than similarity and require the full shape in order to compute image features. The advantages of the proposed approach are its ability to handle images distorted by different viewpoint transformations, its ability to retrieve images even in situations in which part of the shape is missing (i.e., in case of occlusion or sketch-based queries), and its ability to support efficient indexing.
We have implemented our approach in a heterogeneous database having a SQL-like user interface augmented with sketch-based queries. The system is built on top of a commercial database system, and can be activated from the Web. We present experimental results demonstrating the effectiveness of the proposed approach.

1 Introduction

The area of pictorial databases has become recently a subject of extensive research. Large databases of images are used in different applications including multimedia systems, medical imaging systems, documents systems and so forth. The most common form of retrieval is based on image content. For most of the applications content is defined in terms of three basic components: color, shape and texture. This paper concentrates on shape-based retrieval.

Various approaches for shape retrieval differ by the features used for representing the image and for indexing, as by the different ways in which the pictorial information is organized to allow effective retrieval. Several works use *global numerical attributes* for indexing and retrieval shapes [1, 5, 6]. The distance between two images is obtained as the weighted Euclidean distance in

a n-dimensional space of features. Another form of shape description is based on *feature points*, like corners and high-curvature points, detected from the image [2, 7, 8]. Other approaches use symbolic string representation of the images [9], histogram of the edge directions of the image [3], or elastic matching of user-drawn sketches [4].

Common approaches, while providing good accuracy and efficiency, still suffer from instability as a result of viewpoint transformations. An additional disadvantage is the requirement of the full, unoccluded shape in order to compute the needed image features. Some approaches (e.g. [4]) try to overcome the different deformations by using physical models (e.g elastic deformation energy). As they do not address the general geometric transformation it will be hard for them to handle the full spectrum of the transformations, as well as situations where part of the shape is missing. In order to handle such situations *local* invariant features should be used. The most popular and well-studied kind of local invariants are the *differential* invariants, i.e. invariants that are based on derivatives of different order. The main disadvantage of this kind of invariants is their sensitivity to noise, resulting from their differential character. Hybrid, or semi-differential invariants try to combine the advantages of the global and local approaches. Moons et al. [10] present a general theory of local semi-differential invariants. Bruckstein et al. [11, 12] use locally applied global geometric invariants.

In this work we attempt to overcome the limitations of the common approaches, namely a restriction of the viewing transformation to similarity, and a requirement for a complete, unoccluded input image. We use features, invariant for wide range of viewpoint transformations, including affine and perspective. Shape similarity is measured using semi-local multivalued invariant signatures, that allow to handle situations in which part of the shape information is missing (i.e occlusion or sketch-based retrieval). To efficiently handle such cases we introduce a data structure, the *containment tree* , that exploits the topological structure of the image. This structure supports indexing mostly in sketch-based retrieval. In this paper we present a series of experiments done on a heterogeneous database system which we build to test our approach. The system allows to query images by logical or shape descriptions (query by example), or combination of both. It has an interface to the Web, providing a convenient user interface.

The paper is organized as follows. In Sections 2 and 3 we present our approach for invariant-based retrieval. Section 2 presents the machinery needed for indexing and retrieval with full and partial shape information, while Section 3 describes the query processing itself. Section 4 presents variety of results using our approach for retrieving images from three different databases. Section 5 contains the concluding remarks.

2 Indexing and invariants

The purpose of indexing is to support fast retrieval from the database. Usually, two requirements are imposed on features used for indexing: the features should

allow for substantial reduction in the number of candidates from the database, and a feature-based distance function should exist which supports ranking of images according to their distance from the input image.

As primitives for shape description we use various geometric entities. We exploit these entities for filtering the database while searching for a candidates set of images which answer a query. Features on which the description is based include various properties varying from the number of the different geometric entities, their dimensions and positions, to some topological relationships. Our approach emphasizes the use of *geometric invariants* which provide a good mean for indexing while supporting ranking. Different features are used for extraction of descriptors that are invariant under various transformations. Note that different applications may allow for different kinds of invariants. In applications that use images of trademarks, for example, similarity invariants are usually enough. For some other applications, however, a wider set of transformations, affine or projective, may be required. The applicability of a certain feature for indexing can be adjusted as well to the class of transformations derived from the application at hand.

Our invariant representation of objects is based mainly on *semi-local multivalued invariant signatures*. These signatures provide a measure of distance between two curves, which allows us to rank images according to their distance from the input image. Below we describe the extraction and matching of such invariant signatures.

2.1 Semi-local multivalued invariant signatures

Invariant signature is a function of a curve, calculated pointwise, and invariant for a given set of transformations. Invariant signatures can be used for recognition of planar curves. The important property of invariant signatures is their applicability in situations where the input curve is partially occluded. Having an invariant signature, curve matching reduces to matching signatures.

In order to calculate an invariant signature, one should find a local invariant for the given set of transformations, and apply it at each point of the curve. Our approach follows the one presented in [11, 12]. First, the curve (object boundary), given in arbitrary parameterization, is reparameterized invariantly, using the lowest possible order of derivatives. Formally, for a given planar shape transformation set $\mathbf{T}_\psi : \mathbf{R}^2 \to \mathbf{R}^2$, and for a given curve $\mathbf{P}(t)$, having an arbitrary parameterization, the target is to find an *invariant reparameterization* $\mathbf{P}(\tau)$ so that if $\mathbf{P}(t)$ and $\tilde{\mathbf{P}}(\tilde{t})$ are related via $\tilde{\mathbf{P}}(\tilde{\tau}) = \mathbf{T}_\psi[\mathbf{P}(t(\tilde{t}))]$ then $\tilde{\mathbf{P}}(\tilde{\tau}) = \mathbf{T}_\psi[\mathbf{P}(\tau + \tau_0)]$. In other words, the reparameterization is invariant if the corresponding points have the corresponding parameter value, up to some constant cyclic shift.

The equations for invariant reparameterization are obtained using the differential properties of the viewpoint transformations. Let's define $K^{n,m}[x, y|t] \triangleq x^{(n)}y^{(m)} - x^{(m)}y^{(n)}$. For the similarity transformation $\mathbf{T}_\psi(\mathbf{u}) = \tilde{\mathbf{u}} = \alpha \mathbf{U}_\omega \mathbf{u} + \mathbf{v}$, where $\mathbf{U}_\omega$ is a rotation matrix, we obtain that $d\tau = \frac{K^{1,2}[x,y|t]}{|\dot{\mathbf{P}}|^2} dt$ is an invariant, generalized arc length, reparameterization.

For the affine transformation $\mathbf{T}_\psi(\mathbf{u}) = \tilde{\mathbf{u}} = \mathbf{A}u + \mathbf{v}$ we obtain that for $d\tau^* = \left|K^{1,2}[x,y|t]\right|^{\frac{1}{3}} dt, d\tilde{\tau}^* = \left|K^{1,2}[\tilde{x},\tilde{y}|\tilde{t}]\right|^{\frac{1}{3}} d\tilde{t}$ we have $\tilde{\tau}^* = |det\mathbf{A}|^{\frac{1}{3}}\,\tau^* + \tau_0^*$. This gives us linear scaled invariant reparameterization. In the particular case $|det\mathbf{A}| = 1$, the reparameterization is absolutely invariant. Exploring further the properties of affine transformation, we obtain that $d\tau = \left|\frac{d}{d\tau^*}\left|\frac{K^{2,4}[x,y|\tau^*]}{K^{2,3}[x,y|\tau^*]^{3/2}}\right|\right| d\tau^*$ is a non-scaled affine invariant reparameterization.

The implementation of the reparameterization process intends to reduce the influence of the noise. First, the contour is transferred to a parametric form by B-spline approximation. Since we want the spline value at a certain point to depend on the coordinates of the contour points in a sufficiently large region, so that the influence of a small error in the edge detection is diminished, we adopted the technique used in [13]. The solution subsamples the contour, computes the spline values and averages the results for different positions of samples on the contour. So, the first sampling set consists of each l-th point on the contour, the points next to the chosen ones form the second set, etc. Thus, no information is lost, because each point participates in some sample. After the B-spline representation of the curve is obtained, the reparameterization is straightforward.

The computation of the semi-local signature is based on geometric features, that stay invariant under the viewpoint transformation. Under similarity transformation, ratios of lengths, ratios of areas and angles stay invariant. Thus, we can use those invariants locally to generate signature functions of various types. After the invariant reparameterization is completed, the two curves $\mathbf{P}(\tau)$ and $\tilde{\mathbf{P}}(\tilde{\tau})$ are related by $\tilde{\mathbf{P}}(\tilde{\tau}) = \mathbf{T}_\psi[\mathbf{P}(\tilde{\tau} + \tilde{\tau}_0)]$. This equation shows that one can compute the ratio of lengths, or the angle between, the segments defined by $(\mathbf{P}(\tau - s_B), \mathbf{P}(\tau))$ and $(\mathbf{P}(\tau), \mathbf{P}(\tau + s_F))$ for a priori chosen values s_B and s_F (locality parameters). Thus $\frac{\delta[\mathbf{P}(\tau),\mathbf{P}(\tau+s_F)]}{\delta[\mathbf{P}(\tau),\mathbf{P}(\tau-s_B)]}$ and the angle formed by the points as functions of τ are invariant signature functions (see Fig. 1(left)).

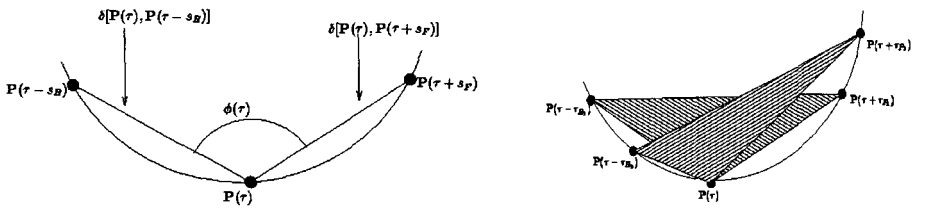


Fig. 1. Semi-local invariants. On the left - invariants for similarity transformation. On the right - invariants for the affine transformation.

Under affine transformation, areas are uniformly scaled by det $\mathbf{A}$. This fact implies that the ratio of areas is affine invariant. Assuming that an invariant

parameterization was already obtained for the curve, we can choose four values for τ: $\tau_{B_1}, \tau_{B_2}, \tau_{F_1}, \tau_{F_2}$, and calculate at each point $\mathbf{P}(\tau)$ the ratio of areas, $\frac{Area_{\triangle}(\mathbf{P}_{B_1}, \mathbf{P}(\tau), \mathbf{P}_{F_1})}{Area_{\triangle}(\mathbf{P}_{B_2}, \mathbf{P}(\tau), \mathbf{P}_{F_2})}$, where $\mathbf{P}_{B_1}, \mathbf{P}_{F_1}, \mathbf{P}_{B_2}, \mathbf{P}_{F_2}$ are defined using locality parameters as for similarity case. This quantity is an invariant signature as a function of the invariant "arc length" τ (see Fig. 1(right)).

If the locality parameters set is let to be free parameters rather than setting them in advance, we obtain a whole range of invariants at each point rather than a single value (Bruckstein et al. [12]). The signature functions for curves become signature vectors or even continuum of values, i.e., surfaces or hypersurfaces. Matching them is less sensitive to peculiarities that may exist at some fixed pre-set value of the locality parameters.

It is important to note that since, in the general case, there is no correspondence between the initial points of the curves, the corresponding points on the curves have, after the invariant reparameterization, the same parameter value up to unknown constant cyclic shift value. This fact adds to the complexity of the matching process.

We designed an effective automatic matching procedure, that can handle images distorted by affine and similarity transformations, and situations in which image is partially occluded. To match two signatures, we map them to the matrices with the number of rows equal to the number of signatures in the multivalued signature, and the number of columns equal to the number of the samples. In order to estimate the unknown value of relative cyclic shift, we exploit reference points. As a reference point we use the signature extrema. Minimizing the difference between the matrices over all checked shift values provides a measure for the distance between the signatures.

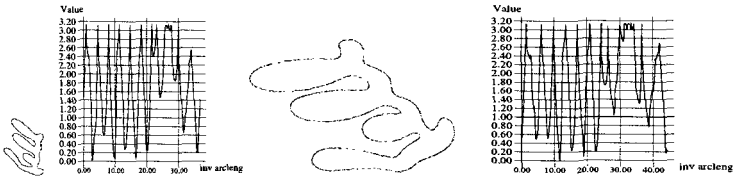
Values of the invariant perimeters of the same curves, up to transformation, may differ due to possible error in the edge detection process (see Fig. 2). We map the signatures having different invariant perimeters to matrices with the same number of columns and calculate the difference, while taking the difference between the invariant perimeters into account. The differences between two multivalued signatures is given by

$$Diff(\mathbf{S}_{inp}, \mathbf{S}_{lib}) = \min_{sh \in Sh} \frac{\sum_{i=0}^{L-1} \sum_{j=0}^{N-1} (M_{inp}(i, \tilde{j}) - M_{lib}(i, j))^2}{L \times N} \times \left(\frac{max(p1, p2)}{min(p1, p2)}\right)^{\mathbf{p}}$$

where $\mathbf{N}$ is the number of signatures in the multivalued signature, $\mathbf{L}$ is the number of samples, $\mathbf{s1}$ and $\mathbf{s2}$ are the compared signatures, $\tilde{\mathbf{j}} = (j + sh) \, mod \, Lev$, $\mathbf{Sh}$ is the set of all the checked shift values, $\mathbf{p1}$ and $\mathbf{p2}$ are invariant perimeters, and $\mathbf{p}$ is a positive constant.

In order to speed up the matching process, we exploit the signature histogram. This feature is independent on the starting point. The low-dimensional histogram allows for using of efficient data structures, like R-trees, for the database organization. First, signature histograms are compared. Only items close enough to the input curve are matched using the whole signature.

The total difference between the images is taken to be the average of the distances between the corresponding curves:



(a) (b) (c) (d)

Fig. 2. Reparameterization inaccuracy. (a) & (b) Library image and its invariant signature. (c) & (d) The transformed library image and its invariant signature. Note the differentce of the domains of (b) and (d).

$$d(im_1, im_2) = \frac{\sum_{\{Maximal\ curves\}} \sqrt{diff(c1_i, c2_i)}}{N},$$

where **N** is the number of compared curves, and $\mathbf{c1_i}$ and $\mathbf{c2_i}$ are corresponding curves of $\mathbf{im_1}$ and $\mathbf{im_2}$, respectively. In order to find pairs of corresponding curves we sort the curves of each image with respect to some invariant criterion. In case of similarity transformation, the curve's area serves as an ordering criterion. Curves that can not be sorted by area are ordered by their perimeters. Under affine transformation, perimeters can not be used any longer; however, the ratio of areas is still invariant and areas can used for ordering.

The following example illustrates extraction and matching of invariant signatures. The input image (Figure 3(a)) is an affine transformed version of the library image (Figure 3(b)). The extracted contours are presented on Fig. 3(c). The size of the biggest contour is about 300 pixels. The contours were ordered (by area) and invariant signatures were extracted. We use the invariant signature described above - the ratio of areas, with 4 parameters, each having 5 sample points. While for similarity invariants we used the angle, which is relatively stable characteristic, the ratio of areas can suffer from numerical instability. If the points $\mathbf{P}(\tau - \tau_{B_2})$, $\mathbf{P}(\tau)$ and $\mathbf{P}(\tau + \tau_{F_2})$ (see Fig.1(right)) become near collinear, the signature becomes unstable. Figure 4 presents the affine invariant signature for curve No. 5 from Fig. 3. The instability points force us to put a threshold on the signature values. While the resulting signature (see Fig. 4(b)) can still be used for the matching process, some of the information apparently is lost. Another possibility is to exchange the values of τ_{B_1} with τ_{B_2}, and τ_{F_1} with τ_{F_2}, respectively, and use the signature $\frac{1}{f(t)}$ instead of $f(t)$ (Figure 4(c)) .

The matching results for the trademarks presented on Fig. 3 are presented on Fig. 5. Each grey-level map, consisting of five strips, corresponds to the multivalued signature, while each strip corresponds to the singular signature. Each pair of the grey-level maps presents the signatures of the input (above) and library (below) contours at the shift position giving the best match (the numbering of the contours is as in Fig. 3(c)). Good match is achieved for all the curves. Note, that the signatures of curves No. 4 and 5 are very similar as the curves are almost identical when going under affine transformation.

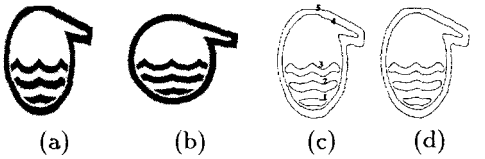


Fig. 3. (a) The input image (b) The source library image (c) The contours of the input image (numbered) (d) The fitted contours

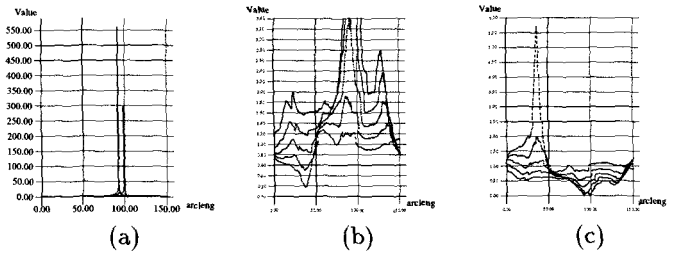


Fig. 4. Numerical instability of an affine invariant signature. (a) – Multivalued affine invariant signature $f(t)$ for curve No. 5 from Fig. 3. The signature is based on the areas ratio. (b) – $f(t)$ after putting a threshold. (c) – The signature $\frac{1}{f(t)}$.

2.2 Indexing while some shape information is missing

One of the main advantages of the proposed approach is its ability to handle situations in which part of shape information is missing. We handle separately two different cases: partial occlusion and user-generated sketch-based queries. In both cases the indexing is based on the same geometric features which are used regularly. However, the indexing algorithms will be used in accordance with the appropriate context.

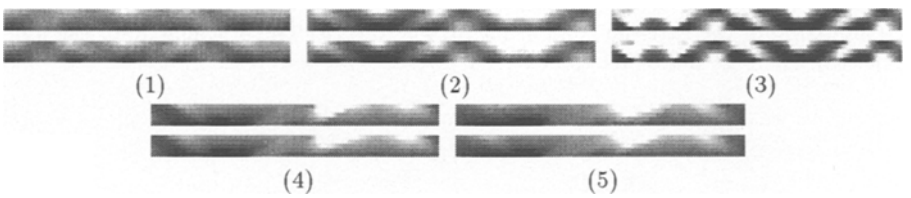


Fig. 5. The best match for each curve presented in Fig. 3. Grey-level maps are used for signatures. The curves are numbered as in Fig. 3. Good match is achieved for all the curves.

Partial occlusion. Partially occluded images are treated in the same way as regular images. Since both the reparameterization and signature extraction stages are absolutely local, the signature values for the part of the contour should match the values computed for the whole curve. The only difference is the need for *partial matching*, i.e. matching of the occluded contour to the part of the library curve. If the input curve is occluded, we map it to a matrix with number of columns proportional to the width of the signature domain. The domain of the multivalued signature is the minimal domain of its components (for open curve, the bigger the locality parameter, the smaller the signature domain). Thus, the number of columns in the matrix is

$$\tilde{Lev} = Lev * \min_{i=0}^{Num-1}\{||domain(S_i)||\}/Per_{lib},$$

where $\mathbf{Per_{lib}}$ is the invariant perimeter of the library signature and **Lev** is the number of columns of the library matrix. The matrix of the occluded signature is then matched to a part of the library matrix.

Figure 6(a) is a rotated, scaled, and occluded version of the library image presented in Fig. 6(b). We use the angle $\phi(\tau)$, formed by the points $\mathbf{P}(\tau - s_B), \mathbf{P}(\tau)$ and $\mathbf{P}(\tau + s_F)$, as an invariant signature (as in Fig. 1(left)). Figure 6(lower) presents the best matchings between the input and the library curves. The signature domain is reduced as a result of the occlusion.

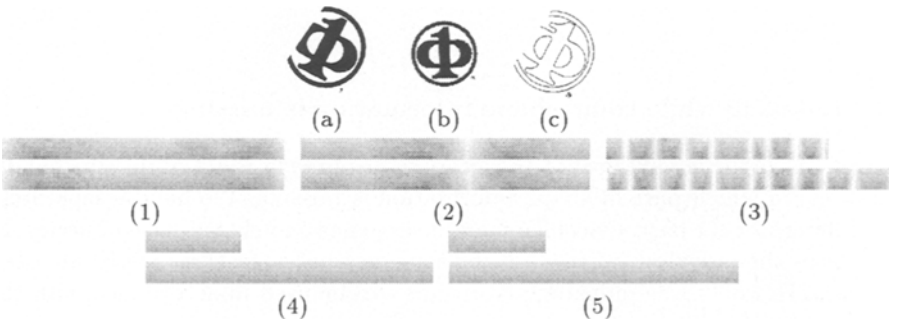


Fig. 6. Above: (a) The input image (occluded) (b) The source library image (c) The contours of the input image. Below: Best matches for the curves presented in (a). (1) and (2) Best matches for the signatures of the inner contours of the Φ letter. (3) The best matches for the external contour of the Φ letter. (4) The best match for the inner circle. (5) The best match for the external circle.

Sketch-based queries. We consider sketch-based retrieval as retrieval that uses the gross shape structure, while the fine details can be omitted. For sketch-based retrieval we exploit a topological invariant which we define here, that is based on the following simple and effective invariant property:

Given the same contour decomposition for an image P and a transformed image $\mathbf{T}_\psi(P)$*, the contour* C_1 *of P resides inside the contour* C_2 *if and only if the same relation holds for the corresponding contours in the transformed image. This holds for any projective transformation* $\mathbf{T}_\psi$*.*

In other words, the property that one contour is an inner contour of another is a projective invariant given the same contour representation. This property allows us to represent images exploiting the relations of internal-external between contours. Each image is represented as a tree which we term the *containment tree* . The vertices of the tree are curves. For two curves, C_1 and C_2, the edge $(C_1 \rightarrow C_2)$ exists if C_2 is inside C_1, and there is no C_3 such that C_2 is inside C_3 and C_3 is inside C_1. The root of the tree is a "dummy" contour which includes all the contours. This representation can easily be obtained after curves extraction (see Figure 7). The representation is unique up to the order between the vertices on the same level. Thus tree-matching algorithms can be used to compare the representation of the input image with the library images. This problem of tree matching has a polynomial solution [14]. The discrimination power of the containment tree is illustrated on Fig. 8.

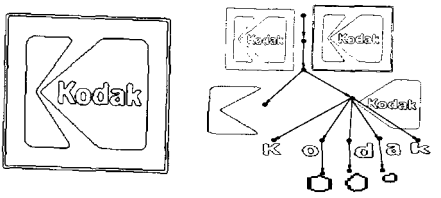


Fig. 7. The input image after curves extraction is presented on the left. The containment tree , representing the containment relationships between the curves is presented on the right. For each subtree, the corresponding curve and all its internal curves are presented.

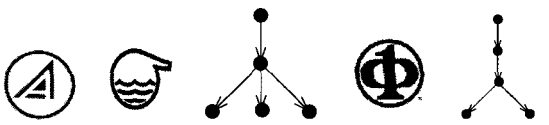


Fig. 8. The discrimination power of a containment tree . (a) & (b) Two images having the same containment tree . (c) The containment tree of (a). (d) The image has the same number of curves as in (a), but a different containment tree . (e) The containment tree of (d).

We exploit the containment tree for sketch-based retrieval. The matching algorithm is presented as Alg. 1. The algorithm requires that for each contour of the specified sketch all its sub-contours should be either omitted or specified. In this way we allow for queries that uses gross structure while omitting details (from some level of the tree).

Algorithm 1
Sketch matching.
Given: An input sketch S, and a database image I,

1. Build a containment tree T_1 for S, and a containment tree T_2 for I. In each containment tree the contours on the same level are ordered in descending order using the area – affine invariant geometric criterion.
2. Check if T_1 matches T_2.

The match is defined in the following manner:
The containment tree T_1 for the sketch S matches the containment tree T_2 for the image I if:

1. T_1 is a leaf **or**
2. The number of subtrees of T_1 equals that of T_2 and the corresponding sons of T_1 and T_2 match.

A flexible version of this algorithm which allows to omit small sub-contours of the specified sketch is presented as Alg. 2. Changing a threshold R allows us to control the retrieval flexibility. As the flexibility threshold R gets bigger, more candidates will pass the threshold and will match the input sketch, giving higher retrieval time. Setting R, therefore, presents a tradeoff between false alarms and misses. Algorithms 1 and 2 serve as pre-selections in the retrieval process. If the containment trees of the sketch and the image match according to the appropriate algorithm, the final verification is performed using multivalued signatures.

Figure 9 presents an example of sketch-based query. The lower row presents the distance values for the database items. One can see that the values confirm the intuitive measure of the similarity between the sketch and the images.

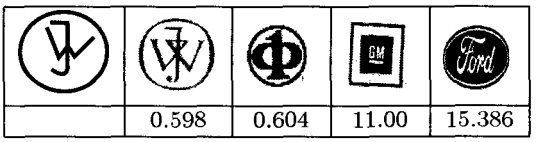


Fig. 9. Sketch query example. The input sketch is on the left.

Algorithm 2
Flexible sketch matching.
Given: An input sketch S, and a database image I,

1. Build T_1 and T_2 as in Alg. 1.
2. Check if T_1 flexibly matches T_2.

We define a flexible match in the following manner:
The containment tree T_1 for the sketch S flexibly matches the containment tree T_2 for the image I if:

1. T_1 is a leaf **or**
2. The number of subtrees of T_1 equals that of T_2 and the corresponding sons of T_1 and T_2 flexibly match **or**
3. The number of subtrees of T_1, n, is less than that of T_2, but the $(n+1)^{th}$ son of T_2 corresponds to a contour which is much smaller than the contour corresponding to the n^{th} son of T_2 (that is, the ratio of their areas is less than some predefined threshold R), and the first n corresponding sons of T_1 and T_2 flexibly match.

3 Shape retrieval

Our shape retrieval scheme consists of two main phases: filtering the database in order to drop the irrelevant images, and ranking the *candidates subset* according to the distance from the input image. For indexing we use geometric entities, such as circles, ellipses, etc. The number of these entities can serve for efficient filtering of the image collection. Final ranking within the set is based mainly on the semi-local multi-valued invariant signatures.

The general scheme of query processing is presented on Fig. 10. After edge detection and curves extraction from the input image, geometric entities are detected. This global features extraction is used as a basis for the filtering process. Relational database is used in this stage in order to retrieve the candidates subset. Features used for indexing to the relational database and eventually for pruning the candidates set vary from the number of the curves and the geometric entities, their relative dimensions, etc. The relational database includes alpha-numerical data as well. The query may contain this kind of information, like the organization name (for the trademarks database) etc., and this part of the query will be processed as a regular query (further pruning the set). The candidates obtained at this stage go to a matching with the input image. The final set, ordered according to the similarity measure is given as an answer.

4 Experiments

To show the applicability of our approach we tested it first on a database of 500 constrained 3D objects, namely surfaces of revolution. Using the KBS bottles collection ([15]) we present queries showing good retrieving ability for a specific,

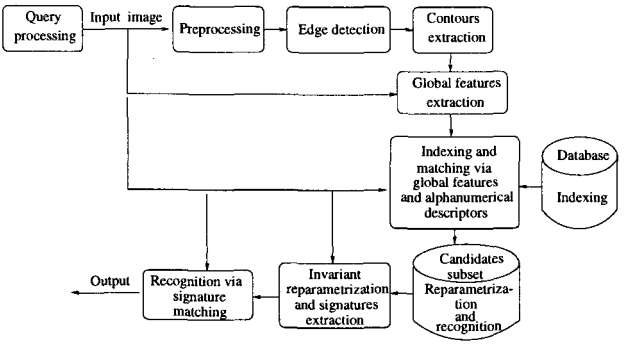


Fig. 10. Query processing.

and similar, objects. The effectiveness of the filtering process and the ability to handle different transformations are demonstrated in the second part in which a database of 100 trademarks is tested. We give results for sketch-based queries on a database of more than 300 "road signs". For all the four databases we used the same system. The system is implemented using ORACLE/SQL environment, and has a WWW interface. The user is allowed to query images by example, logical description or the combination of both. The example image can be the database image or the image provided by the user.

4.1 Invariant signatures for surfaces of revolution

An interesting practical application field for multivalued signatures is recognition in a database of 3-D objects which are surfaces of revolution (see also Mundy et al. [16]). We can treat such objects as planar under a controlled change of the viewpoint (which is common in industrial settings, e.g. in assembly lines). In this case, we can approximately describe the viewpoint change by affine transformation and apply our algorithms. In our experiment, the database contains more than 500 items some of which are photographed objects that are surfaces of revolution and the rest are bottles from the KBS bottles collection ([15]). Given an input image (one of the database objects under rotation and zoom), we compared it with the database by using multivalued signatures. Figure 11(a-c) presents an input image, its external contour and its multi-valued invariant signature. This input image was checked against all the images in the database. A good match was achieved retrieving the right database image in the first place. Figure 11(d) presents the grey-level maps of the input (above) and the database (below) contours at the shift position giving the best match. Retrieval results are presented next (first four of the set). Figure 11 (right) presents a number of queries performed using the surfaces of revolution database.

Figure 12(a) presents our HTML interface. The input scene and the processing results are presented on Fig. 12(b).

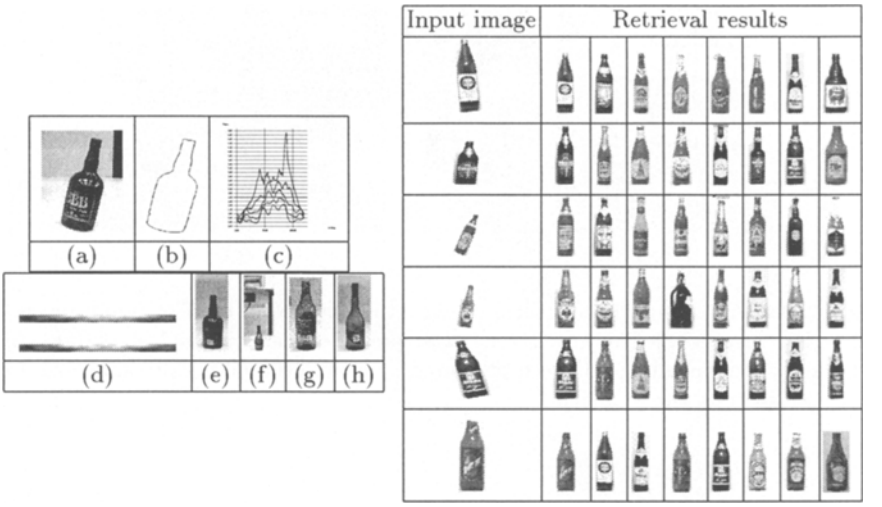


Fig. 11. Left: retrieval process. (a) The input image. (b) The external contour of the input image. (c) The invariant signature for the input image. (d) The best match for the input image. (e) – (h) Retrieval results. Right: results of queries on the bottles database.

4.2 Filtering the candidates set

Next we illustrate the effective filtering obtained by using invariant features. We run our queries on a database which contains more than 100 trademarks. In the first example (Figure 13(1)), the input image is a rotated and scaled version of one of the database trademarks. The user operated with the query
select title ordered by dist(logo30-trans.tiff) where (context = text).
The query looks for images having textual strings in them. Both alpha-numerical information and geometric features are used for filtering the database. Exploiting geometric features is especially effective for images with a large number of curves, because of their complicated structure. One can see that the number of candidates is rather small. In the following example (Figure 13(2)), the user operated with
select title where (N_circles = 1 and N_curv $\geq$ N_curv(logo55.tiff)),
directly specifying the number of geometric entities (circles). The input image is one of the database trademarks. One can see that the candidates comply with the given condition. The user has the freedom to limit the transformation which the images are allowed to undergo by stating the transformation explicitly in the query. In the next example (Figure 13(3)), the user limited the transformations to the affine case, operating with the query

select title ordered by dist(logo50-transf.tiff) under affine.

Both the filtering and the ranking are based on features that are invariant under the affine transformation. The input image is an occluded version of one of the database trademarks.

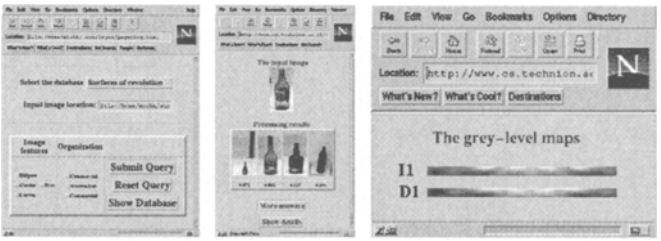


Fig. 12. (a) Querying the database using html interface (b) The input image and processing results (c) The grey-level maps, presenting multivalued signatures: above - the map corresponding to the input image; below - the map, corresponding to the image which took the first place.

The next example presents sketch-based retrieval (Figure 13(4)). The sketch was drawn using curves-drawing software (Xfig). Here we allow to omit some small objects within the image, while preserving its general structure. In this case we used algorithm 2 for filtering the database.

	Input image	Retrieval results			
1		 Place 1 0.128	 Place 2 9.57	... ...	 Place 5 1492.57
2		 Place 1	 Place 2	... ...	 Place 10
3		 Place 1 0.233	 Place 2 0.804	... ...	 Place 23 1347.148
4		 Place 1 0.526	 Place 2 0.632	 Place 3 0.961	 Place 4 0.961

Fig. 13. Queries results

Note that in all the examples the number of the images in the candidates set is sufficiently smaller than the number of the items in the database. The filtering is still effective even where occlusion is present (see, for example Fig. 13(3)). When the filtering is based on the containment tree we expect it to be more effective (see, for example, Fig. 13(4), where only 4 candidates left).

4.3 Sketch-based queries

Figure 14 presents a number of sketch-based queries and retrieval results. In addition, we present the queries in SQL-like notation (See Appendix A). The queries are performed on the database of "road signs", containing about 300 images. After filtering based on Alg. 1 or 2, the candidates passed matching with the input sketch, based on invariant signatures. For each query, the resulting candidates set contains no more than 15 images. One can see that the distance measure reflects the similarity between the sketch and the candidate image.

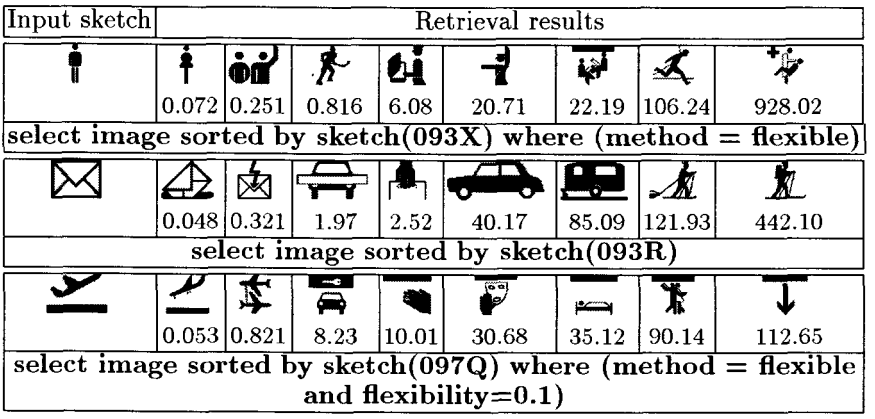


Fig. 14. Results of queries on the "road signs" database.

4.4 Efficiency analysis

In this section we used the combined database which included both the database of the trademarks and the "road signs". Figure 15(a) presents the average processing time (averaging on 100 queries) for a query as a function of the number of the curves in the input image. The upper curve corresponds to a sequential matching, i.e. a full signatures matching against the entries in the database. The middle presents average processing time for a query using the number of geometric entities for pruning the database. The lower curve corresponds to retrieval using the containment tree. The middle and lower curves are non-monotonous, a direct consequence of a tradeoff between a shorter query execution time as a result of the pruning and the extra time needed for matching more signatures. The graphs demonstrate that geometric entities and especially the containment tree are effective means for database pruning and their efficiency grows with the number of contours.

Figure 15(b) demonstrates the scalability of our approach. Queries have been run to retrieve an image from a subset of the database. The queries used the

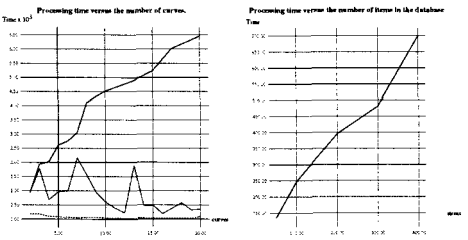


Fig. 15. Left: average evaluation time versus a number of curves. Right: scalability - processing time versus the number of database items.

number of geometric entities for pruning the subset. When the subset grows, processing time increases approximately as a linear function.

Table 1 illustrates the influence of the flexibility threshold R (See Alg. 2) on the number of retrieved images for sketch-based retrieval. The number of the answers depends on the complexity of the input sketch.

Table 1. The number of retrieved images as a function of the flexibility threshold R. For the presented input sketch, the number of retrieved images is given for different values of R.

R			
0.01	2	4	10
0.02	3	6	12
0.04	6	15	18
0.08	10	16	18
0.16	14	25	20

5 Conclusions

In this paper we have addressed the problem of shape-based retrieval from image databases. Our approach emphasizes the use of invariants as shape descriptors. Specifically, we have used geometric invariant features for efficient indexing, while local multi-valued invariant signatures have been used for ranking the answers. The substantial reduction of the candidates set due to the filtering stage guarantees the efficient retrieval. The approach supports image retrieval while part of the shape is missing, can handle images distorted by different viewpoint transformations, and can flexibly answers queries based on logical descriptions, shape (query by example), or combination of both.

We have implemented our approach in a heterogeneous database system having a SQL-like user interface augmented with sketch-based queries. The system is built on top of a commercial database system (Oracle), and can be activated from the Web. We have presented experimental results demonstrating the effectiveness of the proposed approach under various conditions using three different databases.

References

1. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovich, and W. Equitz, Efficient and effective querying by image content, *Journal of Intelligent Information Systems* **3**, 1994, 231–262.
2. A. Pentland, R. Picard, and S. Sclaroff, Photobook: Tools for content-base manipulation of image databases, in *Proceedings, SPIE Conf. on Storage and Retrieval of Image Databases II*, 1994, pp. 37–50.
3. A. Jain and A. Vailaya, Image retrieval using color and shape, *Pattern Recognition* **29**, 1996, 1233–1244.
4. A. D. Bimbo and P. Pala, Visual image retrieval by elastic matching of user sketches, *IEEE Trans. Patt. Anal. Mach. Intell.* **19**, 1997, 121–132.
5. S.-F. Chang, J. Smith, and H. Wang, Automatic feature extraction and indexing for content-based visual query, Tech. Rep. 408-95-14, Columbia University Center for Telecommunications Research, New York, NY, 1995.
6. S.-F. Chang and J. Smith, Extracting multi-dimensional signal features for content-based visual query, in *SPIE Int. Symp. on Visual Communications and Image Processing*, 1995.
7. R. Mehrotra and J. Gary, Similar-shape retrieval in shape data management, *IEEE Computer* **28**, 1995, 57–62.
8. K. Shields, J. Eakins, and J. Boardman, Automatic image retrieval using shape features, *New Review of Document and Text Management* **1**, 1995, 183–198.
9. S. Lee and F.J.Hsu, Spatial reasoning and similarity retrieval of images using 2D–C string knowledge representation, *Pattern Recognition* **25**, 1992, 305–318.
10. T. Moons, E. Pauwels, L. V. Gool, and A. Oosterlinck, Foundations of semi-differential invariants, *Int. Journal Of Comput. Vision* **14**, 1995, 25–47.
11. A. Bruckstein, J. Holt, A. Netravali, and T. Richardson, Invariant signatures for planar shape recognition under partial occlusion, *CVGIP:Image Understanding* **58**, 1993, 49–65.
12. A. Bruckstein, E. Rivlin, and I. Weiss, Recognizing objects using scale space local invariants, in *Int. Conf. on Pattern Recognition*, Vienna, 1996, p. A9M.6.
13. T. Moons, E. Pauwels, L. V. Gool, and A. Oosterlinck, Recognition of planar shapes under affine distortion, *Int. Journal Of Comput. Vision* **14**, 1995, 49–65.
14. J. Köbler, U. Schöning, and J. Torán, *The graph isomorphism problem.* Birkhäuser, Boston, 1993.
15. Kbs beer bottle collection, *URL: http://tuoppi.oulu.fi/kbs/beer/kbsbeer.htm*
16. A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow, 3d object recognition using invariance, *Artificial Intelligence* **78**, 1995, 239–288.