Retrieval from Image Databases using Scale-Space Matching*

S. Ravela R. Manmatha

E. M. Riseman

Computer Vision Research Laboratory
and
Center for Intelligent Information Retrieval
University of Massachusetts, Amherst, MA 01003
{ravela,manmatha}@cs.umass.edu

CMPSCI Technical Report UM-CS-95-104

November 1995.

Abstract

The retrieval of images from a large database of images is an important area of current research. Here, a technique to retrieve images based on appearance is proposed. The database is initially filtered with derivatives of a Gaussian at several scales. A user defined template is then created from an image of an object similar to those being sought. The template is also filtered using Gaussian derivatives. The template is then matched with the filter outputs of the database images and the matches ranked according to the match score. Experiments demonstrate the technique on a number of images in a database. No prior segmentation of the images is required and the technique works with viewpoint changes up to 20 degrees and illumination changes.

Keywords: Filter-based representations, Appearance-based representations, Scale-space Matching, Vector Correlation, Image Retrieval, Applications, Shape Representation.

*This work was supported in part by the Center for Intelligent Information Retrieval, NSF Grant IRI-92089020 and ARPA N66001-94-D-6054

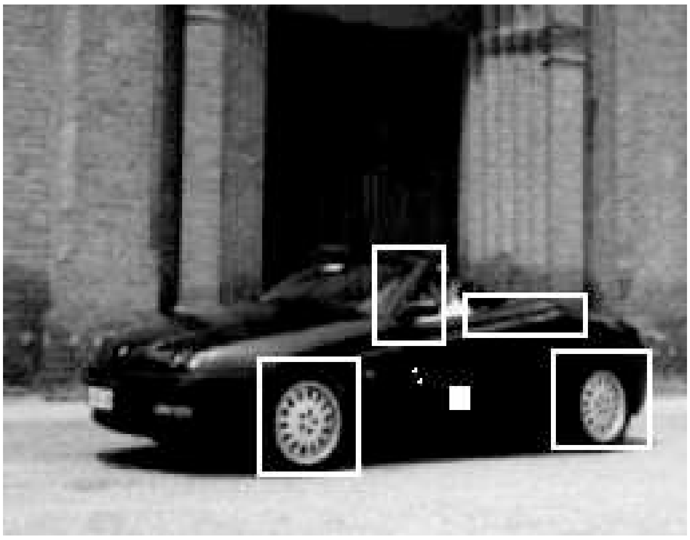


Figure 1: Construction of a query begins with a user marking regions of interest in an image, shown by the rectangles.

1 Introduction

The advent of multi-media and large image collections in several different domains brings forth image retrieval as an important issue. Applications of an image retrieval system range from database management in museums and medicine, architectural and interior design, image archiving, to constructing multi-media documents or presentations[8]. Some of the important challenges in building a retrieval system are the choice of image attributes (or features), their representations, query construction methods and matching techniques. Queries may be based on different attributes of an image [6], such as color distribution, motion, shape, structure, texture or perhaps user drawn sketches or even abstract token sets (such as points, lines etc.). Image retrieval can be viewed as an ordering of match scores that are obtained by searching through the database.

In this paper a method for retrieving images based on appearance is presented. Without resorting to token feature extraction or segmentation, images are retrieved in the order of their *similarity in appearance* to a *query*. Let us examine each of these terms more closely.

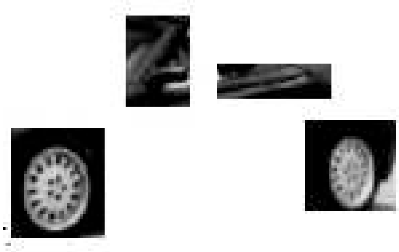


Figure 2: The regions of interest and their spatial relationships define a query.

A query is defined as user selected regions in an image, together with their spatial relationship. An example is shown in Figures 1 and 2. Here, the user wishes to retrieve images similar in view and shape (appearance) to the car shown in Figure 1. In order to do so, the user outlines salient regions (in his or her opinion) on the image(shown as rectangles in Figure 1). These regions along with their spatial relationship are conjunctively called as the query(Figure 2)[1].

Putting a user in the "loop" is one of the most important distinguishing features between the retrieval and recognition paradigms, because, the process of detection of features, their saliency and structure is left to the user. For example, only regions of the car in Figure 1 (namely, the wheels, side-view mirror and mid-section) are considered salient by the user are highlighted. Retrieved images must be similar in view, shape and distribution with respect to these regions. Automatic determination of feature saliency is an extremely hard problem and user input must be exploited when possible. It is observed that allowing users to pick salient features improves both the speed and accuracy of retrieval (see Section 5). An additional difference between the two paradigms is that user interaction can be used in a retrieval system of sufficient speed to evaluate the ordering of retrieved images and

[1] The retrieved images for this case are shown in Figure 10.

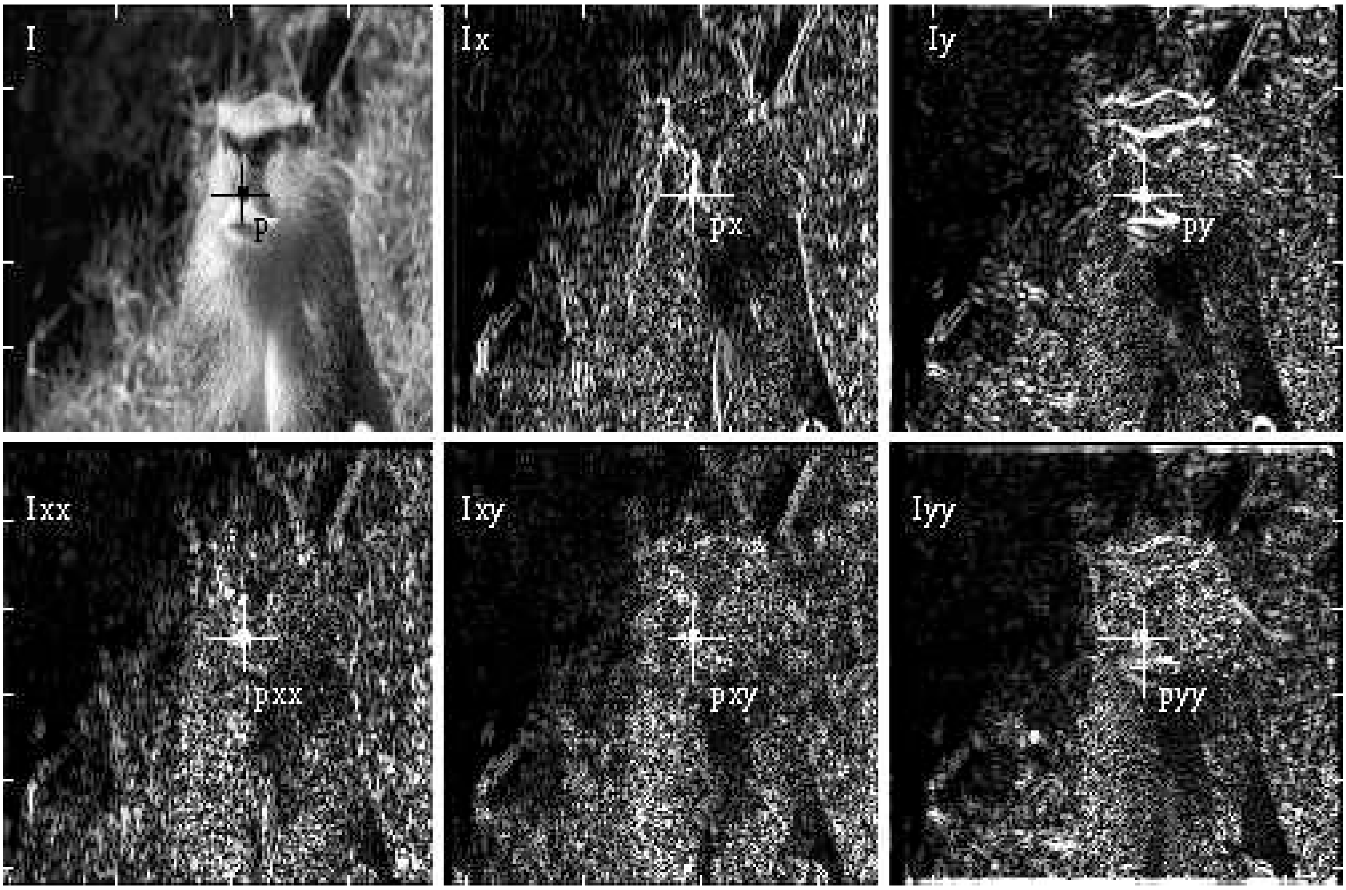


Figure 3: Vector Representation: The pixel p is associated by a vector $\langle p_x, p_y, p_{xx}, p_{xy}, p_{yy}\rangle$. The images $I_x \cdots I_{yy}$ are obtained by filtering I with each of the five partial derivatives of a Gaussian, up to the second order. The derivative pictures are enhanced for visibility.

reformulate queries if necessary. Thus, in the approach presented in this paper, alternate regions could be marked if the retrieval is satisfactory. Another important feature that sets retrieval apart is that, a hundred percent accuracy of retrieval is desirable but not critical. The user ultimately views and evaluates the results, allowing for tolerance to a few incorrect retrieval instances.

In order to measure the similarity of appearance between a query and an image, two issues must be addressed. First, appropriate representations of images must be chosen and second, a mechanism for matching these representations must be presented.

Filtered versions of images are used as representations of appearance. In particular a representation of an image is obtained by associating each pixel with a vector of responses to Gaussian derivative filters of several different orders. For example, Figure 3 contains a picture of an ape (I) that is filtered with each of the five spatial derivatives of a Gaussian(up to the second order), resulting in response-images I_x, I_y, I_{xx}, I_{xy} and I_{yy}. The vector-representation (VR) of a point p in I is, $v(p) = \langle p_x, p_y, p_{xx}, p_{xy}, p_{yy} \rangle$. The choice of Gaussians and their derivatives as a representation is motivated by a number of considerations. It has been argued by Koenderink and others that the structure of an image may be represented using Gaussian derivatives [12]. Hancock et al [9] have shown that the principal components of a set of images containing natural structures may be modeled as the outputs of a Gaussian and its derivatives at several scales. That is, there is a natural decomposition of an image into Gaussian derivatives at several scales. Gaussians and their derivatives have, therefore, been successfully used for matching images of the same object under different viewpoints [1, 24, 25, 10, 15, 20]. This paper is an extension to matching "similar" objects using Gaussian derivatives.

The retrieval scheme presented here is not meant to be a model of retrieval in biological systems. However, some of its components are similar to those used in the human visual system. The use of appearance based representations is in conformity with with both neurophysiological [4] and psychophysical [13] evidence that memory is accessed using imagery rather than symbolic information. In addition, it is known that filtering is performed using Gaussian derivatives (or similar filters) at multiple scales in the striate cortex [23]. Finally, it has been shown by Shepard and Cooper [21] that people warp visual matches to align and match them.

Images are matched by correlating their vector-representations. VR matching is robust to lighting variations and tolerates small variations in view. In addition, well designed queries have yielded significant variation in retrieved shapes(see Section 6). It is quite likely that structures similar to that of a query are present in the database at a different scale. As

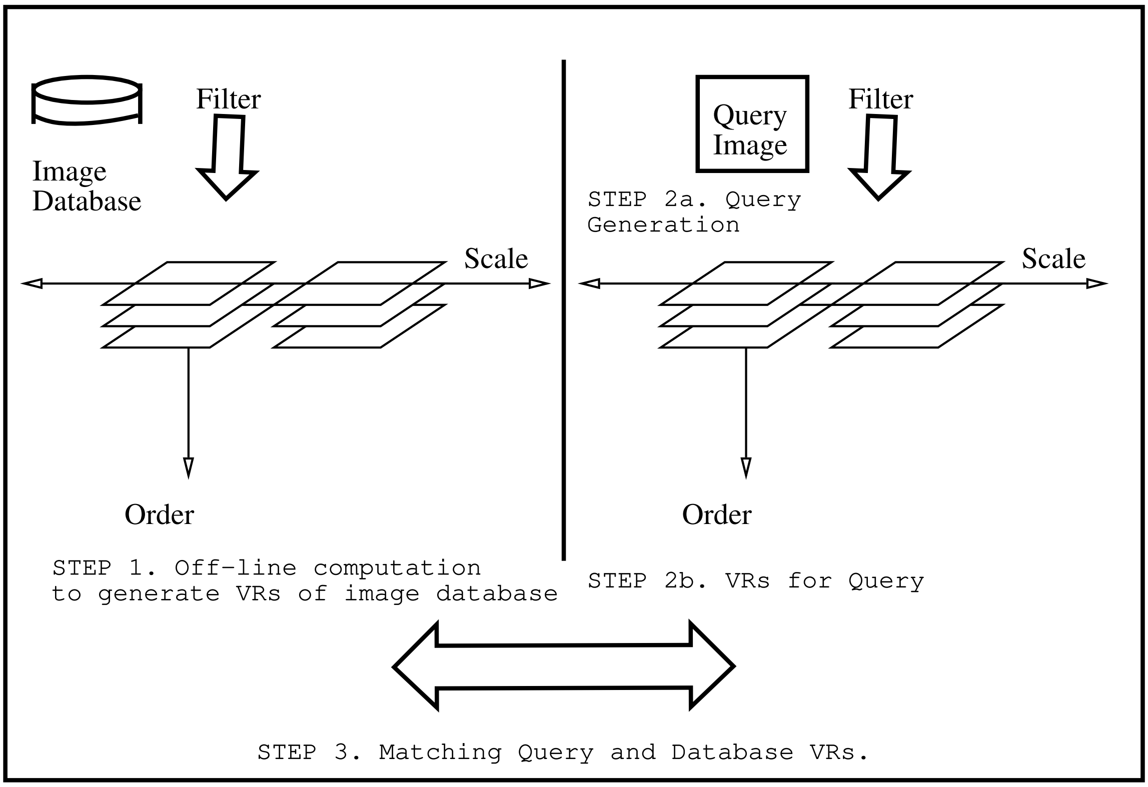

Filter
Image
Database
Query
Image
Filter
STEP 2a. Query
Generation
Scale
Scale
Order
Order
STEP 1. Off-line computation
to generate VRs of image database
STEP 2b. VRs for Query
STEP 3. Matching Query and Database VRs.

Figure 4: An overview of the retrieval method. There are three components. 1. Off-line computation of appearance representations of the database, 2. Query construction and 3. Matching appearance representation of the query and images in the database over scale-space.

described, the VR matching cannot account for gross changes in scale. VRs generated from filters at several scales are used to search over scale-space for possible scale variations of the query. The range of scale variation as well as the step size is a user controlled parameter. Scale-space matching is described in detail in Section 4.

From the above described representation scheme, matching technique and query construction methodology, the entire process of retrieval can be viewed as the following(see Figure 4) three component system. The first is an off-line computation step that generates vector-representations of database images for matching. The second is construction of queries and their VRs. The third is an ordering of images ranked by the correlation of their VRs with that of the query.

The remainder of this paper is organized as follows. In Section 2 other related approaches are examined. In Section 3 VR matching is described and it's performance is analyzed under view variations. In Section 4 VR matching is extended to account for scale variations. Then, in Section 5 query construction and the impact of user-selectivity on the accuracy of retrieval is discussed. In Section 6 retrieval experiments are presented, wherein retrieval is demonstrated on a database with over 300 images containing automobiles and trains (steam and diesel). These images obtained mainly over the internet have uncontrolled lighting and viewing geometry. Retrieval with a hit rate of at-least 60% is obtained. Conclusions are presented in Section 7.

2 Related Work

This paper is related to a number of threads in the literature. The first concerns matching using Gaussian derivative filters at multiple scales.

The idea of using Gaussian derivatives for matching and recovering local structure was suggested among others by Koenderink [12]. Among filter representations, Gaussian derivatives have a number of advantages - they are steer-able [5] and separable. The use of multiple derivative filters requires that correlation be performed between vectors. This is discussed by Granlund et al [7].

Some of the earliest uses of scale in matching go back to the Gaussian and Laplacian pyramids constructed by Burt and Adelson [2] and Crowley [3]. These pyramids have been used to do coarse to fine matching under translation, affine or more general transforms (see [1]). The pyramids speedup the computation as well as performing matching at the appropriate scales. However, as Lindeberg [14] in his extensive discussion of scale space and its properties, points out, they do not form a true scale space.

Kass [10] used the Gaussian and its derivatives at multiple scales for stereo matching. The notion of matching across Gaussians of different scales was used by Manmatha [15] for matching image patches under similarity and affine transforms. He also used the idea of comparing the outputs of Gaussians at different standard deviations to compute large scale changes. Rao and Ballard [20] used Gaussian derivatives at multiple scales to match a moving object when the viewpoint change was small.

The second thread to which our work is related is the area of image indexing and retrieval. To the best of our knowledge, retrieval on the basis of appearance or shape is almost entirely based on prior segmentation of the object. Examples include the QBIC project at IBM [6], the photo book project at the media lab [18] and shape retrieval [17]. These methods all require knowledge of the contour or binary shape of the object. For specific objects like faces, principal component analysis has been used successfully for representation [11] and retrieval [22]. Using texture measures, Picard et al [19] are able to classify images into a few distinct categories (eg city scene, country scene).

3 Matching Vector Representations

Recall that matching is performed by correlating the VR[2] of a query with VRs of the database images. In this section a VR is formally defined and the vector matching technique is presented. Then this technique is evaluated under varying viewing geometry.

Vector-representations of a sample gray level image patch S and a candidate image C are obtained as follows:

Consider a Gaussian described by it's coordinate $\mathbf{r}$ and scale σ

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

[2] VR is short for Vector Representation

A vector-representation $\vec{V}$ of an image I is obtained by associating each pixel with a vector of responses to partial derivatives of the Gaussian at that location. Derivatives up to the second order are considered. More formally, $\vec{V}$ 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.

In this paper, only the first and second derivatives of Gaussians are used. Consider Gaussian derivatives in 1-D. The odd derivatives are all correlated with each other. This also holds true for the even derivatives which are correlated with each other. However, for the same σ, the first derivative of a Gaussian is uncorrelated with the second derivative of a Gaussian [10]. Thus picking only the first and second derivatives of Gaussians insures that maximal information is extracted from the image. Gaussian (as opposed to Gaussian derivatives) filters are not used because they are sensitive to the actual intensity value.

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

$$\eta(m,n) = \sum_{i,j} \hat{C_M}(i,j) \cdot \hat{S_M}(m-i, n-j) \tag{2}$$

where

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

and C_M is the mean response over the area of C and S_M is the mean over the corresponding area around (m, n). $\hat{S_M}$ is computed similarly.

Vector correlation performs well under small view variations. The following example demonstrates the performance of vector correlation under different conditions. In the left-most frame of Figure 5 a region is extracted. This region is marked by a large black square. The white dot indicates the location of best match. The next two images are deformed versions of the original image. Note that "slice3.tif" does not produce a match exactly at the right

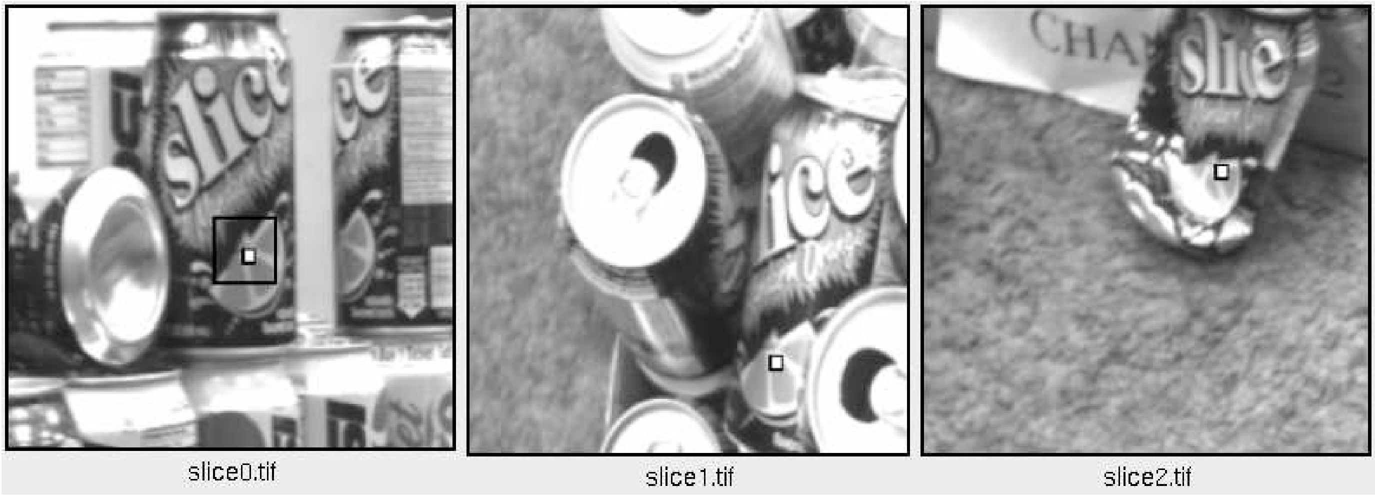


Figure 5: Performance of vector correlation under different conditions.

Image	Correlation
left-top	1.00
right-top	0.81
right-bottom	.52

Table 1: Vector matching under Image Deformations. Results for Figure 5.

location but is a match near it. Yet, given that there no other competing structure present in the image a location close to the original one is picked. In table 1 the correlation coefficient for each image is tabulated.

It is observed 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 [20].

4 Matching Across Scales

The database contains many objects imaged at several different scales. For example, the database used in the experiments has several diesel locomotives. The actual image size of these locomotives depends on the distance from which they are imaged and shows considerable

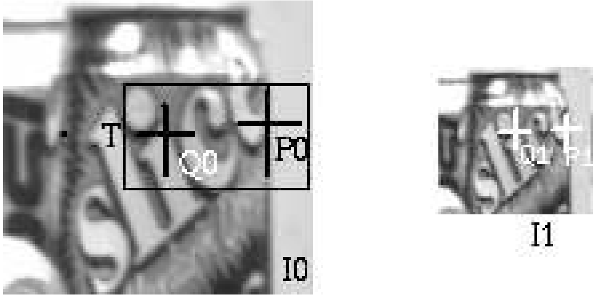


Figure 6: I1 is half the size of I0. To match points p_0 with p_1, Image I_0 should be filtered at point p_0 by a Gaussian of a scale twice that of the Gaussian used to filter image I_1 (at p_1). To match a template from I_0 containing p_0 and q_0, an additional warping step is required. See text in Section 4.

variability in the database. The vector correlation technique described in Section 3 cannot handle large scale changes. The matching technique, therefore, needs to be extended to handle large scale changes.

In Figure 6 image I_1 is half the size of image I_0 (otherwise the two images are identical). Thus,

$$I_0(\mathbf{r}) = I_1(s\mathbf{r}) \tag{3}$$

where $\mathbf{r}$ is any point in image I_0 and $s\mathbf{r}$ the corresponding point in I_1 and the scale change s = 2. In particular consider two corresponding points p_0 and p_1 and assume the image is Gaussian filtered at p_0 Then by substituting for I_0 using equation 3 we have:

$$\int I_0(\mathbf{r})G(\mathbf{r}-\mathbf{p_0},\sigma)d\mathbf{r} = \int I_1(s\mathbf{r})G(\mathbf{r}-\mathbf{p_0},s\sigma)d\,(s\mathbf{r}) * s^{-1} \tag{4}$$

But it can be shown that $G(\mathbf{r},\sigma) = G(s\mathbf{r},s\sigma)$ [15]. Thus,

$$\int I_0(\mathbf{r})G(\mathbf{r}-\mathbf{p_0},\sigma)d\mathbf{r} = \int I_1(s\mathbf{r})G(\mathbf{r}-\mathbf{p_1},\sigma)d\,(s\mathbf{r}) \tag{5}$$

In other words, the output of I_0 filtered with a Gaussian of scale σ at p_0 is equal to the output of I_1 filtered with a Gaussian of scale $s\sigma$ i.e. the Gaussian has to be stretched in the same manner as the image if the filter outputs are to be equal. This is not a surprising result if the output of a Gaussian filter is viewed as a Gaussian weighted average of the intensity. A more detailed derivation of this result is provided in [15].

The derivation above does not use an explicit value of the scale change s. Thus, equation 5 is valid for any scale change s. The form of equation 5 resembles a convolution and in fact it may be rewritten as a convolution

$$I_0(\mathbf{r}) \star G(., \sigma) = I_1(s\mathbf{r}) \star G(., s\sigma) \tag{6}$$

Similar derivations may also be carried out for higher derivatives of Gaussians (see [15]). Here the results for the first and second derivatives of Gaussians are listed. Define the normalized first derivative of Gaussian by

$$\mathbf{G}'(\mathbf{r}, s\sigma) = s\sigma \quad dG(\mathbf{r}, s\sigma)/d\mathbf{r} \tag{7}$$

The first derivative of the Gaussian has been energy normalized by the term $s\sigma$ so that its energy is the same as that of the Gaussian filter [24].

The normalized second derivative of Gaussian may be similarly defined by

$$\mathbf{G}''(\mathbf{r}, s\sigma) = (s\sigma)^2 \quad d^2G(\mathbf{r}, s\sigma)/d(\mathbf{r}\mathbf{r}^{\mathbf{T}}) \tag{8}$$

where the term $(s\sigma)^2$ again ensures that the energy of the second derivative Gaussian filter is the same as the energy of the first derivative Gaussian filter and the Gaussian filter.

Note that the first derivative of a Gaussian is a vector and the second derivative of a Gaussian a 2 by 2 matrix.

Then the Gaussian derivatives are related by (see [16])

$$I_s \star \mathbf{G}'(., \sigma) = I_0 \star \mathbf{G}'(., s\sigma) \tag{9}$$

and,

$$I_s \star \mathbf{G}''(., \sigma) = I_0 \star \mathbf{G}''(., s\sigma) \tag{10}$$

The above equations are sufficient to match the filter outputs (in what follows assume only Gaussian filtering for simplicity) at corresponding points (for example at $\mathbf{p_0}$ and $\mathbf{p_1}$). A further complication is introduced if more than one point is to be matched while preserving the relative distances (structure) between the points. Consider for example the pair of corresponding points $\mathbf{p_0}, \mathbf{q_0}$ and $\mathbf{p_1}, \mathbf{q_1}$. The filter outputs at points $\mathbf{p_0}, \mathbf{q_0}$ may be visualized as a template and the task is to match this template with the filter outputs at points $\mathbf{p_1}, \mathbf{q_1}$. That is, the template is correlated with the filtered version of the image I_1 and a best match sought. However, since the distances between the points $\mathbf{p_1}, \mathbf{q_1}$ are different from those between $\mathbf{p_0}, \mathbf{q_0}$ the template cannot be matched correctly unless either the template is rescaled by a factor of 1/2 or the image I_1 is rescaled by a factor of 2. The matching is, therefore, done by warping either the template or the image I_1 appropriately.

Thus, to find a match for a template from I_0, in I_1, the Gaussians must be filtered at the appropriate scale and then the image I_1 or the template should be warped appropriately. Now consider the problem of localizing a template T, extracted from I_0, in I_1(see Figure 6). For the purpose of subsequent analysis, assume two corresponding points ($\mathbf{p_0}$,$\mathbf{q_0}$) of interest in T and I_1 ($\mathbf{p_1}, \mathbf{q_1}$) respectively. To localize the template the following three steps are performed.

1. *Use appropriate Relative Scale:* Filter the template and I_1 with Gaussians whose scale ratio is 2. That is, filter T with a Gaussian of scale 2σ and I_1 with σ.

2. *Account for size change:* Sub-sample T by half. At this point the spatial and intensity relationship between the warped version (filtered and sub-sampled) of template points p_0 and q_0 should be exactly same as the relationships between filtered versions of p_1 and q_1.

3. *Translational Search:* Perform a translational search over I_1 to localize the template.

This three step procedure can be easily extended to match VRs of T and I_1 using Equations 9

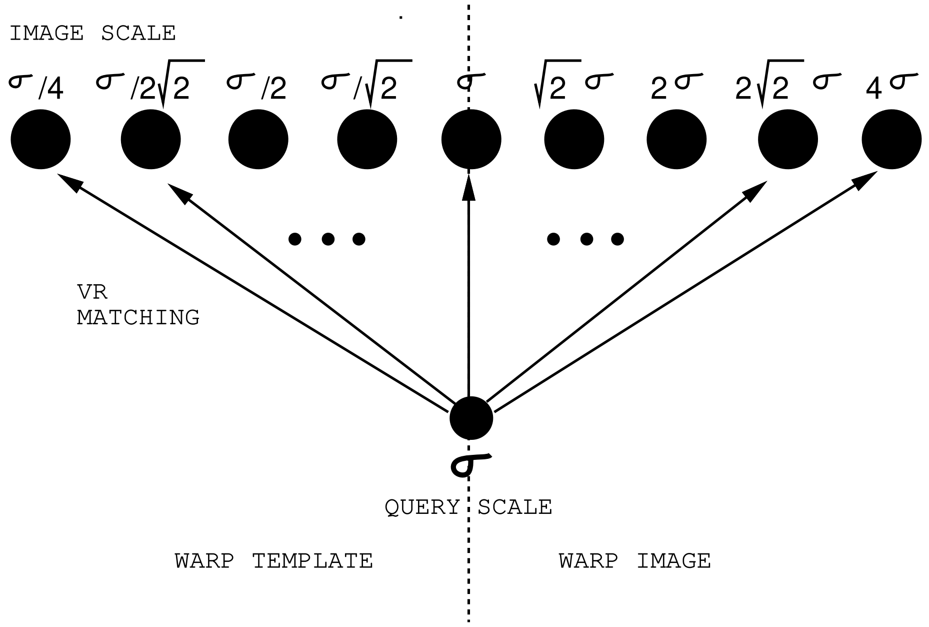


Figure 7: The process of scale space matching. The query is matched against an image VR list generated at several scales. The location of best correlation score over all scales is returned as the final match.

and 10. In step(1) generate VRs of T and I_1 using the mentioned filter scale ratios. In step(2) warp the VR of T instead of just the intensity. In step(3) use vector-correlation(Equation 2 at every step of the translational search.

Without loss of generality any arbitrary template T can be localized in any I_1 that contains T scaled by a factor s.

4.1 Matching Queries over Unknown Scale

The aforementioned steps for matching use the assumption that the relative scale between a template and an image is known. However, the relative scale between structures in the database that are similar to a query cannot be determined *a priori*. That is, the query could occur in a database image at some unknown scale. A natural approach would be to search over a range of possible relative scales, the extent and step size being user controlled parameters.

$Query$	3.2	2.2627	1.6	1.6	1.6	1.6	1.6	1.1313	0.8
$Image$	0.8	0.8	0.8	1.1313	1.6	2.2627	3.2	3.2	3.2
Scale Ratio	$\frac{1}{4}$	$\frac{1}{2\sqrt{2}}$	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	1	$\sqrt{2}$	2	$2\sqrt{2}$	4

Table 2: Filter scale values for the experiments carried out in this paper.

One way of accomplishing this is as follows (see Figure 7). First, VRs are generated for each image in the database over a range of scales, say $\frac{1}{4}\sigma$,$\frac{1}{2\sqrt{2}}\sigma$,...,$4\sigma$. Then, a VR for the query is generated using Gaussian derivatives of scale σ. The query VR is matched with each of the image VRs, thus traversing a relative scale change of $\frac{1}{4}$...4, in steps of $\sqrt{2}$. For each scale pairing the three step procedure for matching VRs is applied. In the warping step of this procedure either the query or the image is warped depending on the relative scale. If the relative scale between the query and a candidate image is less than 1 the candidate VR is warped and if it is greater than 1 the query VR is warped. In Figure 7 the process of filtering and warping is graphically depicted. Each of the blobs represent the relative scale of the filter. The arrows show VR matching. For the left half of this figure, the assumption is that the query image is larger than the candidate image and therefore the query is warped. For the right half the candidate image might have structures similar to, but larger than the query. Therefore, the image is warped. After the query is matched with each of the image VRs, the location in the image which has the best correlation score is returned.

In practice, VR lists are generated both for the query and database images to save computational cost, memory, and to avoid running in to filter discretization problems. For the experiments carried out in this paper the scales of the filters used for both the query and database images are in the range $[0.8 \cdots 3.2]$ in steps of $\sqrt{2}$. The pairings shown in Table 2 (read column-wise) describe the filter scale used for the query, the image and the entailed relative scale. As mentioned before if this value is less than one the image is warped, otherwise the query is warped.

It is instructive to note that VR lists over scale are scale-space representations in the sense described by Lindeberg [14] and also discussed by [7]. By smoothing an image with Gaussians at several different scales Lindeberg generates a scale-space representation. While VR lists are scale-space representations, however, they differ from Lindeberg's approach in two fundamental ways. First VRs are generated from derivatives of Gaussians and second, an assumption is made that smoothing is accompanied by changes in size (i.e. the images are scaled versions rather than just smoothed versions of each other). This is the reason warping is required during VR matching across scales.

On the other hand, the VR list approach should not be confused with pyramidal representations [2]. While pyramidal representations are also generated by filtering and sub-sampling images, there is an important distinction. Pyramids are generated as a translational search reduction mechanism for use in coarse-to-fine matching. Pyramid matching assumes that the scale of the template and the image within which it is being localized is the same. Therefore, matching the coarsest level of the image and template first followed successively by the finer representations yields reductions in translational search. On the other hand, the relative scale between the query and the image is never known, forcing a true search across the scale parameter. As Lindeberg points out recursive application of filters and sub-sampling as is done in pyramidal schemes is not in general a scale-space representation [14]. VR lists, which are not generated recursively, are proper scale-space representations and the matching occurs across scale-space.

With the components of the matching algorithms developed in the last two sections we now turn to a complete flow description of the retrieval technique, charted in Figure 8.

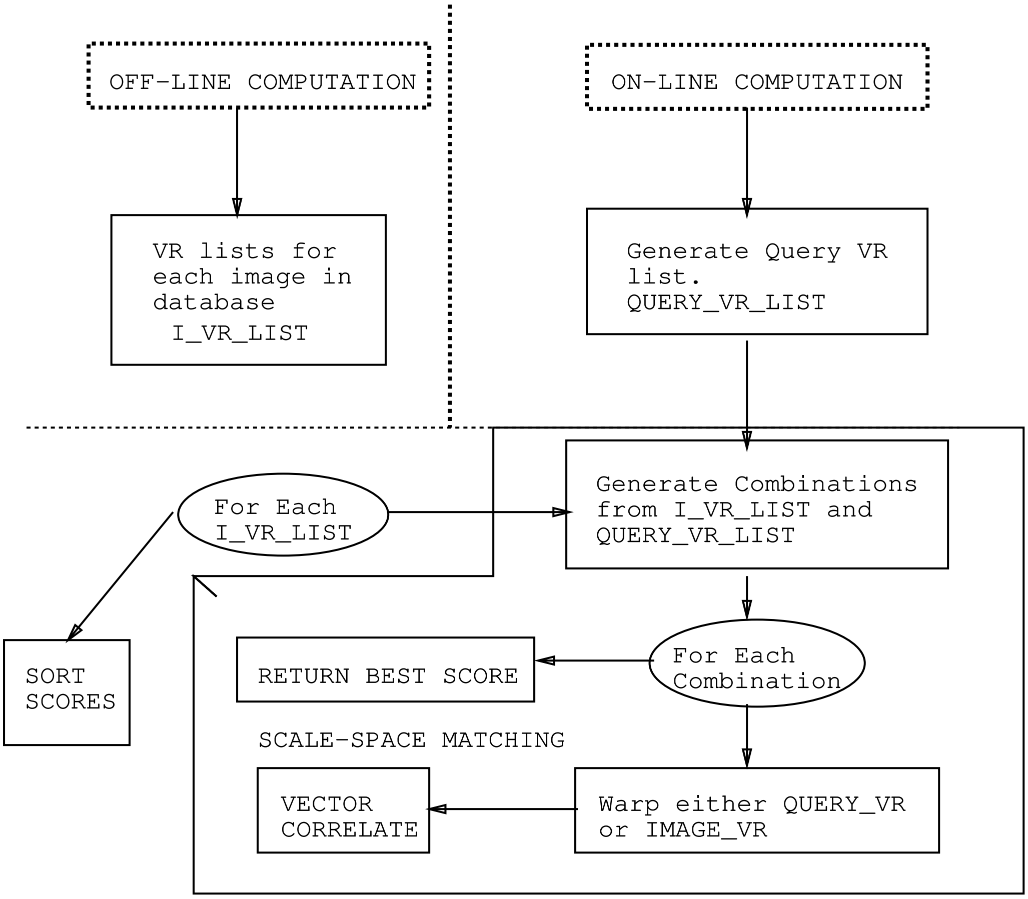

OFF-LINE COMPUTATION
ON-LINE COMPUTATION
VR lists for each image in database I_VR_LIST
Generate Query VR list. QUERY_VR_LIST
For Each I_VR_LIST
Generate Combinations from I_VR_LIST and QUERY_VR_LIST
SORT SCORES
RETURN BEST SCORE
For Each Combination
SCALE-SPACE MATCHING
VECTOR CORRELATE
Warp either QUERY_VR or IMAGE_VR

Figure 8: Algorithm to retrieve images

5 Constructing Query Images

The query construction process begins with the user marking salient regions on an object. VRs generated at several scales within these regions are matched with the database in accordance with the description in Section 4. Unselected regions are not used in matching. One way to think about this is to consider a composite template, such as one shown in Figure 2. The unselected regions have been masked out. The composite template preserves inter-region relationships and hence, the structure of the object is preserved. Warping the composite will warp all the components appropriately. That is, both the regions as well as distances between regions are scaled appropriately. Further, there are no constraints imposed on the selection of regions and the regions need not overlap.

It is claimed that putting a user in the loop has several advantages. First, the need for detecting the saliency of features on an object is alleviated. Second, the final results are evaluated by the user and if found unsatisfactory, new queries can be submitted.

Careful design of a queries, however, is important. Figure 9 shows an example of a poorly designed query. In the top left picture of this Figure, a region (black bounding box) is selected by the user, with the objective of retrieving all diesels of a similar class (within small view and shape variation). There are thirty diesel engines of this particular class in the database. The retrieved images are ranked in left-to-right (and top-to-bottom) fashion. The rectangular patches show the match locations of the centroid of the template. The query picture is also the best match. The top ten contain four correct retrievals. Twelve correct instances were present in the top thirty ranks resulting in a retrieval rate[3] of 40%. The reason for the less-than-satisfactory retrieval is that the user simply marked the entire front of the engine. Therefore a large number of pixels with low gradients are correlated, which increases the probability of matching coincidental structures. One approach to fixing this problem is

[3] The number of correct instances in the top m, where m is the number of similar objects in the database

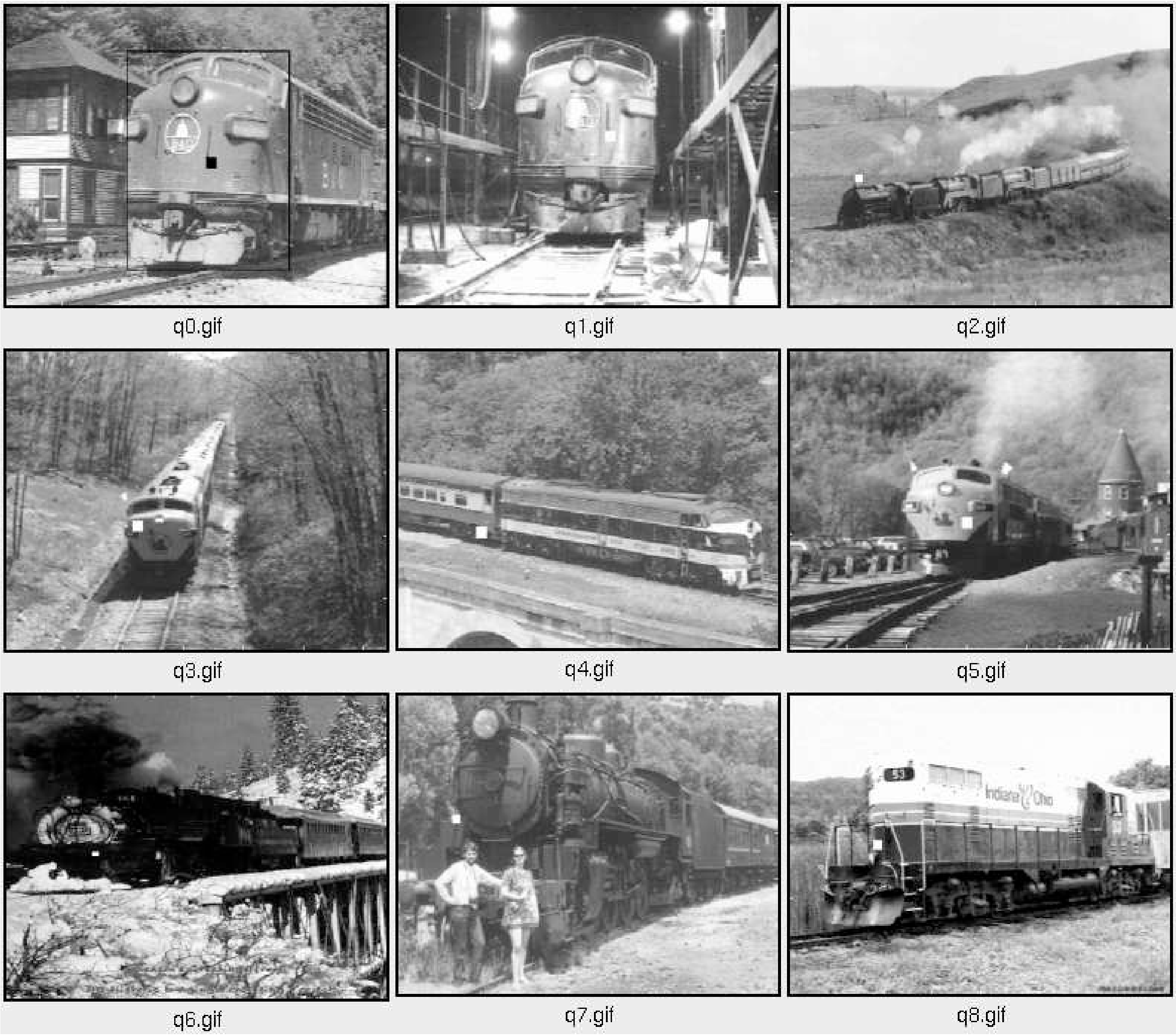


Figure 9: Example of a poor query selection.

to remove small gradients from the template. A second alternative is to redesign the query, and this is the strategy adopted in this paper.

A better designed query to retrieve this class of diesel engines is shown in Figure 12. The retrieval rate for this query is 60%. This query also executes faster since fewer pixels are used in VR matching. It is interesting to note that marking the entire object does not work very well. Marking extremely small regions has also not worked with this database. There are too many coincidental structures that can lead to poor retrieval.

Letting the user design queries makes the system very interactive. With sufficiently fast retrieval the user can be expected to quickly adapt and formulate interesting queries.

6 Experiments

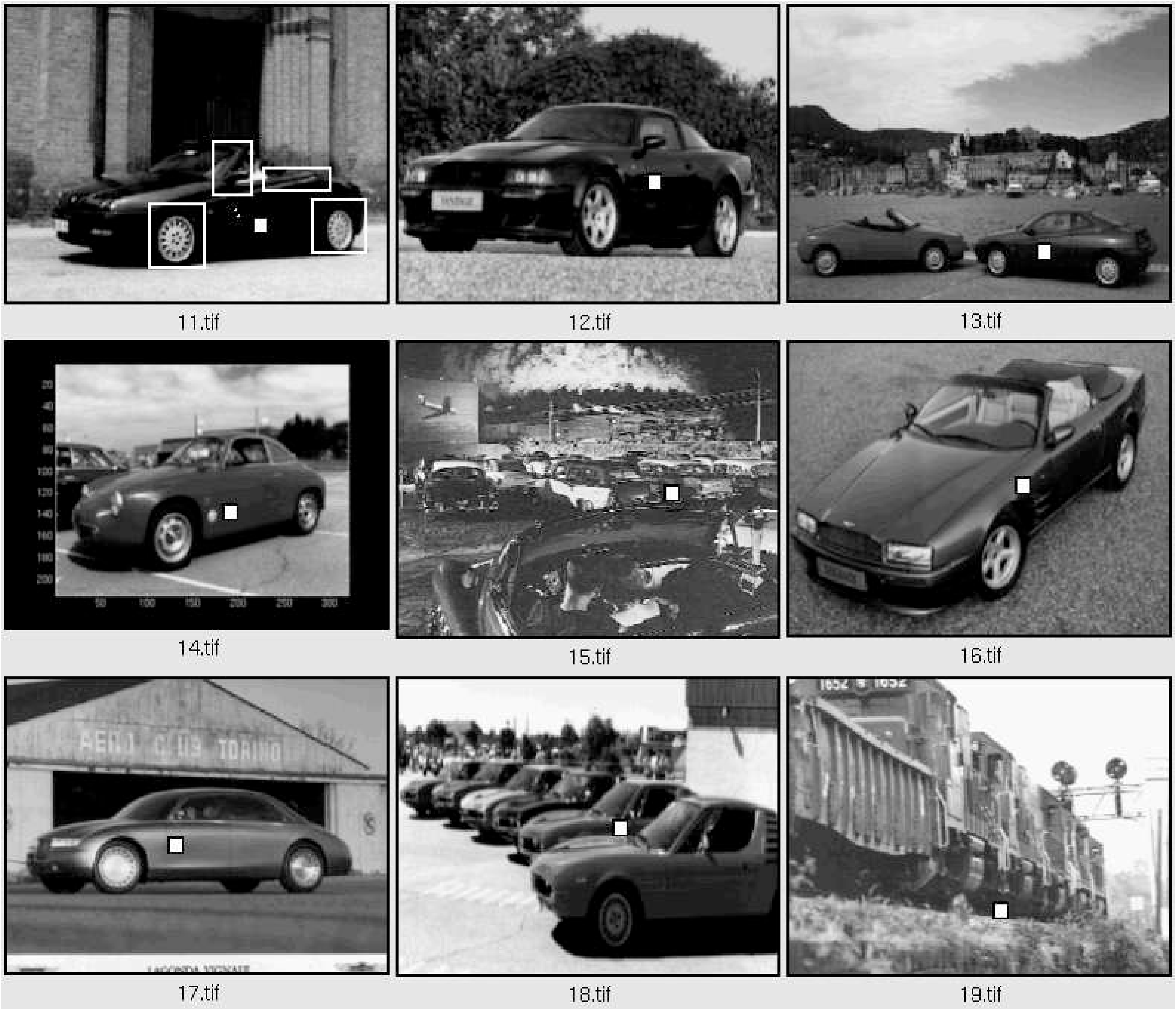
The database used in this paper has digitized images of cars, steam locomotives, diesel locomotives and a small number of other miscellaneous objects such as houses and apes. Over three hundred images were obtained from the internet to construct this database. These photographs, were taken with several different cameras of unknown parameters, and, under varying but uncontrolled lighting and viewing geometry. The objects of interest are embedded in natural scenes such as car shows, railroad stations, country-sides and so on.

The choice of images used in the experiments was based on a number of considerations. 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 limited number of experiments done. Much poorer discrimination is expected if the images are much more 'similar'. For example, man-made vehicles like cars, diesel and steam locomotives should be harder to discriminate. It was therefore decided to use a database consisting of images of cars, diesel and steam locomotives.

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 steams similar in appearance to this steam engine'. 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 VR 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.

Questions of this form are interesting to answer in the context of the types of images present in the database. Diesel locomotives, steam engines and cars are all man made objects and can be expected to be similar. From several experiments performed with this database it is observed that queries can be constructed, such that vector-matching does a good job of ordering the dissimilarities in appearance of these objects. For example, a car query that intuitively captures distinguishing features on a car ranks cars of similar appearance higher than other objects. Additionally, good discrimination is easily obtained between fairly dissimilar objects such as apes and engines for example. 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%.

Several experiments were carried out with the database supporting the three particular examples presented. The first is called *Car retrieval*, the second *Steam retrieval* and the third *Diesel retrieval*. Each of these cases is analyzed separately below.



11.tif 12.tif 13.tif

14.tif 15.tif 16.tif

17.tif 18.tif 19.tif

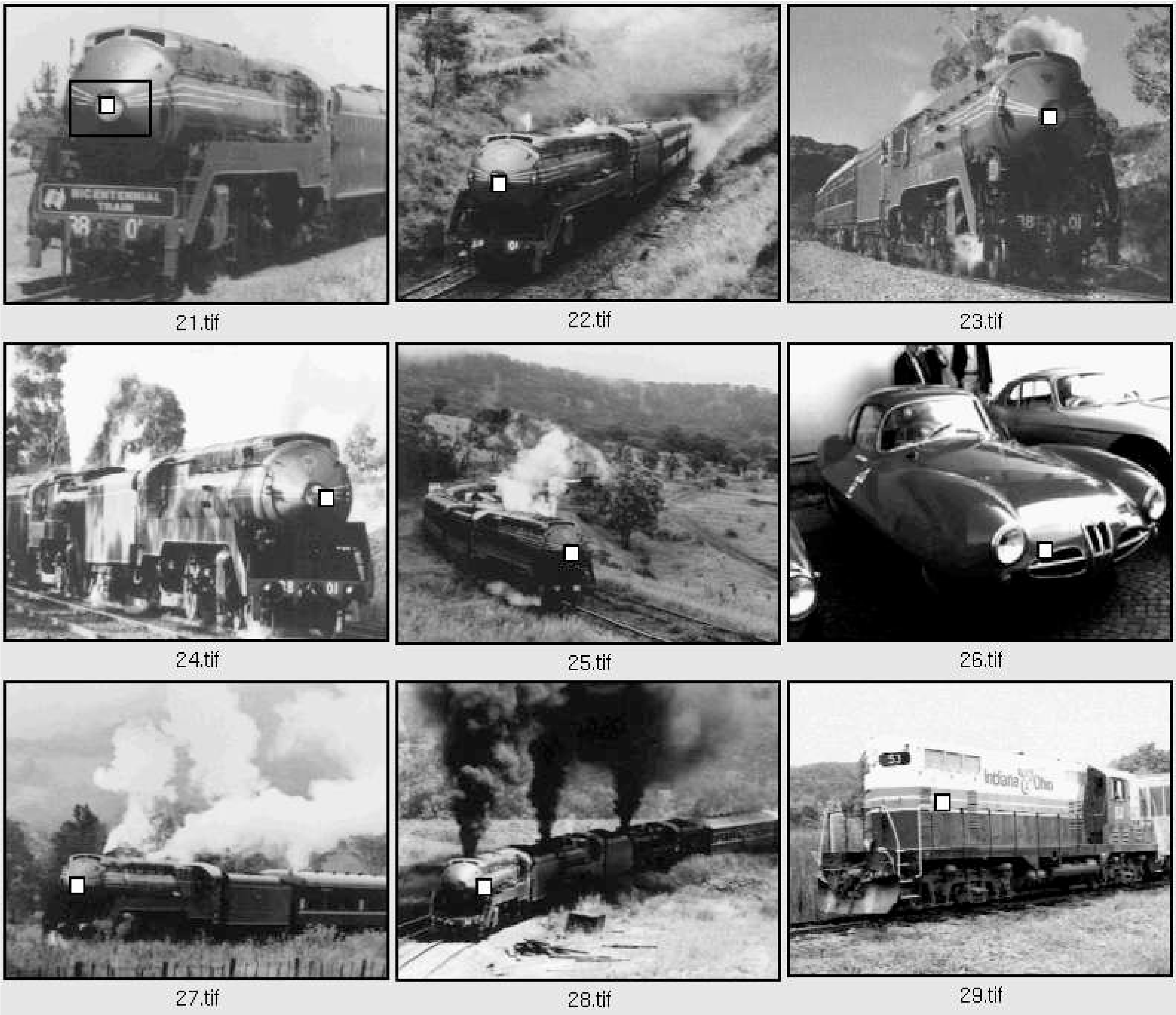
Figure 10: Retrieval results for Car.

6.1 Car Retrieval

The car image used for retrieval is shown in the top left picture of Figure 10. The objective is to 'obtain all similar cars to this picture'. Towards this end a query was marked by the user, highlighting the wheels, side view-mirror and mid section. The results to be read in text book fashion in Figure 10 are the ranks of the retrieved images. The white spots indicate the location of the centroid of the composite template at best match. In the database, there are exactly 16 cars within a close variation in view to the original picture. Fourteen of these cars were retrieved in the top 16, resulting in a 87.5% retrieval. All 16 car pictures were picked up in the top 50. The results also show variability in the shape of the retrieved instances. The mismatches observed in pictures labeled '15.tif' and '19.tif' occur in VR matching when the relative scale between the query VR and the images is $\frac{1}{4}$.

6.2 Steam Retrieval

The steam picture used for retrieval is labeled '21.tif' in Figure 11. Here the objective is to 'retrieve all steams that resemble the steam engine'. The user marks a query that is a distinguishing feature of this particular type of steam engine. Namely, the arrangement of the head-lamp and the stripes around it. There are 12 such engines in the database with varying view and scale. Eight of these were recovered in the top twelve, resulting in a retrieval of 65.7%. All of them were retrieved in the top fifty. It must be stated here that incorrect retrieval instances are of two types. The first, as is the case with picture labeled '26.tif', is that while this particular query matches the head-lamp of the car, it is an incorrect instance of retrieval since, only steam engines are desired. Picture '29.tif' was obtained at relative query-image scale of $\frac{1}{2\sqrt{2}}$ and is a mismatch.



21.tif 22.tif 23.tif

24.tif 25.tif 26.tif

27.tif 28.tif 29.tif

Figure 11: Retrieval results for Steam.

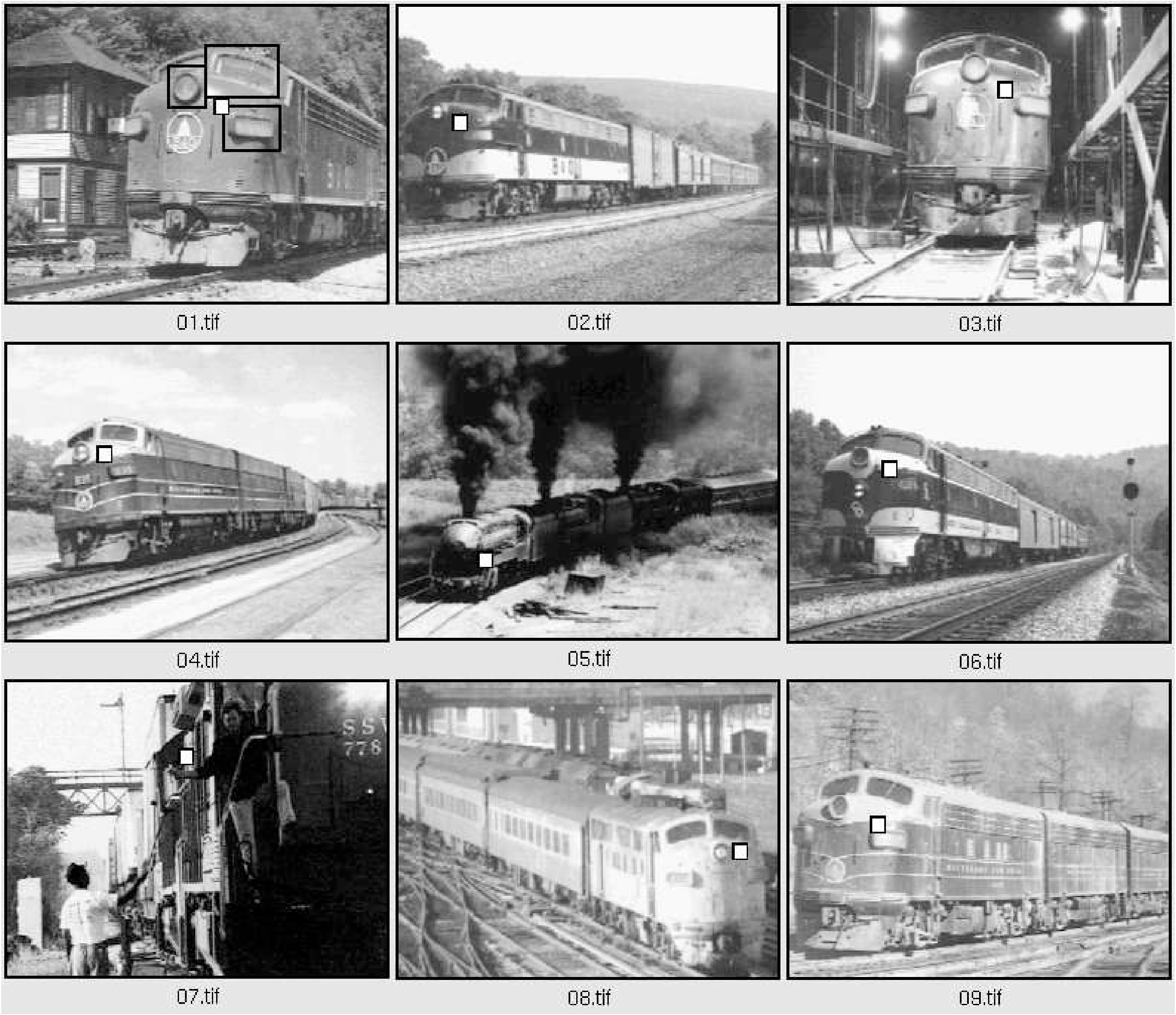


Figure 12: Retrieval results for Diesel.

Retrieved Image Nos.	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 3: Correct Retrieval instances for the Car, Steam and Diesel Queries in intervals of ten.

6.3 Diesel Retrieval

The picture of the diesel engine used to construct the query is labeled '01.tif' in Figure 12. The objective here is to 'retrieve all diesels of a similar class'. The query is designed by the user to capture this particular class of diesels. The database contains 30 diesels of this class within small view variations of the first. Seventeen of these were retrieved in the top 30 ranks. The mismatches occuring in pictures '05.tif' and '08.tif' are when the query VR is scaled $\frac{1}{4}$ times the VRs of these images. All diesels of this class were retrieved in the top sixty.

The number of retrieved images for each of these cases in intervals of ten is charted in table 6.3.

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. An example is the incorrect steam instances retrieved in Figure 9. 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. For example, the mismatch '05.tif' in the *diesel retrieval* occurs when the query is matched at a relative scale of 4.

Overall the queries selected were also able to distinguish steam engines, diesel engines and 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. The retrieval system presented here performs well under it's stated purpose: that is to extract objects of similar shape and view to that of a query.

7 Conclusions and Limitations

A method to retrieve images based on shape properties of images was presented. The vector-correlation algorithm is robust to lighting changes and small deformations. Vector-Correlation was extended to incorporate gross scale changes. Thus, the resulting representation of images is a proper scale-space representation and matching is performed over this space.

Using this technique objects of similar appearance were retrieved. There are several factors that affect retrieval results, including query selection, and the range of scale-space search. The results indicate that this method has sufficient accuracy for image retrieval applications.

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 a current step is to incorporate it in to the vector-correlation routine.

While these results execute in a reasonable time they are still far from the high speed performance desired of image retrieval systems. Work is on-going towards building indices of images based on local shape properties and using the indices to reduce the amount of translational search.

Acknowledgments

The authors thank Prof. Bruce Croft and the Center for Intelligent Information Retrieval (CIIR) for continued support of this work. We also thank Jonathan Lim and Robert Heller for systems support. The pictures of trains were obtained from John Hurst's home page at

http://www.cs.monash.edu.au/image_lib/trains/. The pictures of cars were obtained from the Team.Net home page at ftp://ftp.team.net/ktud/pictures/.

References

[1] J. R. Bergen, P. Ananadan, K. J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation", *Proc. Second European Conference on Computer Vision*, pp. 237-252, 1992.

[2] P. J. Burt, and E. H. Adelson, "The Laplacian pyramid as a compact image code", *IEEE Transactions on Communications*, 9:(4), pp. 532-540, 1983.

[3] J. L. Crowley, and A. C. Anderson, "Multiple Resolution Representation and Probabilistic Matching of 2-D Gray-scale Shape", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):113-121, 1987.

[4] Antonio R. Damasio, "Descartes' Error",*G. P. Putnam's Sons*, New York, 1994

[5] William T. Freeman, and E. H. Adelson, "The Design and Use of Steerable Filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891-906, September 1991.

[6] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovix, David Steele, and Peter Yanker,"Query By Image and Video Content: The QBIC System", *IEEE Computer Magazine*, pp.23-30, September 1995.

[7] Gösta H. Granlund, and Hans Knutsson, "Signal Processing in Computer Vision",*Kluwer Academic Publishers*, ISBN 0-7923-9530-1, Dordrecht, The Netherlands, 1995.

[8] Venkat N. Gudivada, and Vijay V. Raghavan, “Content-Based Image Retrieval Systems”, *IEEE Computer Magazine*, pp.18-21, September 1995.

[9] P. J. B. Hancock, R. J. Bradley and L. S. Smith, “The Principal Components of Natural Images”, *Network*, 3:61-70, 1992.

[10] M. Kass, ”Linear Image Features in Stereopsis”, *International Journal of Computer Vision*, Vol. 1, pp. 357-368, 1988.

[11] M. Kirby, and L. Sirovich, “Application of the Karuhnen-Loeve Procedure for the Characterization of Human Faces”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103-108, January 1990

[12] J. J. Koenderink, and A. J. van Doorn, “Representation of Local Geometry in the Visual System”, *Biological Cybernetics*, vol. 55, pp. 367-375, 1987.

[13] Stephen M. Kosslyn, and Oliver Konig, “Wet Mind: The new cognitive neuroscience”, *The Free Press*, 1992.

[14] Tony Lindeberg, “Scale-Space Theory in Computer Vision”, *Kluwer Academic Publishers*, ISBN 0-7923-9418-6 , Dordrecht, The Netherlands, 1994.

[15] R. Manmatha, “Measuring Affine Transformations Using Gaussian Filters”, *Proc. European Conference on Computer Vision*, vol II, pp. 159-164, 1994.

[16] R. Manmatha and J. Oliensis, “Measuring Affine Transform - I, Scale and Rotation”, *Proc. DARPA IUW*, pp. 449-458, Washington D.C., 1993.

[17] Rajiv Mehrotra and James E. Gary, “Similar-Shape Retrieval In Shape Data Management”,IEE Computer Magazine, pp. 57-62, ,September 1995.

[18] A. Pentland, R. W. Picard, and S. Sclaroff,"Photobook: Tools for Content-Based Manipulation of Databases", Proc. Storage and Retrieval for Image and Video Databases II, Vol.2, 185, SPIE, pp. 34-47, Bellingham, Wash. 1994.

[19] Monika M. Gorkani, and Rosalind W. Picard, "Texture Orientation for Sorting Photos "at a Glance"", *TR-292, M.I.T., Media Labortory, Perceptual Computing Section*, 1994.

[20] R. Rao, and D. Ballard, "Object Indexing Using an Iconic Sparse Distributed Memory", *Proc. International Conference on Computer Vision*, pp. 24-31, 1995.

[21] R. N. Shephard, and L. A. Cooper, "Mental Images and their transformations", MIT Press, Cambridge, MA, 1982.

[22] M. Turk, and A. Pentland, "Eigen Faces for Recognition", *Journal of Cognitive Neuroscience*, vol 3., pp.-71-86, 1991.

[23] R. L. De Valois, and K. K. De Valois, "Spatial Vision", *Oxford University Press*, New York, 1988.

[24] P. Werkhoven and J. J. Koenderink, "Extraction of Motion Parallax Structure in the Visual System 1", *Biological Cybernetics*, 1990.

[25] P. Werkhoven and J. J. Koenderink, "Extraction of Motion Parallax Structure in the Visual System 2", *Biological Cybernetics*, 1990.