A syntactic characterization of appearance and its application to image retrieval

R. Manmatha and S. Ravela

Center for Intelligent Information Retrieval
University of Massachusetts, Amherst, MA 01003

ABSTRACT

The goal of image retrieval is to retrieve images "similar" to a given query image by comparing the query and database using visual attributes like color, texture and appearance. In this paper, we discuss how to characterize appearance and use it for image retrieval.

Visual appearance is represented by the outputs of a set of Gaussian derivative filters applied to an image. These outputs are computed off-line and stored in a database. A query is created by outlining portions of the query image deemed useful for retrieval by the user (this may be changed interactively depending on the results). The query is also filtered with Gaussian derivatives and these outputs are compared with those from the database. The images in the database are ranked on the basis of this comparison. The technique has been experimentally tested on a database of 1600 images which includes a variety of images.

The system does not require prior segmentation of the database. Objects can be embedded in arbitrary backgrounds. The system handles 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, rmage 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.

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

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 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.

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.

Koenderink[7] and others[2] have argued that the local structure of an image at a particular scale can be represented by the outputs of a set of Gaussian derivative filters applied to an image. If Gaussian derivatives up to order N are used, this representation is called the local N-jet of an image at that particular scale.[7] A representation invariant to 2D rigid transformations can also be derived from the local N-jet.[2] In this paper, both these representations will be used to represent the visual appearance of an object.

The adequacy of this representation of appearance for image retrieval is shown by using correlation.[17,18] Briefly, the retrieval is carried out as follows. The database is filtered (off-line) with Gaussian derivatives. The query image is also filtered with Gaussian derivatives. A user defined rigid template obtained from the query image is then correlated with the database. The database images are then ranked using the correlation scores. It is observed that the images are ranked in order of similarity i.e. images which are similar to the query image are ranked ahead of other images.

Correlation is, however, not indexable and is, therefore, slow. Further the use of derivatives limits the matching to within 25 degrees. The following indexable technique is used for comparing images. The filter outputs are transformed so that they give rise to an invariant (to 2D rigid transforms) representation. For each invariant, an inverted file list is created which is indexed by the value of the invariant. The inverted file lists the location(s) in different images where that value occurs. At query time, for each point p_i in the query a set of probable matching points are obtained from each inverted file (there is one inverted file per invariant). The intersection of these probable matching points gives the set of points M_i which correspond to the query point p_i. The query points p_i have a particular spatial constraint. A spatial constraint is, therefore, enforced on the sets M_i which further reduces the number of possible matching points. The number of possible matching points/image provides a score which is then used to rank the images. Experiments show that the images are generally ranked in order of visual similarity.

The system performs entirely in a syntactic manner, that is, all the system does is to compare signals. Semantic information is implicitly provided by the user during query construction. Queries are constructed by the user by selecting regions from an example image (see Figure 1(a)). The regions selected should be strongly associated with the object and are determined by the user's considerable semantic knowledge about the world. Such semantic information is difficult to incorporate in a system. For example, the salient parts of a face are the eyes, nose and mouth. Again, the user may decide that the wheels are the most salient parts of cars and, therefore, the query should include them Figure 1(a). The association of wheels to cars is not known to the system, rather it is one that the user decides is meaningful. We believe that this natural human ability in selecting salient regions must be exploited. Further, in a fast system, feedback can be quickly obtained by browsing through the results (see Figure 2). If the results are unsatisfactory a new query can be designed.

Some of the salient features of our system include:

1. The ability to retrieve "similar" images. This is in contrast with techniques which try to recover the *same* object. In our system, a car used as a query will also retrieve other cars rather than retrieving only cars of a specific model.
2. The ability to retrieve images embedded in a background (see for example the cars in Figure 2 which appear against various backgrounds).
3. It does not require any prior manual segmentation of the database.
4. No training is required.
5. It can handle a range of variations in size.
6. It can handle 3D viewpoint changes up to about 20 to 25 degrees.

The system has been experimentally tested on a database of 1600 images which includes a variety of images including cars, faces, locomotives, apes and people embedded in various backgrounds.

The rest of the paper is organized as follows. Section 2surveys related work in the literature. In section 3.the notion of appearance is developed further and characterized using an N jet. A correlation measure for image retrieval is devised in section 4.2.Section 5discusses the construction of queries by the user. Section 6discusses an invariant form of the representation and shows how it may be used for indexing. Retrieval results for correlation are discussed in section 7.1while those for the indexable technique are given in section 7.2.The results are in the form of retrieved images in response to a query as well as recall and precision.

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[13] 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[5] face recognition is performed in.[22] In[20] 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[7] and its uniqueness in representing the scale space of a function[8,6,23,21] and the fact that the principal component of images are best described as Gaussians and their derivatives.[4] 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.[2] In[15] recognition is done by using a vector of Gaussian derivatives which are indexed. Schmid and Mohr[19] 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.[1,14] In[1] 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[9] the authors try to classify the entire Brodatz texture and in[3] attempt to classify scenes, such as city and country. Of particular interest is work by[10] 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[7] and others[2] that the local structure of an image I at a given scale σ can be represented using its N-jet. The N-jet of an image can be derived using a Taylor series expansion. Let a Gaussian of scale σ be given by:

$$G(\mathbf{r}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\mathbf{r}^2}{2\sigma^2}}. \tag{1}$$

where $\mathbf{r}$ is the coordinate of a point. A Gaussian filtered image $I_\sigma = I * G$ is obtained by convolving the image I with a Gaussian $G(\mathbf{r}, \sigma)$ i.e. given the intensity $I_g(\mathbf{p})$ at some point $\mathbf{p}$, 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. That is,

$$I(\mathbf{p} + \mathbf{r}) = \sum_{i,j} f_{i,j} I(\mathbf{p} + \mathbf{r}) * G_{x^i y^j}(\mathbf{r}, \sigma) \tag{2}$$

where

$$G_{x^i y^j}(\mathbf{r},\sigma) = \sigma^{i+j}\frac{\delta^{i+j}G(\mathbf{r},\sigma)}{\delta x^i \delta y^j} \tag{3}$$

is the energy normalized Gaussian derivative, $\mathbf{r} = (x, y)$, $f_{i,j}$ is the coefficient of the Taylor series and * denotes convolution.

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

$$J^N[I](\mathbf{p},\sigma) = (I(\mathbf{p}+\mathbf{r}) * G(\mathbf{r},\sigma), I(\mathbf{p}+\mathbf{r}) * G_x(\mathbf{r},\sigma), I(\mathbf{p}+\mathbf{r}) * G_y(\mathbf{r},\sigma), ..., I(\mathbf{p}+\mathbf{r}) * G_{x^i y^j}(\mathbf{r},\sigma), ...) \tag{4}$$

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

Consider two images of I_1 and I_2 of an object taken under identical conditions. Let their corresponding N-jets at scale σ be ${J_1}^N$ and ${J_2}^N$. Then due to noise the two N-jets will be identical only up to some finite value of N (say N = m). If other factors like small viewpoint variations and illumination are also included, m will be quite low. In general, as images become more dissimilar, their response vectors become less correlated. For similar images, the correlation will be strongest at the lowest frequencies, that is the low order derivatives will be correlated more than the high order derivatives. For example, Schmid and Mohr[19] have used 4 jets to compare two images of the same object under small viewpoint variations.

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:

$$J^2[I](\mathbf{p},\sigma) = \{I_\sigma(\mathbf{p}), I_{x,\sigma}(\mathbf{p}), I_{y,\sigma}(\mathbf{p}), I_{xx,\sigma}(\mathbf{p}), I_{xy,\sigma}(\mathbf{p}), I_{yy,\sigma}(\mathbf{p})\}^*$$

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.

3.1. A scale space description

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:

$$\left\{J^N[I](\mathbf{p},\sigma_1), J^N[I](\mathbf{p},\sigma_2) \ldots J^N[I](\mathbf{p},\sigma_k)\right\} \tag{5}$$

for some order N and 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.

The scale space is also useful for describing the varying sizes of objects. Many objects like the cars in the database appear at a range of sizes. For example, as an object moves from a camera (in depth) its image appears smaller. It is desirable that any retrieval system find objects at such varying sizes. As a consequence of the scale space description using Gaussian derivatives, image patches that are scaled versions of each other can also be compared in a straightforward manner. Consider two images

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

I_0 and I_1 that are scaled versions of each other (but otherwise identical). Without loss of generality assume that the scaling is centered at the origin. That is $I_0(\mathbf{p}) = I_1(s\mathbf{p})$ Then the following relations hold[11,12]

$$\begin{array}{ll} I_0(\mathbf{p}) \star G(\cdot,\sigma) & = I_1(s\mathbf{p}) \star G(\cdot, s\sigma) \\ I_0(\mathbf{p}) \star G_{x^i y^j}(\cdot,\sigma) & = I_1(s\mathbf{p}) \star G_{x^i y^j}(\cdot, s\sigma) \end{array} \tag{6}$$

where $G_{x^i y^j}(\cdot, s\sigma)$ is given by equation (3).

These equations state that if the image I_1 is a scaled version of I_0 by a factor s then in order to compare any two corresponding points in these images the filters must also be stretched (i.e. scaled) by the same factor. For example, if a point p_0 is being compared with a point p_1 in images I_0 and I_1 where I_1 is twice the size of I_0, then the filter used to compute the response at p_1 must be twice that of p_0 for the responses to be equal.

The multi-scale approach is, therefore, a robust representation of appearance which may be used to directly compare images that are scaled versions of each other.

4. RETRIEVAL USING CORRELATION

We now show that the local 2 jet at a particular scale provides a good description of appearance for retrieving images similar in appearance to a given query. The easiest way to show this is to use vector correlation.

4.1. Vector representations of an image

The key processing involves obtaining and matching vector-representations (VRs) of a sample gray level image patch S and a candidate image C. The steps involved in doing this will now be described:

A vector-representation $\vec{I}$ of an image I is obtained by associating each pixel with its local 2-jet. To account for changes in illumination, the zeroth order term of the 2-jet, that is $I * G$ is omitted from the vector-representation. More formally, $\vec{I}$ takes the form $\langle I_x, I_y, I_{xx}, I_{xy}, I_{yy} \rangle$ where I_x, I_y denote the the filter response of I to the first partial derivative of a Gaussian in direction x and y respectively. I_{xx},I_{xy} and I_{yy} are the appropriate second derivative responses.[16]

4.2. Matching using correlation

The correlation coefficient η between images $\vec{C}$ and $\vec{S}$ at location (m,n) in $\vec{C}$ is given by:

$$\eta(m,n) = \sum_{i,j} \hat{C_M}(i,j) \cdot \hat{S_M}(m-i, n-j) \tag{7}$$

where

$$\hat{S_M}(i,j) = \frac{\vec{S}(i,j) - S_M}{\left|\left|\vec{S}(i,j) - S_M\right|\right|}$$

and S_M is the mean of $\vec{S}(i,j)$ computed over S. $\hat{C_M}$ is computed similarly from $\vec{C}(i,j)$. The mean C_M is in this case computed at (m,n) over a neighborhood in C (the neighborhood is the same size as S).

Vector correlation performs well under small view variations. It is observed in[16,17] that typically for the experiments carried out with this method, in-plane rotations of up to 20^o, out-of plane rotation of up to 30^0 and scale changes of less than 1.2 can be tolerated. Similar results in terms of out-of-plane rotations were reported by.[15]

The database contains many objects imaged at several different scales. For example, the database used in our experiments has several diesel locomotives. The actual image size of these locomotives depends on the distance from which they are imaged and shows considerable variability in the database. The vector correlation technique described here cannot handle large scale (size) changes, and the matching technique, therefore, needs to be extended to handle large scale changes. This is discussed in.[16,17]

5. CONSTRUCTING QUERY IMAGES

The success of a retrieval in part depends on well designed queries. That implies that the user should be provided with a facility to design queries. Several other approaches in the literature take the entire feature set or some global representation over the entire image. While this may be reasonable for certain types of retrieval, it cannot necessarily be used for general purpose retrieval.

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.

6. AN INDEXABLE REPRESENTATION

Correlation is not indexable and, therefore, slow. An indexable representation can be derived by transforming the multi-scale feature vector (that is, the multi-scale N-jet) in equation 5 so that it is invariant to 2D rigid transformations. One reason for using invariants is that using the derivatives directly in a feature vector limits the viewing range (both out-of-plane and in-plane rotations of the image). Invariants partially address this problem.

6.1. Multi-Scale invariant vectors

Given the derivatives of an image I *irreducible differential invariants*, that are invariant under the group of displacements can be computed in a systematic manner.[2] The term irreducible is used because other invariants can be reduced to a combination of the irreducible set. The value of these entities independent of the choice of coordinate frame (up to rotations) for the low orders (two here) terms are enumerated.

The irreducible set of invariants up to order two of an image I are:

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

In experiments conducted in this paper, 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. The multi-scale invariant vector $D = \langle \Delta_{\sigma_1}, \Delta_{\sigma_2}, \Delta_{\sigma_3} \rangle$ is computed at sampled locations and the list of multi-scale vectors across the entire database is then indexed for rapid retrieval.

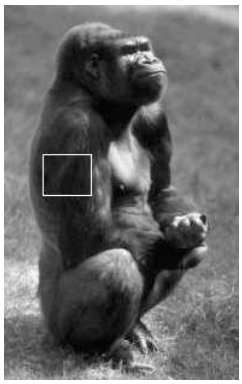
6.2. Indexing Invariant Vectors

The multi-scale invariant vector D can be viewed as a fixed length record. 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 in the previous sub-section generates an association between generalized coordinates and invariant vectors. This association can be viewed as a table $M : (i, x, y, D)$. The number of columns in this table are $3 + k$, where, k is the number of fields in an invariant vector. Each row is simply the invariant vector corresponding to a generalized coordinate and the number of rows is the total number of invariant vectors across the entire database.

To retrieve images, a 'find by value' functionality is needed, wherein, a query invariant vector can be found within M and the corresponding generalized coordinate is returned. A log-time solution for 'find by value' is obtained by generating inverted files (or tables) for M based on each field of the invariant vector. 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 that in M. Then, each $M_p^{'}$ is sorted by it's first column and a binary tree structure on this column is generated. As a result, the entire database is indexed.

The following steps are needed to perform a find-by-value operation on a query invariant vector. Each field of the query vector is used to traverse through the corresponding index tree. Once a match is found, the generalized coordinate is extracted.

(a) Car query



(b) Ape query: The user decides that the texture on the coat is important

Figure 1. Queries for retrieval by indexing

After all the k fields complete their search successfully, k generalized coordinates would have been extracted. If all of these are exactly the same then the find-by-value routine has succeeded.

The entire process of off-line computation can be summarized in the following steps: First, filter each image at uniformly sampled locations with Gaussian derivatives at several scales up to order two. Second generate the multi-scale invariants at these points and hence the table M. Third, compute the inverted file $M_k^{'}$ for each key of the record across the entire database. and finally sort the inverted file by key value and create a binary index.

6.3. Matching invariant vectors

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 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.

6.3.1. 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 $t = (t_1 \ldots t_k) > 0$. 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 $D_n - t < p < D_n + t$. This can be rewritten as

$$\&_{j=1}^{k} \left[D_n(j) - t(j) < p(i) < D_n(j) - t(j) \right]$$

where & is the logical and operator and k is the number of fields in the invariant vector. 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 largest entry smaller than $D_n(j) - t(j)$ and then a search for the upper-bound i.e. the smallest entry larger 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. To implement the logical-and part, 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.

Figure 2. The results of the car query shown in Figure 1(a)

In practice, the fields over which the intersection operation is performed is a matter of experimentation. For example, for several queries and those listed here, the last two fields of the invariant vector are used. This corresponds to six field searches or twelve traversals through the index trees.

6.3.2. 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

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

. Here x_n, y_n are the coordinates of the query entry Q_n and $c_{n_1} \ldots c_{n_f}$ 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:

$$score\,(u) \equiv \overset{max}{m}\ S_m\,(u) \tag{8}$$

where, $S_m\,(u) = \sum_{n=1}^{f} \mathcal{F}\,(u)_{m,n}$. 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.

7. EXPERIMENTS

The choice of images used in the experiments is based on a number of considerations. First it is general in that it doesn't reflect a bias towards any particular method, such as texture alone or shape alone. Second, it is expected that when very dissimilar images are used the system should have little difficulty in ranking the images. For example, if a car query is used with a database containing cars and apes, then it is expected that cars would be ranked ahead of apes. This is borne out by the experiments done to date. Much poorer discrimination is expected if the images are much more 'similar'. For example, different species of apes should be harder to discriminate.

The database used in this paper 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 but uncontrolled lighting and viewing geometry. Also, the objects of interest are embedded in natural scenes such as car shows, railroad stations, country-sides and so on.

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'. To that end one needs to evaluate if a query can be designed such that it captures the appearance of a generic steam engine or perhaps that of a generic car. Also, one needs to evaluate the performance of matching under a specified query. 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.

Experiments were done using both the correlation scheme and the indexing scheme.

7.1. Experiments with correlation

Experiments with several different queries were constructed to retrieve objects of a particular type. It is observed that under reasonable queries at least 60% of m objects underlying the query are retrieved in the top m ranks. Best results indicate retrieval results of up to 85%. This performance compares very well with typical text retrieval systems[†].

Several experiments were carried out with the database.[16,17] The results of the experiments carried out with a car query, a diesel query and a steam query are presented in table 7.1. The number of retrieved images in intervals of ten is charted in Table 7.1.The table shows, for example, that there are 16 car images "similar" in view to the car in the query and 14 of these are ranked in the top 20. For the steam query there are 12 "similar" images (as determined by a person), 9 of which are ranked in the top 20. Finally, for the diesel query there are 30 "similar" images, 12 of which are found in the top 20 retrievals. Pictorial results are shown in.[16,17]

Wrong instances of retrieval are of two types. The first is where the VR matching performs well but the objective of the query is not satisfied. In this case the query will have to be redesigned. The second reason for incorrect retrieval is mismatches due to the search over scale space. Most of the VR mismatches result from matching at the extreme relative scales.

[†]The average retrieval rate for text-based systems is 50%

	No. Retrieved Images				
Query	1-10	11-20	21-30	31-40	41-50
Car	8	6	1	0	1
Steam	7	2	1	0	2
Diesel	7	5	5	6	4

Table 1. *Correct retrieval instances for the Car, Steam and Diesel queries in intervals of ten. The number of "similar" images in the database as determined by a human are 16 for the Car query, 12 for the Steam query and 30 for the Diesel query.*

Overall the queries designed were also able to distinguish steam engines and diesel engines from cars precisely because the regions selected are most similarly found in similar classes of objects. As was pointed out in Section 5.query selection must faithfully represent the intended retrieval, the burden of which is on the user. Using the correlation metric presented here, the system performs it's stated purpose well: that is to extract objects of similar appearance which whose viewpoint is close to that of a query.

7.2. Experiments using the indexing scheme

The database and the considerations upon which the results were evaluated were the same as for the experiments using correlation. In this section we start out by demonstrating two retrieval examples and then go on to discuss the performance of the system in terms of recall and precision. Finally the typical computation times for running a query are presented.

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. Five queries were submitted to the database to compute the recall/precision shown in Table 2. These queries are enumerated below. For lack of space pictorial results are shown only for the first two.

1. Using the white wheel as the salient region find all cars with white wheels. This query is depicted in Figure 1(a). The top twenty five results of this query are shown in Figure 2 read in a text-book manner. Although, as it is clear from the results picture, several valid cars were found within reasonable viewpoint the user is only interested in white wheels and the average precision for this query is 48.6%.

2. This query is depicted in Figure 1(b). The user seeks to find similar dark textured apes including monkeys and points to the texture on this ape's coat. The average precision is 57.5% and the top 25 are shown in Figure 3. Note that although the 20^{th} image is a monkey (patas monkey), it is not a valid match in as far as the user is concerned because it is not a dark textured ape or monkey. Hence, it is not counted.

3. The third query is that of the face of a human and the user expects all human faces in the database. The average precision is 74.7%

4. The fourth query is that of the same human face but this time the user expects to obtain all the pictures of this particular person in the database. The average precision is 61.7%

5. The fifth query is that of the face of a Patas monkey and the user expects to retrieve all Patas monkeys whose faces are clearly visible. The average precision is 44.5%.

The recall/precision curve for all these queries together is shown is Table 2. The average precision over all the queries is 57.4%. This compares well with text retrieval where some of the best systems have an average precision of 50%[‡].

Table 2. Precision at standard recall points for Five Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision %	100	94.1	90.6	76.4	61.8	55.3	44.1	39.5	35.8	20.0	14.1

average	57.4%

[‡]Based on personal communication with Bruce Croft

Figure 3. 'Monkey' query results

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. Also Synapse allows users to drop any of the displayed results in to a query box and re-submit. Therefore, the user can not only redesign queries on the original image, but also can use any of the result pictures to refine the search. A second source of error is in matching generalized coordinates by value. The choice of scales in the experiments carried out in this case are $\frac{3}{\sqrt{2}}, 3, \frac{3}{\sqrt{2}}$ with the top two invariant vectors i.e. $\langle d_3, d_4 \rangle$. It is possible that locally the intensity surface may have a very close value, so as to lie within the chosen threshold and thus introduce an incorrect point. By adding more scales or derivatives such errors can be reduced, but at the cost of increased discrimination. 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.

8. CONCLUSIONS AND LIMITATIONS

This paper demonstrates retrieval of similar objects on the basis of their visual appearance. Visual appearance was characterized using filter responses to Gaussian derivatives over scale space. The adequacy of this representation was examined using correlation. Faster retrieval was achieved by using an indexable strategy. In addition, rotations in the image plane can be fully handled by transforming the derivatives into a representation invariant to 2D rigid rotations.

One of the limitations of our current approach is the inability to handle large deformations. The filter theorems described in this paper hold under affine deformations and it may be possible to use them to handle large deformations.

Finally, although the indexed implementation is some what slow, it is yet a remarkable improvement over correlation. We believe that by examining the the spatial checking and sampling components a further increase in speed is possible.

ACKNOWLEDGEMENTS

The authors wish to thank Adam Jenkins and Morris Hirsch for programming support and Prof. Bruce Croft and the Center for Intelligent Information Retrieval (CIIR) for continued support of this work.

REFERENCES

1. Myron Flickner et al. Query by image and video content: The qbic system. *IEEE Computer Magazine*, pages 23–30, Sept. 1995.
2. L. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, University of Utrecht, Utrecht, Holland, 1993.
3. 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.
4. P. J. B. Hancock, R. J. Bradley, and L. S. Smith. The principal components of natural images. *Network*, 3:61–70, 1992.
5. 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.
6. J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.
7. J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.
8. Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
9. 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.
10. W. Y. Ma and B. S. Manjunath. Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications*, 2(1):35–51, January 1996.
11. R. Manmatha. Measuring the affine transform using gaussian filters. In *Proc. 3rd European Conference on Computer Vision*, pages 159–164, 1994.
12. R. Manmatha and J. Oliensis. Measuring the affine transform – i: Scale and rotation. Technical Report CMPSCI TR 92–74,University of Massachusetts at Amherst,MA,1992. Also in Proc. of the Darpa Image Understanding Workshop 1993.
13. S. K. Nayar, H. Murase, and S. A. Nene. Parametric appearance representation. In *Early Visual Learning*. Oxford University Press, February 1996.
14. 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.
15. 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.
16. S. Ravela, R. Manmatha, and E. M. Riseman. Retrieval from image databases using scale space matching. Technical Report CS-UM-95-104, Computer Science Dept, University of Massachusetts at Amherst, MA, 1995.
17. 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.
18. S. Ravela, R. Manmatha, and E. M. Riseman. Scale space matching and image retrieval. In *Proc. DARPA Image Understanding Workshop*, 1996.
19. 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.
20. D. L. Swets and J. Weng. Using discriminant eigen features for retrieval. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 18:831–836, August 1996.
21. Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, 1994.
22. M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive NeuroScience*, 3:71–86, 1991.
23. A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.