On computing local and global similarity in images

R. Manmatha, S. Ravela and Y. Chitti

Multimedia Indexing and Retrieval Group
Center for Intelligent Information Retrieval
University of Massachusetts, Amherst, MA 01003

ABSTRACT

The retrieval of images based on their visual similarity to an example image is an important and fascinating area of research. Here, we discuss various ways in which visual appearance may be characterized for determining both global and local similarity in images.

One popular method involves the computation of global measures like moment invariants to characterize global similarity. Although this means that the image may be characterized using a few numbers, the performance is often poor. Techniques based on moment invariants often perform poorly. They require that the object be a binary shape without holes which is often not practical. In addition, moment invariants are sensitive to noise.

Visual appearance is better represented using local features computed at multiple scales. Such local features may include the outputs of images filtered with Gaussian derivatives, differential invariants or geometric quantities like curvature and phase. Two images may be said to be similar if they have similar distributions of such features. Global similarity may, therefore, be deduced by comparing histograms of such features. This can be done rapidly. Histograms cannot be used to compute local similarity. Instead, the constraint that the spatial relationship between the features in the query be similar to the spatial relationship between the features of its matching counterparts in the database provides a means for computing local similarity. The methods presented here do not require prior segmentation of the database. In the case of local representation objects can be embedded in arbitrary backgrounds and both methods handle a range of size variations and viewpoint variations up to 20 or 25 degrees.

Keywords: filter based representations, appearance based representations, scale space matching, vector correlation, image retrieval, image indexing.

1. INTRODUCTION

The advent of large multi-media collections and digital libraries has led to a need for good search tools to index and retrieve information from them. For text available in machine readable form (ASCII) a number of good search engines are available. However, there are as yet no good tools to retrieve images. The traditional approach to searching and indexing images using manual annotations is slow, labor intensive and expensive. In addition, textual annotations cannot encode all the information available in an image. There is thus a need for retrieving images using their content. The indexing and retrieval of images using their content is a poorly understood and difficult problem. A person using an image retrieval system usually seeks to find semantic information. For example, a person may be looking for a picture of a leopard from a certain viewpoint. Or alternatively, the user may require a picture of Abraham Lincoln from a particular viewpoint. Retrieving semantic information using image content is difficult to do. The automatic segmentation of an image into objects is a difficult and unsolved problem in computer vision. However, many image attributes like color, texture, shape and "appearance" are often directly correlated with the semantics of the problem. For example, logos or product packages (e.g., a box of Tide) have the same color wherever they are found. The coat of a leopard has a unique texture while Abraham Lincoln's appearance is uniquely defined. These image attributes can often be used to index and retrieve images.

A common model for retrieval, and one that is adopted here, is that images in the database are processed and described by a set of feature vectors. A priori these vectors are indexed. During run-time, a query is provided in the form of an example image and its features are compared with those stored. Images are then retrieved in the order indicated by the comparison

Email: {manmatha,ravela}@cs.umass.edu This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademarks Office and the Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation under grant number IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

operator. In this paper, images “similar” to a given query image are retrieved by comparing the query with the database using a characterization of the visual appearance of objects. Intuitively an object’s visual appearance in an image is closely related to a description of the shape of its intensity surface. An object’s appearance depends not only on its three dimensional shape, but also on the object’s albedo, its surface texture, the viewpoint from which it is imaged and a number of other factors. It is non-trivial to separate the different factors constituting an object’s appearance and it is usually not possible to separate an object’s three dimensional shape from the other factors. For example, the face of a person has a unique appearance that cannot just be characterized by the geometric shape of the ’component parts’. Similarly the shape of a car such as the one shown in Figure 1(a) is not just a matter of a geometric-shape outline of its ’individual parts’. In this paper we characterize the shape of the intensity surface of imaged objects and the term *appearance* will be associated with the ’shape of the intensity surface’. The experiments conducted in this paper verify this association. That is, objects that appear to be visually similar can be retrieved by a characterization of the shape of the intensity surface.

Specifically, a claim is made that, visual appearance can be captured using a set of multi-scale Gaussian derivative filters. Koenderink[12] and others[5] have argued that the local structure of an image can be represented by the outputs of a set of Gaussian derivative filters applied to an image. That is, images are filtered with Gaussian derivatives at several scales and the resulting response vector locally describes the structure or shape of the intensity surface. Note that the response vector or *multi-scale derivative feature vector* is a local descriptor of the intensity structure since responses are defined only in a finite neighborhood around the point where they are computed. Using this local characterization as a basis, we argue that representations appropriate to querying parts of images (local similarity) or images as a whole (global similarity) can be generated.

Using the multi-scale derivative features local similarity retrieval is carried out as follows. Database images are uniformly sampled. At each sampled location, a multi-scale derivative feature vector is computed and transformed so that it is invariant to 2D rotations of the underlying surface. Then, multi-scale invariant vectors computed for all the images in the database are indexed using a binary tree structure. During run-time, the user picks an example image and “designs” a query. Since an image is described spatially (uniform sampling) parts of images or (imaged objects) can be selected. For example, consider Figure 1(a). Here the user wants to retrieve white wheeled cars and therefore, appropriately selects the white wheel. The feature vectors that lie within this region are submitted to the system. Database images with feature vectors that match the set of query vectors both in feature space (L2 norm of the vector) and are consistent with the spatial distribution of query vectors are then returned as the retrievals.

Sampled multi-scale invariant vectors form a local representation that naturally lend to local similarity retrieval by appearance. Local similarity allows the use of only those parts of the image which a person considers to be consistent over a number of images, and hence allows for querying parts of images. The approach presented here avoids segmentation, and furthermore, avoids object recognition, or detection of “salient features”. In fact, all the system does is compare signals. The context within which comparison is performed is presented in the form of a query. Queries are designed by users to express their “semantic intent”. Sometimes such queries will be appropriate and at others they may not. With a sufficiently fast system, the user can quickly evaluate the results and reformulate queries when retrievals are unsatisfactory.

The main disadvantage of using local similarity is the computational complexity inherent the process. Determining local similarity requires finding a part of an image which is similar to the query. This implies that the space of translations must be explicitly searched or represented (as is done here with uniform sampling). In addition, isolated invariant vectors can be similar in feature space even if the object on the whole is dissimilar. Adding the spatial constraint (coordinate space) eliminates many of these “false” matches, but at a computational cost. Finally, the local similarity retrieval method as presented here cannot be efficiently used to query images as a whole.

However, the same Gaussian derivative model can be used to efficiently retrieve by global similarity of appearance. Using the multi-scale derivative features global similarity retrieval is carried out by representing the distribution of feature vectors over the entire image. Database images are filtered with Gaussian derivative filters as in the local similarity case. Then two geometric entities namely curvature and phase are computed and their histograms over the entire image are generated. These curvature and phase histograms are generated at multiple scales. Thus, the image is now described represented by a single vector (multi-scale histograms) as opposed to a set of spatially distributed responses. During run-time the user presents an example image as a query and the query histograms are compared with ones stored, ranked and displayed in order to the user.

Histograms of features derived from the multi-scale Gaussian derivatives form a global representation because they capture the distribution of local features. This global representation can be efficiently used for global similarity retrieval by appearance and retrieval is very fast. Thus images as a whole can be queried efficiently. The utility of global similarity retrieval is evident for example in finding similar scenes or similar faces in a face database. In addition, practical applications such as finding similar trademarks in a trademark database significantly benefit from global similarity retrieval.

In the context of global similarity retrieval it should be noted that representations using moment invariants have been well studied. In these methods global representation of appearance may involve computing a few numbers over the entire image. Two images are then considered similar if these numbers are close to each other (say using an L2 norm). We will argue that such representations are not able to really capture the "appearance" of an image, particularly in the context of trademark retrieval where moment invariants are widely used.

The rest of the paper is organized as follows. Section 2 surveys related work in the literature. In section 3, the notion of appearance is developed further and characterized using Gaussian derivative filters. Section 4 discusses an invariant form and local characterization of appearance and shows how it may be used for local similarity retrieval. Then in Section 5 a global characterization of images using local derivative features is discussed. Comparisons are made in the context of trademark retrieval with the traditional moment invariants. A discussion and conclusion follows in Section 6.

2. RELATED WORK

Several authors have tried to characterize the appearance of an object via a description of the intensity surface. In the context of object recognition[17] represent the appearance of an object using a parametric eigen space description. This space is constructed by treating the image as a fixed length vector, and then computing the principal components across the entire database. The images therefore have to be size and intensity normalized, segmented and trained. Similarly, using principal component representations described in[9] face recognition is performed in.[25] In[23] the traditional eigen representation is augmented by using most discriminant features and is applied to image retrieval. The authors apply eigen representation to retrieval of several classes of objects. The issue however is that these classes are manually determined and training must be performed on each. The approach presented in this paper is different from all the above because eigen decompositions are not used at all to characterize appearance. Further the method presented uses no learning, does not depend on constant sized images and deals with embedded backgrounds and heterogeneous collections of images using local representations of appearance.

The use of Gaussian derivative filters to represent appearance is motivated by their use in describing the spatial structure[12] and its uniqueness in representing the scale space of a function[13,10,27,24] and the fact that the principal component of images are best described as Gaussians and their derivatives.[7] That is there is a natural decomposition of images into Gaussians and their derivatives. The use of invariant transformations of Gaussians is borrowed from descriptions provided by.[5] In[19] recognition is done by using a vector of Gaussian derivatives which are indexed. Schmid and Mohr[22] use indexed differential invariants for object recognition. We also index on differential invariants but there are several differences between the approach presented here and theirs. First, in this work only the low two orders are used, which is more relevant to retrieving similar images (see section 3) while they use nine-invariants. Second, their indexing algorithm depends on interest point detection and is, therefore, limited by the stability of the interest operator. We on the other hand sample the image. Third, the authors do not incorporate multiple scales into a single vector whereas here three different scales are chosen. In addition the index structure and spatial checking algorithms differ. Schmid and Mohr apply their algorithm primarily to the problem of object recognition, do not allow for the user to determine saliency and therefore have not applied their algorithm to retrieving similar images.

The earliest general image retrieval systems were designed by.[4,18] In[4] the shape queries require prior manual segmentation of the database which is undesirable and not practical for most applications.

Texture based image retrieval is also related to the appearance based work presented in this paper. Using Wold modeling, in[14] the authors try to classify the entire Brodatz texture and in[6] attempt to classify scenes, such as city and country. Of particular interest is work by[15] who use Gabor filters to retrieve texture similar images, without user interaction to determine region saliency.

3. SYNTACTIC REPRESENTATION OF APPEARANCE

It has been argued by Koenderink and van Doorn[12] and others[5] that the local structure of an image I at a given scale can be represented using Gaussian derivative filters, and they term it the N-jet. The N-jet of an image can be derived using a Taylor series expansion. However, in order to guarantee that the Taylor series expansion is stable, the derivatives must themselves be stable. That is, the Image must be regularized, or smoothed or band-limited. A Gaussian filtered image $I_\sigma = I * G$ obtained by convolving the image I with a normalized Gaussian $G(\mathbf{r}, \sigma)$ is a band-limited function. Its high frequency components are eliminated and derivatives will be stable. Now given the intensity $I_g(\mathbf{p})$ at some point $\mathbf{p}$ in the smoothed image, the intensity $I_g(\mathbf{p} + \mathbf{r})$ at a point $\mathbf{p}$ + $\mathbf{r}$ in the neighborhood of $\mathbf{p}$ can be obtained by using a Taylor series expansion.

If two filtered images are locally identical at some scale σ, then their Taylor series expansions over some neighborhood must be the same. The terms in the Taylor series, therefore, capture the local structure or *appearance*of the image. Formally, the N-jet at a point $\mathbf{p}$ in an image at a scale σ is given by a vector

$$J^N[I](\mathbf{p},\sigma) = (I(\mathbf{p}+\mathbf{r}) * G(\mathbf{r},\sigma), I(\mathbf{p}+\mathbf{r}) * G_x(\mathbf{r},\sigma), I(\mathbf{p}+\mathbf{r}) * G_y(\mathbf{r},\sigma), ..., I(\mathbf{p}+\mathbf{r}) * G_{x^i y^j}(\mathbf{r},\sigma), ...) \tag{1}$$

obtained by expanding the Taylor series in terms of Gaussian derivatives up to order N.

A key choice in constructing the N-jet is the order N. We are interested in retrieving not just the same object but also images of similar objects. For example, given a query which consists of Abraham Lincoln's face, it is desirable that other examples of Lincoln's face be retrieved first, followed by faces of other people. Images of similar objects will be much less correlated than images of the same object. Empirically, it is determined that for retrieving similar objects, m = 2, that is, only the 2-jet's of two similar images can be expected to be the same.

The local 2-jet of an image I at a point $\mathbf{p}$ and a scale σ is given by:

$$J^2\left[I\right]\left(\mathbf{p},\sigma\right) = \left\{I, I_x, I_y, I_{xx}, I_{xy}, I_{yy}\right\}\left(\mathbf{p},\sigma\right)^*$$

where $I_{x^i y^j,\sigma} = I * G_{x^i y^j,\sigma}$ That is image I filtered with the first two Gaussian derivatives (and the Gaussian itself) in both x and y directions. Point p is, therefore, associated with a *feature vector* of responses at scale σ.

The shape of the local intensity surface depends on the scale at which it is observed. An image will appear different depending on the scale at which it is observed. For example, at a small scale the texture of an ape's coat will be visible. At a large enough scale, the ape's coat will appear homogeneous. A description at just one scale is likely to give rise to many accidental mis-matches. Thus it is desirable to provide a description of the image over a number of scales, that is, a scale space description of the image. The local N-jet captures the local structure of the image only at a particular scale. However, it can be extended to provide a multi-scale description. From an implementation stand point a *multi-scale feature vector* at a point p in an image I is simply the vector:

$$\left\{J^2\left[I\right]\left(\mathbf{p},\sigma_1\right), J^2\left[I\right]\left(\mathbf{p},\sigma_2\right) \ldots J^2\left[I\right]\left(\mathbf{p},\sigma_k\right)\right\} \tag{2}$$

for a set of scales $\sigma_1 \ldots \sigma_k$. In practice the zeroth order terms are dropped to achieve invariance to constant intensity changes.

Not all scales are required. Since adjacent scales are strongly correlated only a finite number of scales are used. In this paper, adjacent scales are half an octave apart. In addition, two objects are similar only over a range of scales. For similarity matching, the fine details which distinguish say one car model from another are a hindrance. These fine details correspond to small scales. That is, for similar images their low frequency (large scales) descriptions are likely to correspond. Therefore, we use a small set of scales to characterize appearance. Thus the multi-scale derivative feature vector forms the substrate upon which representations suitable for local and global similarity retrieval are built. In the next section we discuss local similarity that allows us to retrieve objects embedded within images.

4. LOCAL SIMILARITY RETRIEVAL

Local similarity retrieval implies that databases can be queried for parts of images. In this section we develop a part-image retrieval system that uses local derivative feature vectors to retrieve images by local similarity of appearance.

One of the simplest mechanisms to verify that the derivative feature vectors are suitable for retrieval by appearance is correlation. That is, the feature vectors corresponding to the query are correlated with those in the database and images ranked by the correlation score. In earlier work[20] such a verification was conducted and it was found that visually similar images can be retrieved within small view variations (20^o) and finite size variations by explicitly searching across scales.

However, correlation is not indexable and is expensive. Secondly, the use of derivative feature vectors directly makes the approach sensitive to rotations. Here a local representation is derived by transforming the multi-scale feature vector (that is, the multi-scale N-jet) in equation 2 so that it is invariant to 2D rigid transformations. Further these vectors are indexed for fast retrieval.

The approach for retrieval can be divided into two parts. During the off-line phase, images are sampled and multi-scale derivatives are computed. These are transformed to rotational invariants and indexed. During the On-line phase the user designs a query by selecting regions within an image. Feature vectors within the query are matched with those in the database both in feature space and coordinate space. The off-line and on-line phases are discussed next.

[*] $I_{yx} = I_{xy}$ and is therefore dropped

4.1. Off-line Operations: Invariant Vectors and Indexing

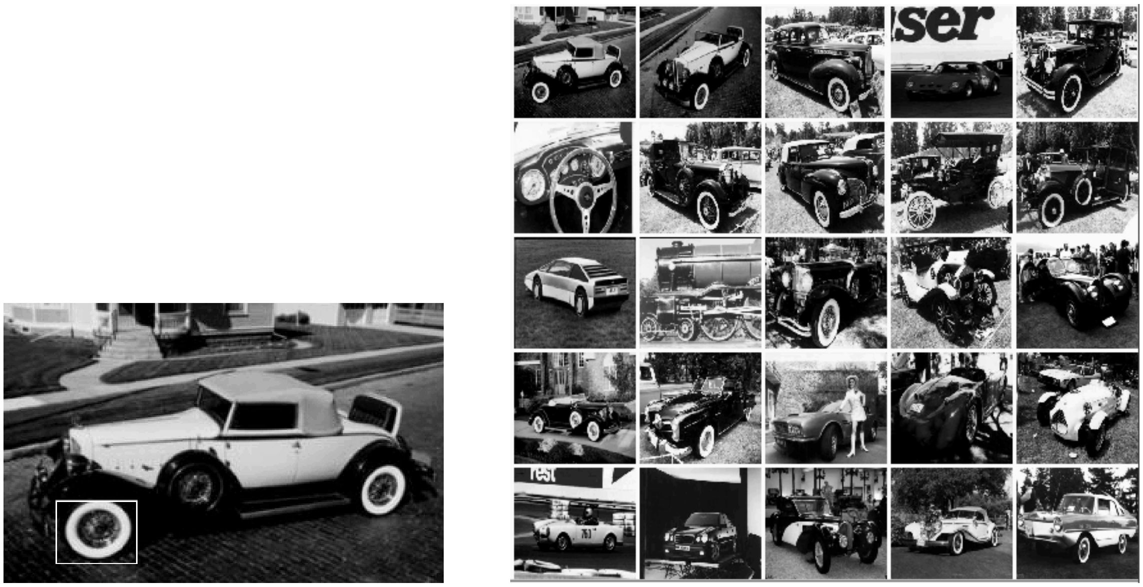
Given the derivatives of an image I, *irreducible differential invariants* (invariant under the group of displacements) can be computed in a systematic manner.[5] The value of these entities is independent of the choice of coordinate frame (up to rotations). The irreducible set of invariants up to order two of an image I are:

$$\begin{array}{lll} d_0 & = I & \text{Intensity} \\ d_1 & = I_x^2 + I_y^2 & \text{Magnitude} \\ d_2 & = I_{xx} + I_{yy} & \text{Laplacian} \\ d_3 & = I_{xx} I_x I_x + 2 I_{xy} I_x I_y + I_{yy} I_y I_y & \\ d_4 & = I_{xx}^2 + 2 I_{xy}^2 + I_{yy}^2 & \end{array}$$

Here, the vector, $\Delta_\sigma = \langle d_1, \ldots d_4 \rangle_\sigma$ is computed at three different scales. The element d_0 is not used since it is sensitive to gray-level shifts. The resulting multi-scale invariant vector has at most twelve elements. Computationally, each image in the database is filtered with the first five partial derivatives of the Gaussian (i.e. to order 2) at three different scales at uniformly sampled locations. Then the multi-scale invariant vector $D = \langle \Delta_{\sigma_1}, \Delta_{\sigma_2}, \Delta_{\sigma_3} \rangle$ is computed at those locations.

A location across the entire database can be identified by the *generalized coordinates*, defined as, $c = (i, x, y)$ where i is the image number and (x, y) a coordinate within this image. The computation described above generates an association between generalized coordinates and invariant vectors. This association can be viewed as a table $M : (i, x, y, D)$ with $3 + k$ columns(k is the number of fields in an invariant vector) and number of rows, R, equal to the total number of locations (across all images) where invariant vectors are computed. To retrieve images. a 'find by value' functionality is needed, with which, a query invariant vector is found within M and the corresponding generalized coordinate is returned. Inverted files (or tables) are based on each field of the invariant vector are first generated for M. To index the database by fields of the invariant vector, the table M is split into k smaller tables $M_1^{'} \ldots M_k^{'}$, one for each of the k fields of the invariant vector. Each of the smaller tables $M_p^{'}, p = 1 \cdots k$ contains the four columns $(D(p), i, x, y)$. At this stage any given row across all the smaller tables contains the same generalized coordinate entries as in M. Then, each M_p' is sorted and a binary tree is used to represent the sorted keys. As a result, the entire database is indexed. A given invariant value can, therefore, be located in log(R) time (R = number of rows).

4.2. On-line Operation



(a) Car Query

(b) Ranked Retrieval

Figure 1. A Query and its retrieval

Run-time computation begins with the user marking selecting regions in an example image. At sampled locations within these regions, invariant vectors are computed and submitted as a query. The success of a retrieval in part depends on well designed queries. More importantly, letting the user design queries eliminates the need for automatically detecting the salient portions of an object, and the retrieval can be customized so as to remove unwanted portions of the image. Based on the feedback provided by the results of a query, the user can quickly adapt and modify the query to improve performance.

The search for matching images is performed in two stages. In the first stage each query invariant is supplied to the 'find-by-value' algorithm and a list of matching generalized coordinates is obtained. In the second stage a spatial check is performed on a per image basis, so as to verify that the matched locations in an image are in spatial coherence with the corresponding query points. In this section the 'find-by-value' and spatial checking components are discussed.

4.2.1. Finding by invariant value

The multi-scale invariant vectors at sampled locations within regions of a query image can be treated as a list. The n^{th} element in this list contains the information $Q_n = (D_n, x_n, y_n)$, that is, the invariant vector and the corresponding coordinates. In order to find by invariant value, for any query entry Q_n, the database must contain vectors that are within a threshold $t = (t_1 \dots t_k) > 0$. The coordinates of these matching vectors are then returned. This can be represented as follows. Let p be any invariant vector stored in the database. Then p matches the query invariant entry D_n only if $D_n - t < p < D_n + t$. To implement the comparison operation two searches can be performed on each field. The first is a search for the lower bound, that is the smallest entry larger than $D_n(j) - t(j)$ and then a search for the upper-bound i.e. the largest entry smaller than $D_n(j) + t(j)$. The block of entries between these two bounds are those that match the field j. In the inverted file the generalized coordinates are stored along with the individual field values and the block of matching generalized coordinates are copied from disk. Then an intersection of all the returned block of generalized coordinates is performed. The generalized coordinates common to all the k fields are the ones that match query entry Q_n. The find by value routine is executed for each Q_n and as a result each query entry is associated with a list of generalized coordinates that it matches to.

4.2.2. Spatial-fitting

The association between a Query entry Q_n and the list of f generalized coordinates that match it by value can be written as

$$A_n = \left\langle x_n, y_n, c_{n_1}, c_{n_2} \dots c_{n_f} \right\rangle = \left\langle x_n, y_n, (i_{n_1}, x_{n_1}, y_{n_2}) \dots (i_{n_f}, x_{n_f}, y_{n_f}) \right\rangle$$

. Here x_n, y_n are the coordinates of the query entry Q_n and $c_{n_1} \dots c_{n_f}$ are the f matching generalized coordinates. The notation c_{n_f} implies that the generalized coordinate c matches n and is the f^{th} entry in the list. Once these associations are available, a spatial fit on a per image basis can be performed. Any image u that contains two points (locations) which match some query entry m and n respectively are coherent with the query entries m and n only if the distance between these two points is the same as the distance between the query entries that they match. Using this as a basis, a binary fitness measure can be defined as

$$\mathcal{F}_{m,n}(u) = \begin{cases} 1 & \text{if } \exists j \exists k \mid \left| \delta_{m,n} - \delta_{c_{m_j}, c_{n_k}} \right| \leq T, i_{m_j} = i_{n_k} = u, m \neq n \\ 0 & \text{otherwise} \end{cases}$$

where $\delta_{m,n}$ is the Euclidean distance between the query points m and n, and $\delta_{c_{m_j}, c_{n_k}}$ is the Euclidean distance between the generalized coordinates c_{m_j} and c_{n_k} That is, if the distance between two matched points in an image is close to the distance between the query points that they are associated with, then these points are spatially coherent (with the query). Using this fitness measure a match score for each image can be determined. This match score is simply the maximum number of points that together are spatially coherent (with the query). Define the match score by: $score(u) \equiv \overset{max}{m} \sum_{n=1}^{f} \mathcal{F}(u)_{m,n}$. The computation of $score(u)$ is at worst quadratic in the total number of query points. The array of scores for all images is sorted and the images are displayed in the order of their score. T used in $\mathcal{F}$ is a threshold and is typically 25% of $\delta_{m,n}$. Note that this measure not only will admit points that are rotated but will also tolerate other deformations as permitted by the threshold. It is placed to reflect the rationale that similar images will have similar responses but not necessarily under a rigid deformation of the query points.

4.3. EXPERIMENTS

The database used for the local similarity retrieval has digitized images of cars, steam locomotives, diesel locomotives, apes, faces, people embedded in different background(s) and a small number of other miscellaneous objects such as houses. 1561 images were obtained from the Internet and the Corel photo-cd collection to construct this database. These photographs were taken with several different cameras of unknown parameters, and under varying uncontrolled lighting and viewing geometry.

Prior to describing the experiments, it is important to clarify what a correct retrieval means. A retrieval system is expected to answer questions such as ’find all cars similar in view and shape to this car’ or ’find all faces similar in appearance to this one’. In the examples presented here the following method of evaluation is applied. First, the objective of the query is stated and then retrieval instances are gauged against the stated objective. In general, objectives of the form ’extract images similar in appearance to the query’ will be posed to the retrieval algorithm.

A measure of the performance of the retrieval engine can be obtained by examining the recall/precision table for several queries. Briefly, recall is the proportion of the relevant material actually retrieved and precision is the proportion of retrieved material that is relevant.[26] Consider as an example the query described in Figure 1(a). Here the user wishes to retrieve ’white wheel cars’ similar to the one outlined and submits the query. The top 25 results ranked in text book fashion are shown in Figure 1(b). Note that although there are several valid matches as far as the algorithm is concerned (for example image 12 a train), they are not considered valid retrievals as stated by the user and are not used in measuring the recall/precision. This is inherently a conservative estimate of the performance of the system. The average precision (over recall intervals of 10[†]) is 48.6%.

One of the important parameters in constructing indices is the sample rate. Recall that indices are generated by computing multi-scale invariant feature vectors at uniformly sampled locations within the image. The performance of the system is evaluated under sample rates of three pixels and five pixels respectively. The case where every pixel is used could not be implemented due to prohibitive disk requirements and lack of resources to do so. Six other queries that are also submitted are depicted in table 1. The recall/precision table over all seven queries is in Table 4.3. The third column of table shows the average precision for each query with a database sampling of 5 pixels and the fourth column shows with 3 pixels. The average precision and precision at recall intervals of 10, over all the queries for both samplings is shown in Table 4.3. This compares well with text retrieval where some of the best systems have an average precision of 50%[‡]. The average precision over the same seven queries with both three and five pixel sampling is 56.2% for the five pixel case and 61.7% in the three pixel case. However, while the increase in sampling improves the precision it results in an increased storage requirement.

Given(User Input)	Find	Precision (5)	Precision (3)
Face	All Faces	74.7%	61.5%
Face	Same Person’s Face,	61.7%	75.5%
Monkey’s coat	Dark Textured Apes	57.5%	57%
Both wheels	White Wheeled Cars	57.0%	63.7%
Coca Logo	All Coca Cola Logos	49.3%	74.9%
wheel see Figure 1(a)	White Wheeled Cars, see Figure 1(b)	48.6%(see text)	54.4%
Patas Monkey Face	All Visible Patas Monkey Faces	44.5%	47.1%

Table 1. Queries submitted to the system and expected retrieval

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision(5) %	100	95.8	90.3	80.1	67.3	48.9	39.9	34.2	31.1	18.2	12.4
Precision(3) %	100	100	90.4	80.9	75.7	55.9	49.4	47.6	40.6	20.7	17.1

average(5)	56.2%
average(3)	61.7%

Table 2. Precision at standard recall points for seven Queries

Unsatisfactory retrieval occurs for several reasons. First it is possible that the query is poorly designed. In this case the user can design a new query and re-submit. A second source of error is in matching itself. It is possible that locally the intensity surface may have a very close value. Many of these ’false matches’ are eliminated in the spatial checking phase. Errors can also occur in the spatial checking phase because it admits much more than a rotational transformation of points with respect to the query configuration. Overall the performance to date has been very satisfactory and we believe that by experimentally evaluating each phase the system can be further improved. The time it takes to retrieve images is dependent linearly on the number of query points. On a Pentium Pro-200 Mhz Linux machine, typical queries execute in between one and six minutes. However, the the primary limitations of the local matching technique are that it is relatively slow, and requires considerable disk space. Further, as presented the system cannot search for images in their entirety. That is does not address global similarity.

[†] The value $n(=10)$ is simply the retrievals up to recall n.

[‡] Based on personal communication with Bruce Croft

5. GLOBAL SIMILARITY RETRIEVAL

Global similarity retrieval enables the user to query a database for images as a whole. The applicability of whole image retrieval is evident in tasks such as similar scene retrieval, face images in a face database and trademark retrieval.

At first glance moment based techniques seem to lend themselves naturally to representing global shape and/or visual appearance. A signature for a particular shape is obtained by computing a set of moment invariants (invariant to affine or similarity transforms). These signatures may be made invariant to similarity or affine transforms by choosing the moment invariants appropriately. Different shapes may then be compared using these moment invariants. One could also obtain a signature for grey level images by computing the moment images over the grey level images. A number of researchers have used moment invariants to retrieve images. The QBIC project[4] uses moment invariants to recover shape. In the special case of trademark retrieval, where global similarity matching is useful and images are geometric and/or binary researchers have used moment invariants for retrieval.[28,3,16] Here we re-implement moment invariants and conclude that moment invariants do not really capture "appearance". They describe the spatial distribution of the intensity surface very coarsely and only seem to work reasonable with objects that have no holes in it.

We argue that global similarity can be determined by computing local features and comparing distributions of these features. This technique seems to give good results, and is reasonably tolerant to view variations. Schiele and Crowley[1] used such a technique for recognizing objects using grey-level images. Their technique used the outputs of Gaussian derivatives as local features. A multi-dimensional histogram of these local features is then computed. Two images are considered to be of the same object if they had similar histograms.

The choice of features often determines how well the image retrieval system performs. For example, directly using the intensity of each image pixel is often a poor choice. Thus, systems for retrieving color images often use features other than the RGB color values at each pixel. In the case of visual appearance what we are concerned with is characterizing the 3 dimensional intensity surface. A 3-dimensional surface is uniquely determined if the local curvatures everywhere are known. Thus, it is appropriate that one of the features be local curvature. However spatial orientation information is lost when constructing histograms of curvature alone. Therefore we augment the local curvature with local phase, and the representation is a histogram of local curvatures and phase. Curvature and phase are computed from responses to Gaussian derivative filters, and are generated for several scales. These are matched to retrieve images. Thus the distribution local derivative features is used to generate a global representation for global similarity by appearance.

Both these methods are tested using a trademark database of 2048 images obtained from the US Patent and Trademark Office. The images obtained from the PTO are binary and are converted to gray-level and reduced for the experiments. The curvature and phase implementation is also tested on the 1561 assorted images used for local similarity retrieval. Below the moment invariant and global appearance similarity retrieval methods are presented.

5.1. MOMENT TECHNIQUES

The strategy for retrieval is as follows. Invariant moments are computed from images and are thus described by a small set (seven) numbers. Then a query image is compared with each database image and a score based on the L2 norm is generated. All the images in the database are ranked by this score and returned. Below we briefly discuss computation of the invariant moment feature set and the matching formula.

Following is a brief review of moment invariants and the reader is referred to[8,21] for a full review. The implementation closely follows that described by,[16] where the authors use moment invariants for global similarity retrieval.

Given the intensity function of an image $f(x,y)$, which is assumed to be piecewise continuous and with compact support, one can define the regular moments $m_{pq} = \int\limits_{-\infty}^{+\infty}\int\limits_{-\infty}^{+\infty} x^p \ y^q \ f(x,y) \ dx \ dy$where $p, q = 1, 2, ...$. Central moments are defined as $\mu_{pq} = \int\limits_{-\infty}^{+\infty}\int\limits_{-\infty}^{+\infty} (x-\bar{x})^p \ (y-\bar{y})^q \ f(x,y) \ dx \ dy$ where $\bar{x} = \dfrac{m_{10}}{m_{00}}$ and $\bar{y} = \dfrac{m_{01}}{m_{00}}$. Invariant expressions, i.e. invariance to similarity transformations can be derived from the central moments. Let, $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$, and $\gamma = \frac{p+q}{2} + 1$. Then the following are moment some low order moment invariants invariant to similarity transformations of the image. Note that the moment invariants described here can in principle be applied to gray-level as well as binary images.

$$M_1 = \eta_{20} + \eta_{02} \qquad M_2 = [\eta_{20} - \eta_{02}]^2 + 4\ \eta_{11}^2$$

$$M_3 = [\eta_{30} - 3\ \eta_{12}]^2 + [3\ \eta_{21} - \eta_{03}]^2 \qquad M_4 = [\eta_{30} + \eta_{12}]^2 + [\eta_{21} + \eta_{03}]^2$$

$$\begin{aligned}
M_5 = {} & [\eta_{20} - \eta_{02}] * \left[[\eta 30 + \eta 12]^2 - [\eta_{21} + \eta_{03}]^2 \right] + 4\ \eta_{11} * [\eta 30 + \eta 12] * [\eta_{21} + \eta_{03}] \\
M_6 = {} & [\eta_{30} - 3\ \eta_{12}] * [\eta_{30} + \eta_{12}] * \left[[\eta 30 + \eta 12]^2 - 3 * [\eta_{21} + \eta_{03}]^2 \right] \\
& + [3 * \eta_{21} + \eta_{03}] * [\eta_{21} + \eta_{03}] * \left[3 * [\eta 30 + \eta 12]^2 - [\eta_{21} + \eta_{03}]^2 \right] \\
M_7 = {} & [3 * \eta_{21} - \eta_{03}] * [\eta_{30} + \eta_{12}] * \left[[\eta 30 + \eta 12]^2 - 3 * [\eta_{21} + \eta_{03}]^2 \right] \\
& + [3 * \eta_{12} - \eta_{30}] * [\eta_{21} + \eta_{03}] * \left[3 * [\eta 30 + \eta 12]^2 - [\eta_{21} + \eta_{03}]^2 \right]
\end{aligned}$$

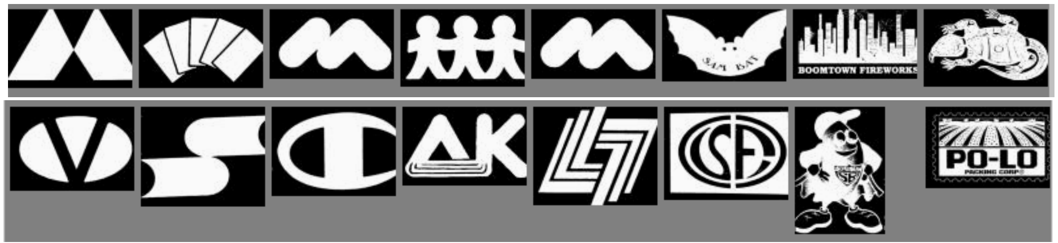


Figure 2. Trademark retrieval using moments

An image D_j of the database is compared using the L2 norm with a query image D_q and the L2 distance is written as $d_{qj} = \sum_{i=1}^{7} (M_{j,i} - M_{q,i})^2$ where $M_{j,i}, (i = 1, .., 7)$ represents the first seven invariants of a database image and $M_{q,i}, (i = 1, .., 7)$ represents the first seven invariants of the query image. Note that the moments have to be energy normalized so that they contribute equally to the L2 norm. The query image is compared with all database images and the resulting distances d_{qj} are sorted and images are displayed in the sorted order. In this implementation, and for evaluation purposes, the ranks are computed in advance, since every query image is also a database image.

Moment techniques work best when there is a single binary shape without holes in it. Figure 2(top row) shows an example of a good ranking achieve-able with the moment technique. The figure shows the 8 top ranked images (left-right) that are retrieved when the first image is used as a query. However, in practice, moment invariants suffer many serious problems. Many of them arise from the fact that moment techniques attempt to describe the image using a small set of numbers. However, it is difficult to describe an image properly using a small set of numbers. The reasonably good results in Figure 2 are attributable to the fact that the query is a solid shape without holes. In practice, the results are much worse than those shown here. As an example consider Figure 2 (bottom row) where poor results are obtained using this technique (compare with Figure 3 obtained using curvature and phase).

The central problem is that moment techniques preserve distribution information poorly i.e. they contain only gross information on how the image is spatially distributed. We argue that a robust global representation can be achieved by representing the distribution of robustly computed local features over the entire image. That is, the fundamental measurements made from the image are local. When the requirement is for finding parts of images, local features can be used directly. However, when the requirement is for global similarity retrieval, the distribution of local features can be represented using histograms and used for finding whole images. In the next section we elaborate on this argument to develop a global similarity retrieval using distributions of local geometric features, namely, curvature and phase, and demonstrate that they are a better representation of appearance.

5.2. CURVATURE AND PHASE

The normal and tangential curvatures of a 3-D surface (X,Y,Intensity) are defined as[5]:

$$N(\mathbf{p}, \sigma) = \left[\frac{I_x^2 I_{yy} + I_y^2 I_{xx} - 2 I_x I_y I_{xy}}{\left(I_x^2 + I_y^2 \right)^{\frac{3}{2}}} \right] (\mathbf{p}, \sigma) \quad T(\mathbf{p}, \sigma) = \left[\frac{\left(I_x^2 - I_y^2 \right) I_{xy} + (I_{xx} - I_{yy}) I_x I_y}{\left(I_x^2 + I_y^2 \right)^{\frac{3}{2}}} \right] (\mathbf{p}, \sigma)$$

Where $I_x(\mathbf{p},\sigma)$ and $I_y(\mathbf{p},\sigma)$ are the local derivatives of Image I around point $\mathbf{p}$ using Gaussian derivative at scale σ. Similarly $I_{xx}(\cdot,\cdot)$, $I_{xy}(\cdot,\cdot)$, and $I_{yy}(\cdot,\cdot)$ are the corresponding second derivatives. The normal curvature N and tangential curvature T are then combined[11] to generate a shape index as follows:

$$C(\mathbf{p},\sigma) = atan\left[\frac{N+T}{N-T}\right](\mathbf{p},\sigma)$$

The index value C is $\frac{\pi}{2}$ when $N = T$ and is undefined when either N and T are both zero, and is therefore not computed. This is interesting because very flat portions of an image (or ones with constant ramp) are eliminated. For example in Figure 4(middle-row), the background in most of these face images does not contribute to the curvature histogram. The curvature index or shape index is rescaled and shifted to the range $[0,1]$ as is done in.[2] A histogram is then computed of the valid index values over an entire image.

The second feature used is phase. The phase is simply defined as $P(\mathbf{p},\sigma) = atan2\left(I_y(\mathbf{p},\sigma), I_x(\mathbf{p},\sigma)\right)$. Note that P is defined only at those locations where C is and ignored elsewhere. As with the curvature index P is rescaled and shifted to lie between the interval $[0,1]$. Although the curvature and phase histograms are in principle insensitive to variations in scale, in early experiments we found that computing histograms at multiple scales dramatically improved the results. An explanation for this is that at different scales different local structures are observed and therefore multi-scale histograms are a more robust representation. Consequently a feature vector is defined for an image I as the vector $V_i = \langle H_c(\sigma_1)\ldots H_c(\sigma_n), H_p(\sigma_1)\ldots H_p(\sigma_n)\rangle$ where H_p and H_c are the curvature and phase histograms respectively. We found that using 5 scales gives good results and the scales are $1\cdots 4$ in steps of half an octave. Two feature vectors are compared using normalized cross-covariance defined as

$$d_{ij} = \frac{V_i^{(m)} \cdot V_j^{(m)}}{\left|\left|V_i^{(m)}\right|\right| \left|\left|V_j^{(m)}\right|\right|}$$

where $V_i^{(m)} = V_i - mean(V_i)$.

Retrieval is carried out as follows. A query image is selected and the query histogram vector V_q is correlated with the database histogram vectors V_i using the above formula. Then the images are ranked by their correlation score and displayed to the user. In this implementation, and for evaluation purposes, the ranks are computed in advance, since every query image is also a database image.

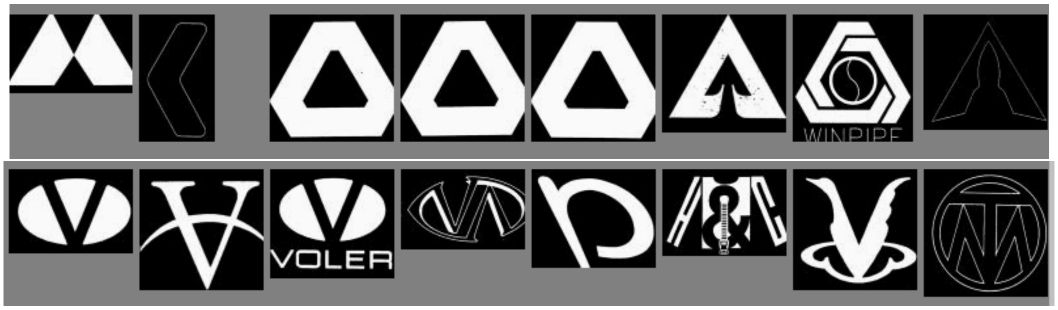


Figure 3. Trademark retrieval using Curvature and Phase

Experiments are conducted for both the trademark collection and the image collection. For each query in both databases the relevant images were decided in advance. These were restricted to 48. The top 48 ranks were then examined to check the proportion of retrieved images that were relevant. All images not retrieved within 48 were assigned a rank equal to the size of the database. That is, they are not considered retrieved. These ranks were used to interpolate and extrapolate precision at all recall points. Six queries were submitted each to the trademark and assorted image collection for the purpose of computing recall/precision. In general the query is of the form “find similar images to this one”. The judgment of relevance is qualitative. in the case of assorted images relevance is easier to determine and more similar for different people. However in the trademark case it can be quite difficult and therefore the recall-precision can be subject to some error. The recall/precision results are summarized in Table 3 and both databases are individually discussed below.

The curvature and phase method works much better in comparison to moment invariants on the trademark collection. For example, consider Figures 2 and 3. In both queries (top row, bottom row) the curvature and phase method works well, but the

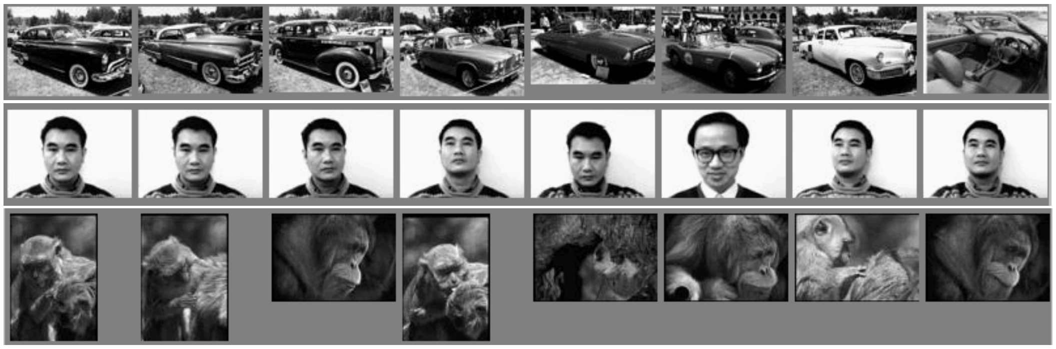


Figure 4. Image retrieval using Curvature and Phase

Table 3. Precision at standard recall points for six Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision(trademark) %	100	93.2	93.2	85.2	76.3	74.5	59.5	45.5	27.2	9.0	9.0
Precision(assorted) %	100	92.6	90.0	88.3	87.0	86.8	83.8	65.9	21.3	12.0	1.4

average(trademark)	61.1%
average(assorted)	66.3%

moment method works reasonably only on one of them. Six queries are submitted to the trademark retrieval and recall/precision is depicted in Table 3.

Experiments are also carried out with assorted gray level images. Ten queries are submitted for recall-precision, three of which are shown in Figure 4. The left most image in each row is the query and is also the first retrieved. The rest from-left to right are seven retrievals depicted in rank order. Note that, Flat portions of the background are never considered because the principal curvatures are very close to zero and therefore do not contribute to the final score. Thus for example, the flat background in Figure 4(second row) is not used. Notice that visually similar images are retrieved even when there is some change in the background (row 1). This is because the dominant object contributes most to the histograms. In using a single scale the poorer results are achieved and background affects the results more significantly.

The first three queries depicted here have an precision at average recall of 65% (car), 87.4% (face) and 64.2%(ape). In the face query the objective is to find the same face. The query “find similar faces” as described for the local case resulted in a 100at 48 ranks because there are far more faces than 48. Therefore it was not used in the final precision computation. The ape query results in several other light textured apes and country scenes with similar texture. Although these are not mis-matches they are not consistent with the intent of the query which is to find dark textured apes (see Table 1). Three other queries were submitted and described below

1. Find other patas monkeys. (47.1%) Compare this with the local similarity case. Here there are 16 patas monkeys in all and 9 within a small view variation. The local method cannot be expected to retrieve all of them, since the query was a face (see Table 1. However, here the whole image is being matched so the number of patas monkeys is 16. The precision is low because the method cannot distinguish between light and dark textures, leading to irrelevant images. Note, that it finds other apes, dark textured ones, but those are deemed irrelevant with respect to the query.

2. Given a wall with coca-cola painting find other coca-cola images (63.8%). This query clearly depicts the limitation of global matching. Although all three database images that had a certain texture of the wall (also had coca-cola logos) were retrieved (100% precision), there are two other very dissimilar images with coca-cola logos were not.

3. Scenes with Bill Clinton (72.8%). The retrieval in this case results in several mismatches. However, three of the four are retrieved in succession at the top and the scenes appear visually similar.

While the queries presented here are not “optimal” with respect to the design constraints of global similarity retrieval, they are however, realistic queries that can be posed to the system. Mismatches can and do occur. The first is the case where the global appearance is very different. The coca-cola retrieval is a good example of this. Second, mismatches can occur at the algorithmic level. Histograms coarsely represent spatial information and therefore will admit images with non-trivial deformations. The recall/precision presented here compares well with text retrieval. The time per retrieval is of the order of milli-seconds. In on going work we are experimenting with a database of 63000 images and the amount of time taken to retrieve is still less than a second. The space required is also a small fraction of the database. These are the primary advantages of global similarity retrieval. That is, to provide a low storage, high speed retrieval with good recall/precision.

6. CONCLUSIONS AND LIMITATIONS

This paper demonstrates retrieval of similar objects on the basis of their visual appearance. Visual appearance is characterized using filter responses to Gaussian derivatives over scale space.

More importantly global and local similarity matching can be achieved using the same notion of appearance. In addition, we claim that global representations are better constructed by representing the distribution of robustly computed local features. Currently we are investigating two issues. First is to scale the database up to about 100000 images and second is to provide a mechanism for combining global and local similarity matching in a single framework.

REFERENCES

1. bernt Schiele and James L. Crowley. Object recognition using multidimensional receptive field histograms. In Bernard Buxton and Roberto Cipolla, editors, *Computer Vision - ECCV '96*, volume 1 of *Lecture Notes in Computer Science*, Cambridge, U.K., April 1996. 4th European Conf. Computer Vision, Springer.
2. Chitra Dorai and Anil Jain. Cosmos - a representation scheme for free form surfaces. In *Proc. 5th Intl. Conf. on Computer Vision*, pages 1024–1029, 1995.
3. J.P. Eakins, K. Shield, and J.M. Boardman. Artisan: A Shape Retrieval System Based on Boundary Family Indexing. In J.K. Sethi and R.C. eds. Jain, editors, *Storage and Retrieval for Image Video and Databases IV*, volume **2670** of *Proc. SPIE*, pages 17–28, 1996.
4. Myron Flickner et al. Query by image and video content: The qbic system. *IEEE Computer Magazine*, pages 23–30, Sept. 1995.
5. L. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, University of Utrecht, Utrecht, Holland, 1993.
6. M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos 'at a glance'. In *Proc. 12th Int. Conf. on Pattern Recognition*, pages A459–A464, October 1994.
7. P. J. B. Hancock, R. J. Bradley, and L. S. Smith. The principal components of natural images. *Network*, 3:61–70, 1992.
8. M.K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, IT–8, 1962.
9. M Kirby and L Sirovich. Application of the kruhnen-loeve procedure for the characterization of human faces. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 12(1):103–108, January 1990.
10. J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.
11. J. J. Koenderink and A. J. Van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8), 1992.
12. J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.
13. Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
14. Fang Liu and Rosalind W Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. PAMI*, 18(7):722–733, July 1996.
15. W. Y. Ma and B. S. Manjunath. Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications*, 2(1):35–51, January 1996.
16. B.M. Methre, M.S. Kankanhalli, and W.F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing and Management*, 33, 1997.
17. S. K. Nayar, H. Murase, and S. A. Nene. Parametric appearance representation. In *Early Visual Learning*. Oxford University Press, February 1996.
18. A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of databases. In *Proc. Storage and Retrieval for Image and Video Databases II,SPIE*, volume 185, pages 34–47, 1994.
19. Rajesh Rao and Dana Ballard. Object indexing using an iconic sparse distributed memory. In *Proc. International Conference on Computer Vision*, pages 24–31. IEEE, 1995.
20. S. Ravela, R. Manmatha, and E. M. Riseman. Image retrieval using scale-space matching. In Bernard Buxton and Roberto Cipolla, editors, *Computer Vision - ECCV '96*, volume 1 of *Lecture Notes in Computer Science*, Cambridge, U.K., April 1996. 4th European Conf. Computer Vision, Springer.
21. T.H. Reiss. *Recognizing Planar Objects Using Invariant Image Features.*, volume 676 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.
22. C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 872–877, 1996.
23. D. L. Swets and J. Weng. Using discriminant eigen features for retrieval. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 18:831–836, August 1996.
24. Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, 1994.
25. M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive NeuroScience*, 3:71–86, 1991.
26. C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.
27. A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.
28. J.K. Wu, B.M. Mehtre, Y.J. Gao, P.C. Lam, and A.D. Narasimhalu. Star – a multimedia database system for trademark registration. In *Lecture Notes in Computer Science: Application of Database*, volume 819, pages 109–122, 1994.