Retrieving Images by Similarity of Visual Appearance*

S. Ravela R. Manmatha
Multimedia Indexing and Retrieval Group
University of Massachusetts, Amherst
{ravela,manmatha}@cs.umass.edu

Abstract

A system to retrieve images using a description of visual appearance is presented. A multi-scale invariant vector representation is obtained by first filtering images in the database with Gaussian derivative filters at several scales and then computing low order differential invariants. The multi-scale representation is indexed for rapid retrieval. Queries are designed by the users from an example image by selecting appropriate regions. The invariant vectors corresponding to these regions are matched with those in the database both in feature space as well as in coordinate space and a match score is obtained for each image. The results are then displayed to the user sorted by the match score. From experiments conducted with over 1500 images of objects embedded in arbitrary backgrounds, it is shown that images similar in appearance and whose viewpoint is within 25 degrees of the query image can be retrieved with an average precision[1] of 57.4%.

1 Introduction

The goal of image retrieval systems is to operate on collections of images and, in response to visual queries, extract relevant images. The application potential for fast and effective image retrieval is enormous, ranging from database management in museums and medicine, architecture and interior design, image archiving, to constructing multi-media documents or presentations[4]. However, there are several issues that must be understood before image retrieval can be successful. Foremost among these is an understanding of what 'retrieval of relevant images' means. Relevance, for users of a retrieval system, is most likely associated with semantics. Encoding semantic information into a general image retrieval system entails solving such problems as feature extraction, segmentation and, object and context recognition. These are extremely hard problems that are as yet unsolved. However, in many situations attributes associated with an image, when used together with some level of user input, correlate well with the kind of semantics that are desirable. Consequently, recent work has focused directly on retrieval using surface level image content descriptions such as color[20], texture features [10, 3, 14, 11], shape [12, 24] and combinations thereof [1, 5, 14].

In this paper images are retrieved using a characterization of the visual appearance of objects. The focus is on retrieving 'similar' objects. For example, when a face is presented as a query it is expected that the system should not only retrieve the same person's face but rank other faces before it ranks, cars, trains or apes. Similarly if a car is a query(see Figure 1) then it is expected that cars be ranked before faces or trains(see Figure 2). That is retrieved objects must appear visually similar. Intuitively, an object's visual appearance in an image depends not only on its three-dimensional geometric shape, but also on its albedo, its surface texture, the view point from which it is imaged, among other factors. It is non-trivial to separate the different factors that constitute an object's visual appearance. However, we posit that the shape of an imaged object's intensity surface closely relates to its visual appearance. Here a local characterization of the intensity surface is constructed and images are retrieved using a measure of similarity for this representation. The experiments conducted in this paper verify the association that objects that appear to be visually similar can be retrieved by a characterization of the shape of the intensity surface.

Different representations of appearance have been used in object recognition [13, 18] and have been applied to specific types of retrieval such as face recognition [6, 23]. To the best of our knowledge the system presented here is the first attempt to character-

*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 Trademark Office and 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.

[1] precision is the proportion of retrieved images that are relevant

ize appearance to retrieve similar images and in this paper the development of Synapse (Syntactic Appearance Search Engine), an image database search engine, is described. The approach taken here does not rely on image segmentation (manual or automatic) or binary feature extraction. Unlike some of the previously mentioned methods, no training is required. Since the representation is local, objects can be embedded in different backgrounds. Using an *example image and user interaction to construct queries*, Synapse retrieves similar images within small view and size variation in the order of their *similarity in syntactic appearance to a query.*

The claim is that, up to a certain order, the *local appearance* of the intensity surface (around some point) can be represented as responses to a set of scale parameterized Gaussian derivative filters (see Section 3). This set or vector of responses, called a multi-scale feature vector, is obtained solely from the signal content and without the use of "global context" or "symbolic interpretation". Further, the family of Gaussian filters are unique in their ability to describe the *scale-space* or *deep structure* [7, 9, 22, 2] of a function and are well suited for representing appearance.

In this paper we first verify, using correlation, that the proposed representation can retrieve visually similar images within small view and a range of size variation(see Section 4). Then, to overcome the limitations of the correlation approach, an indexable strategy for image retrieval is then developed using feature vectors constructed from combinations of the derivative filter outputs. These combinations yield a set of differential invariants [2] that are invariant to two-dimensional rigid transformations. Retrieval is achieved in two computational steps. During the off-line computation phase each image in the database is first filtered at sampled locations and then filter responses across the entire database are indexed(see Section 5). The run-time computation of the system begins with the user selecting an example image and marking a set of salient regions within the image. The responses corresponding to these regions are matched with those of the database and a measure of fitness per image in the database is computed in both feature space and coordinate space (see Section 5.2). Finally, images are displayed to the user in the order of fitness (or match score) to the query (see Section 7).

2 Related Work

Eigen-space representations [13, 6, 23, 21] are one of the earliest attempts to characterize appearance or the intensity shape. 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 involves training. The approach presented in this paper does not characterize appearance by eigen decomposition or any variation thereof. Further, the method presented uses no learning, does not depend on constant sized images, tolerates significant variation in background and retrieves from heterogeneous collections of images using local representations of appearance.

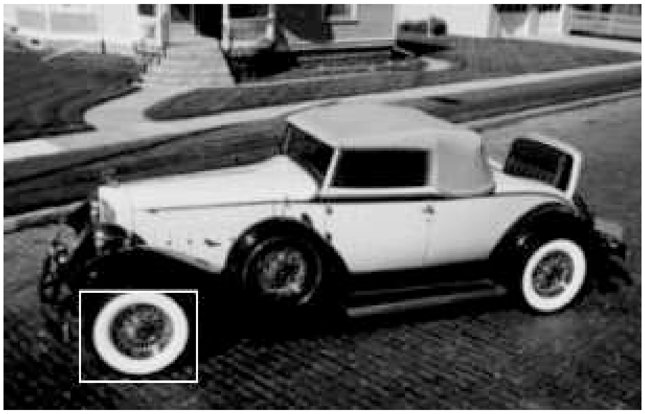

Figure 1: Allowing the user to construct queries by selecting the box shown

Gaussian derivative representations have been used in the context of recognition [15]. Indexed differential invariants have recently been used [18] for object recognition. We also index on differential invariants but there are several differences compared with [18]. First, the invariants corresponding to the low two order derivatives are used (as opposed to the first nine invariants), for reasons of speed as well as relevance to retrieving similar images(see section 3). 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.

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 cost-effective for most applications. Texture based image retrieval is also related to the appearance based work presented in this paper. Using Wold modeling[10] , the authors try to classify the entire Brodatz texture set and in [3] they attempt to classify scenes, such as city and country. Of particular interest is work by [11] who use Gabor filter representation (globally over the entire image) to retrieve by texture similarity.

3 Characterization of Appearance

This section begins by making explicit the notion of appearance and the uniqueness of Gaussian derivative filters therein. Then a representation, namely a multi-scale feature vector is constructed by filtering an image with a set of Gaussian derivative filters. The multi-scale feature vector are transformed so that the elements within this vector are invariant to 2D rigid transformations. This transformed feature vector is called the multi-scale invariant vector. Then a scheme for indexing multi-scale invariant vectors computed over the entire image database is presented. This completes all the steps of the off-line computation described earlier.

A function can be locally characterized by its Taylor series expansion provided the derivatives at the point of expansion are well conditioned. The intensity function of the image, on the other hand, need not satisfy this condition. However, it is well known that the derivative of a possibly discontinuous function can be made well posed if it is convolved with the derivative of a smooth test function [19]. Consider the normalized Gaussian as a choice for the smooth test function. Then the derivatives of the image $I_\sigma(\mathbf{x}) = (I \star G)(\mathbf{x}, \sigma), \mathbf{x} \in \Re^2, \sigma \in \Re^+$, are well conditioned for some value of σ. This is written as

$$I_{i_1 \ldots i_n, \sigma}(\mathbf{x}) = (I \star G_{i_1 \ldots i_n})(\mathbf{x}, \sigma)$$

$$G_{i_1 \ldots i_n} = \frac{\delta^n}{\delta_{i_1} \ldots \delta_{i_n}} G$$

and $i_k = x_1 \ldots x_D, k = 1 \ldots n$.

The local N-jet of $I(\mathbf{x})$ at scale σ and order N is defined as the set [8]:

$$J^N[I](\mathbf{x},\sigma) = \{I_{i_1 \ldots i_n, \sigma} | n = 0 \ldots N\} \qquad (1)$$

It can be observed that the set $\lim_{N \to \infty} J^N[I](\mathbf{x}, \sigma)$ bundles all the derivatives required to fully specify the Taylor expansion of I_σ up to derivatives of order N. Thus, for any order N, the local N-jet at scale σ locally contains all the information required to reconstruct I at the scale of observation σ up to order N. This is the primary observation that is used to characterize appearance. That is, up to any order the derivatives locally characterize the shape of the intensity surface, i.e. appearance, to that order. From the experiments shown in this paper it is also observed that this representation can be used to retrieve images that appear visually similar.

The choice of the Gaussian as the smooth test function, as opposed to others, is motivated by the fact that it is unique in describing the scale-space or deep structure of an arbitrary function. A full review of scale-space is beyond the scope of this paper and the reader is referred to [26, 7, 2, 9, 22]. Here some of the important consequences of incorporating scale space are considered. For increasing values of σ the Gaussian filter admits a narrowing band of frequencies and I will appear smoother. The scale-space of I is simply I_σ, where σ is the free variable. Similarly, the scale space of the derivatives of I is the range of $I_{i_1 \ldots i_n, \sigma}$ where σ is the free variable. Scale-space has an important physical interpretation in that it models the change in appearance of an imaged object as it moves away from a camera. An argument is therefore made for a *multi-scale feature vector* which describes the intensity surface locally at several scales. From an implementation stand point a *multi-scale feature vector* at a point p in an image I is simply the elements of the vector:

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

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. Multi-scale vectors represent appearance more robustly than a single-scale vector. This can viewed from several different perspectives. Since, multi-scale vectors are values computed at several different kernel sizes, therefore, they contain more information than fixed window operators. Equivalently, multi-scale vectors contain information at several different bandwidths and with the choice of a Gaussian accurately represent the intensity shape at different depths from the camera. From a practical standpoint this means that mis-matches due to an accidental similarity at a single scale can be reduced.

4 Verification Using Correlation

A measure of similarity between two feature vectors can be obtained by correlating them or computing the distance between the vectors. We begin with a simple approach wherein the feature vector is the local 2-jet without the zeroth order term, computed at a fixed scale [17]. Specifically, $\langle I_x, I_y, I_{xx}, I_{xy}, I_{yy} \rangle_\sigma$, computed at scale σ, is a derivative feature vector of an image where each pixel is associated with the first five partial derivatives (up to order two) computed in the neighborhood around that point. Using this representation in conjunction with correlation, we verify that, at any scale a reasonable retrieval of visually similar images is possible. Further, it is experimentally observed that the method tolerates small rotations and a range of size changes between a query patch and matching database images.

To compare a candidate image patch with a database image patch, their derivative feature vectors are correlated. The correlation coefficient η between the feature vectors of a query image patch $\vec{S}$ and those of a database image $\vec{C}$ at location (m,n) in $\vec{C}$ is given by:

$$\eta(m,n) = \sum_{i,j} \hat{C_M}(i,j) \cdot \hat{S_M}(m-i, n-j) \qquad (3)$$

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).

In order to retrieve images the following steps are employed. First, feature vectors are computed for each image in the database and stored. This is an off-line computation step. Then, during run-time the user marks regions in a query image and the derivative feature vectors of this query patch is correlated with the precomputed vectors for each database image. Finally, the results are presented to the user ranked by the correlation score. Note that similar images within the database occur at different sizes. While the above mentioned method as stated does not account for large relative size changes between a query and a matching database image, however, it has been extended to handle a range of size changes, discussed in [17].

From the experiments (see Section 7) the following observations are made. First, vector correlation performs well under small view variations. Typically, in-plane rotations of up to 20^o and out-of-plane rotations of up to 30^o can be tolerated. Second, a range of size variations, determined a priori, can be handled by searching across the scale parameter of the Gaussian. In particular similar objects within size changes of $\frac{1}{4} \ldots 4$ could be retrieved [17]. Finally it is observed that as images become more dissimilar their response vectors become less correlated, starting at the higher order. Thus, similar images can be expected to be more correlated in their lower order than higher ones.

5 Indexable Retrieval Strategy

There are several limitations to the correlation approach. First, correlation is computationally expensive. Second, using the derivatives directly in a feature vector restricts tolerance to rotations. Third, the use of vectors at a fixed scale can lead to mismatches due to accidental similarity solely as a result of the fixed scale of observation. These issues are partially addressed below. First, the derivative feature vector is transformed so that it is invariant to 2D rigid transformations. Second, correlation is replaced with an indexable strategy that results in an order of magnitude of speed increase and third vectors at multiple scales are used simultaneously to improve robustness. The arguments for the choice of lower order derivatives can be extended in the scale dimension as well. As images get dissimilar, they can be expected to retain strong correlation only at large scales (lower spatial frequency). Further the range of scales over which they correlate well gets smaller. As a consequence, in this paper the multi-scale vector is computed at three different scales placed half an octave apart.

5.1 Multi-Scale Invariant Vectors

Given the derivatives of an image I, *irreducible differential invariants* (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 is independent of the choice of coordinate frame (up to rotations) and the terms for the low orders (two here) are enumerated below.

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. 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. The brute force approach entails a linear search in M which is extremely time consuming. The solution is to generate inverted files (or tables) for M, based on each field of the invariant vector and index

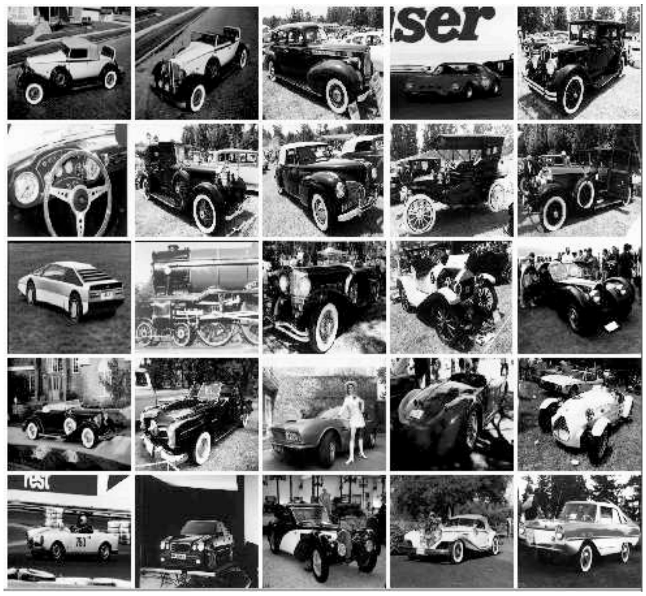

Figure 2: The results of the car query shown in Figure 1

them. Then the operation of 'find-by-value' can be performed in $\log(R)$ time (number of rows) and is described below.

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.

5.2 Matching Invariant Vectors

Run-time computation begins with the user marking selected 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, in order 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.

5.3 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.

5.4 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

$$\begin{aligned} A_n &= \left\langle x_n, y_n, c_{n_1}, c_{n_2} \ldots c_{n_f} \right\rangle \\ &= \left\langle x_n, y_n, (i_{n_1}, x_{n_1}, y_{n_1}) \ldots (i_{n_f}, x_{n_f}, y_{n_f}) \right\rangle \end{aligned}$$

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. In order to describe the fitness measure, two definitions are needed. First, define the distance between the coordinates of two query entries m and n as $\delta_{m,n}$. Second, define the distance between any two generalized coordinates c_{m_j} and c_{n_j} that are associated with two query entries m, n as $\delta_{c_{m_j}, c_{n_k}}$

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

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

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:

$$score\,(u) \equiv \overset{max}{m}\; S_m\,(u) \tag{4}$$

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. The value of the threshold is selected to reflect the rationale that similar images will have similar responses but not necessarily under a rigid deformation of the query points.

6 Query Construction

The ability for the user to construct queries by selecting regions is an important distinction between the approach presented here and elsewhere. Users can be expected to employ their considerable semantic knowledge about the world to construct a query. Such semantic information is difficult to incorporate in a system. An example of query construction is shown in Figure 1, where the user has decided to find cars similar to the one shown and decides that the most salient part are 'wheels'[2]. It is clear that providing such interaction removes the necessity for automatic determination of saliency. In the car example, the user provides the context to search the database by marking the wheel and retrieved images mostly contain wheels. The association of wheels to cars is not known to the system, rather it is one that the user decides is meaningful. Several other approaches in the literature take the entire feature set or some global representation over the entire image[1, 4, 21, 11]. While this may be reasonable for certain types of retrieval, it cannot necessarily be used for general purpose retrieval.Therefore, we believe that the natural human ability in selecting salient regions must be exploited. More importantly, letting the user design queries eliminates the need for 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.

7 Experiments

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 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. The choice of images reflects two primary considerations. First, the images should not reflect a bias towards any particular attribute and second, the system must be able to rank dissimilar images with little difficulty. This is confirmed by the experiments performed to date. Below the experiments conducted with correlation and the indexing methods are presented.

7.1 Experiments with Correlation

The first set of experiments are rotation tests. The Columbia image database (COIL-20) is used for the purpose of measuring rotation tolerance. In Figure 3 three pictures are shown with the left and right pictures rotated 20^o in either direction from the middle. A query is marked in the center picture as shown. Then vector correlation method is carried out over the entire set of images of this object. The position of the box on the left and right images indicate the location where the query patch correlates best. The tolerance to rotation is 20^o and all the 'anacin' pictures within this rotation from the center image (in 5^o increments) match successfully. The graph in Figure 4 depicts this result. The highest score is when there is no rotation and the curve drops gracefully as the rotation increases. These curves are shown for different values of σ of the Gaussian. Another interesting observation is depicted in the graph shown in Figure 5. Here the curves are labelled by the sampling of the query patch. That is, the correlation curves are plotted for the case when the derivative vectors for every pixel in the query patch is used, when the query is sampled in to a 7x7 region (49 samples), 5x5 and 3x3. These results indicate that the representation is robust so that a substantial increase in correlation speed can be achieved without significantly sacrificing speed. This is a motivation for the sampling approach used for indexing.

[2] see Figure 2 for the results

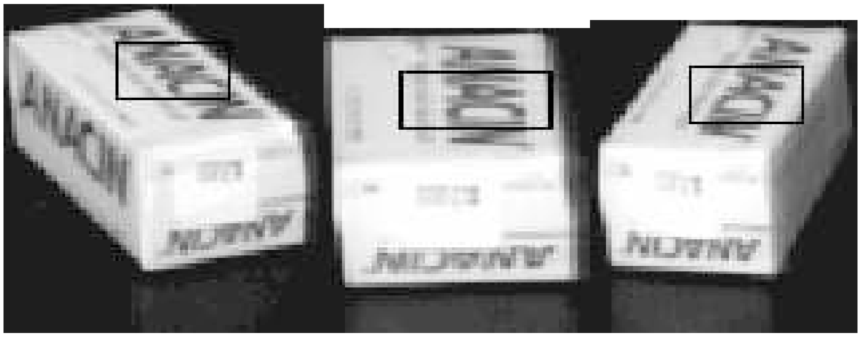

Figure 3: Correlation Under Rotation

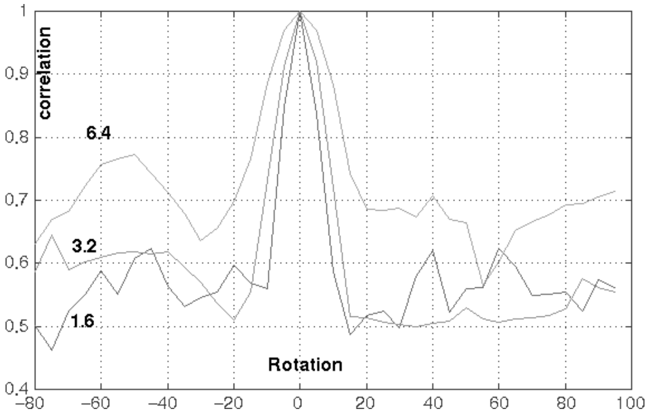

Figure 4: Correlation curves and Gaussian scale

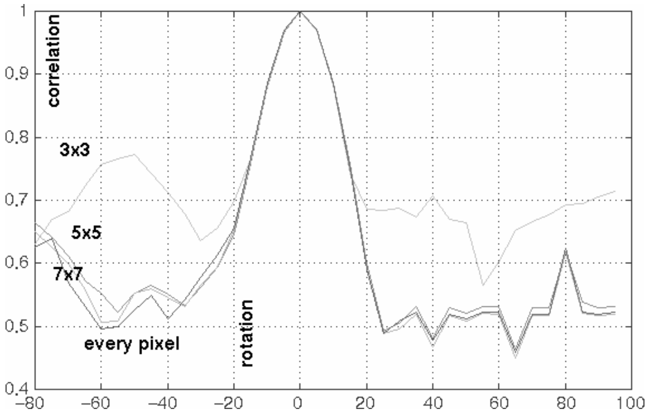

Figure 5: Correlation curves and query sampling

Rotation tests were carried out with all the objects in the COIL-20 database and the results suggest an average in-plane rotation tolerance of up to 20^o and out-of-plane tolerance of 30^o. While such rotation tests measure the degradation of correlation with rotations, the tolerance results can only be sound if these objects can be retrieved from a general database in the same rank order as suggested by the correlation curve. This test called, the embedded rotation test, was conducted. All the COIL-20 images were embedded in our database of 1561 images. Then a query similar to the one depicted in Figure 3 is posed and the results are observed. The results verify the above hypothesis. That is, objects within a small view variation were retrieved in rank order (correlated with increasing rotation from the query image).

The last set of experiments apply the correlation method to finding 'similar' images. 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[3]. In particular three experiments, 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 [17].

Wrong instances of retrieval are of two types. The first is where the correlation performs well but the objective of the query is not satisfied. In this case the

[3] 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.*

query will have to be redesigned. The second reason for incorrect retrieval is mismatches due to the search over scale space but using query vectors constructed at a fixed scale. Most of the mismatches result from matching at the extreme relative scales.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.

7.2 Experiments with Indexing

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 [25]. Consider as an example the query described in Figure 1. 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 2. 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[4]) is 48.6%. Five other queries that were also submitted are depicted in table 2. Due to lack of space detailed explanations are not provided and the reader is referred to [16] for details. The recall/precision table over these five queries is in Table 3. The average precision over all the queries is a 57.4%. This compares well with text retrieval where some of the best systems have an average precision of 50%[5].

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 into 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 $\frac{3}{\sqrt{2}}, 3, \frac{3}{\sqrt{2}}$. 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 and decreased generalization. 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, Limitations and Future Work

Within small view variations, images that are similar to a query are retrieved. These images are also observed to be visually similar and we posit that this method has good potential for image retrieval.

While a discussion of matching objects across different sizes was presented and has been implemented elsewhere [17] using correlation, in this paper, the multiscale invariant vector was used only to robustly characterize appearance. The next immediate step is to explicitly incorporate matching across size variations akin to the correlation approach.

A second important question is, what types of invariants should constitute a feature vector ? This is an open research issue. Finally, although the current system is some what slow, it is yet a remarkable improvement over our previous work. We believe that by examining the spatial checking and sampling components further increases in speed are possible.

References

[1] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Denis Lee, Dragutin Petkovix, Devid Steele, and Peter Yanker. Query by image and video content: The qbic system. *IEEE Computer Magazine*, 28(9):23–30, September 1995.

[2] Ludvicus Maria Jozef Florack. *The Syntactic Structure of Scalar Images.* PhD thesis, University of Utrecht, 1993.

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

[5] Based on personal communication with Bruce Croft

Table 2: Queries submitted to the system and expected retrieval

Given(User Input)	Find	Precision
Both wheels	White Wheeled Cars	57.0%
wheel, see Figure 1	White Wheeled Cars, see Figure 2	48.6%(see text)
Monkey's coat	Dark Textured Apes	57.5%
Face	All Faces	74.7%
Face	Same Person's Face	61.7%
Patas Monkey Face	All Visible Patas Monkey Faces	44.5%

Table 3: Precision at standard recall points for Five Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision %	100	95.1	88.7	76.8	61.8	55.3	44.8	38.5	34.8	21.0	14.2

average	57.4%

[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] Venkat N. Gudivada and Vijay V. Raghavan. Content-based image retrieval systems. *IEEE Computer Magazine*, 28(9):18–21, September 1995.

[5] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244, 1996.

[6] 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.

[7] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.

[8] J. J Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[9] Tony Lindeberg. *Scale-Space Theroy in Computer Vision.* Kluwer Academic Publishers, 1994.

[10] 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.

[11] W. Y. Ma and B. S. Manjunath. Texture-based pattern retrieval from image databases. *Multimedia Tools and Applications*, 2(1):35–51, January 1996.

[12] Rajiv Mehrotra and James E. Gary. Similar-shape retrieval in shape data management. *IEEE Computer*, 28(9):57–62, September 1995.

[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. Storafe and Retrieval of Image and Video Databases II*, volume 2, pages 34–47. SPIE, 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 and R. Manmatha. Image retrieval by appearance. In *(To appear SIGIR*, 1997.

[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] Cordelia Schmid and Roger Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. Computer Vision and Pattern Recognition.* IEEE, June 1996.

[19] L. Schwartz. Théorie des distributions. In *Actualités scientifiques et industrielles*, volume I,II, pages 1091–1122. Publications de l'Institut de Mathématique de l'University de Strasbourg, 1950-51.

[20] M. Strickler and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*, volume 2420 of *SPIE Proceedings Series*, pages 318–192, 1995.

[21] D. L. Swets and J. Weng. Using discriminant eigen features for retrieval. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 18:831–836, August 1996.

[22] Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, 1994.

[23] M. Turk and A. Pentland. Eigen faces for recognition. *Jrnl. Cognitive Neuroscience*, 3:71–86, 1991.

[24] A. Vailaya, Y. Zhong, and A. K. Jain. A hierarchical system for efficient image retrieval. In *Proc. Int. Conf. on Patt. Recog.*, August 1996.

[25] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[26] A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.