Finding Images of Landmarks in Video Sequences

Yataka Takeuchi and Martial Hebert

{takeuchi,hebert}@ri.cmu.edu
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213

Abstract

Recognizing landmarks is a critical task for mobile robots. Landmarks are used for robot positioning, and for building maps of unknown environments. In this context, the traditional recognition techniques based on strong geometric models cannot be used. Rather, models of landmarks must be built from observations using image-based techniques. This paper addresses the issue of building image-based landmark descriptions from sequences of images and of recognizing those landmarks. Beyond its application to mobile robot navigation, this approach addresses the more general problem of identifying groups of images with common attributes in sequences of images. We show that, with the appropriate domain constraints and image descriptions, this can be done using efficient algorithms.

1 Introduction

We consider here the problem of recognizing landmarks in sequences of images taken from a moving vehicle. Even with reasonable geometric constraints, such as the fact thatthe optical axis of the camera is generally at a small upward angle from the ground plane, this is a challenging problem for a number of reasons. First of all, the appearance of any given landmark varies substantially from one observation to the next. Changes in viewpoints, illumination, and external clutter all contribute to the variability of the observed landmarks. For those reasons, it is not possible to use many of the object recognition techniques based on strong geometric models.

The alternative is to use image-based techniques in which landmarks are represented by collection of images which are supposed to capture the "typical" appearance of the objects. The information most relevant to recognition is extracted from the collection of raw images and used as the model for recognition.

Progress has been made recently in developing such approaches. For example, in object modeling [4], 2D or 3D models of objects are built for recognition applications. Extensions to generic object recognition were reported [5]. Other approaches use the images directly to extract a small set of characteristic images of the objects; these images are compared with observed views at recognition time, e..g, eigen-images techniques.

A similar problem, although in a different context, is encountered in image indexing, where the main problem is to store and organize images to facilitate their retrieval [1][7]. The emphasis in this case is on the kind of features used and the type of requests that can be made by the user.

Our approach attempts to combine these two aspects of the problem. In a training stage, the system is given a set of images in sequence. The aim of the training is to organize these images into groups based on similarity of image attributes. The basic image representation is based on distributions of different feature characteristics. A distance is defined to compare the distributions and to measure the similarity between images. This distance is then used to group the images. Each group is itself characterized by a set of attributes. When new images are given to the matching algorithm, it evaluates a distance between these images and the groups. The system determines to which group this image is the closest, and a set of thresholds is used to decide if the image belongs to this group.

Section 2 of the paper deals with representing the images with the appropriate attributes. Section 3 addresses the comparison of images using the feature distributions. Because of the potentially wide variations in viewpoint, the images must be registered before comparing their feature distributions. An algorithm for fast, approximate, image registration is described, also in Section 3. Section 4 describes the grouping of images from a training sequence into groups to form landmark models. Section 5 describes the algorithm for matching images with models. Experimental data on training and test sequences from an urban area are presented throughout the paper.

2 Representing Images

Two standard classes of attributes are used for describing the images: color and edge distributions. As was demonstrated in image retrieval work, color distribution can be a powerful attribute [11]. However, color information must be used with caution because large regions may have little color information and the effect of shadows may change the color distribution drastically. The approach taken here is to consider only those pixels that have high enough saturation and to discard all the other pixels from the image. For the remaining pixels, the normalized red value is used in order to minimize the effect of shadows. This is similar to the approach taken in [8] for shadow reduction in outdoor imagery. In the remainder of the paper, the term “color” refers to the single normalized red value computed at points of high enough saturation.

The color values are resampled by using a standard equal-size equalization. Specifically, the histogram of color values is divided into eight classes of roughly equal numbers of pixels. The color image is then coded on eight levels using those classes. This coarse quantization of color is necessary due to the potentially large color variations which make direct histogram comparison impossible.

Figure 1 shows the color images, coded on eight levels, for two typical images from a training sequence. In the normalized images, only the pixels with sufficient saturation are shown. Although the images are taken under substantially different illuminations, the images are comparable after filtering and red normalization.

Because of the potentially large differences in viewpoint and illumination, color distribution cannot be directly compared in image space. Metrics have been proposed for comparing color histograms which can tolerate substantial variation in color distribution [9]. The approach chosen here uses a transition matrix rather than a direct histogram to represent the color distribution. Specifically, a color transition matrix *Cij* is created in which *Cij* is the number of pixels with value i and with at least one neighbor with value j. This transition matrix captures the global distribution, as in a histogram, and the spatial distribution of colors. The 8x8 transition matrix is used in the computation of the image distance metric described below.

Intensity edges constitute the second class of features. A typical edge image is shown in Figure 2 after linking and expansion of edge elements into segments. Several image attributes are computed using the image segments:

•*Edge Image:* A reduced, 120x160 binarized edge image E is used for image registration and distance comparison.

•*Segments:* The histogram, H_l, of segment lengths is computed and normalized by the total number of segments N_s. The histogram is taken over 20 buckets in the current implementation. N_s is also retained as an image attribute. Similarly, a histogram H_o of the orientation of the segments is computed over 18 buckets.

•*Intersecting Segments:* Pairs of intersecting segments are identified and a histogram of their relative orientations, H_i, is constructed over 18 buckets. The histogram is normalized by the total number of pairs N_i. N_i is also retained as one of the image attributes.

•*Parallel Segments:* Pairs of parallel segments are also identified and histograms of their lengths and orientations are computed in H_{pl} and H_{po}, respectively. The histograms are normalized by the total number of parallel segments, N_p.

The total set of attributes for an image consists of five histograms, the number of segments of three different types, the edge image, and the transition matrix. Those attributes were selected to convey the main features of the scene, while at same time being resistant to changes in illumination and viewpoint.

So far, we have described attributes computed over the entire image. In reality, global comparison of images is not going to perform well because of substantial variations in the image as the viewpoint changes. For example, features that are visible in one view may disappear in another view even though the feature distribution on the object of interest may be identical in the two images. In order to handle this problem, the images are divided into sub-images; and the attributes described above are computed within each sub-image. For example, S histograms H_l^j are computed for j = 1..S, where S is the number of sub-images. This representation allows for considering only part of the image.

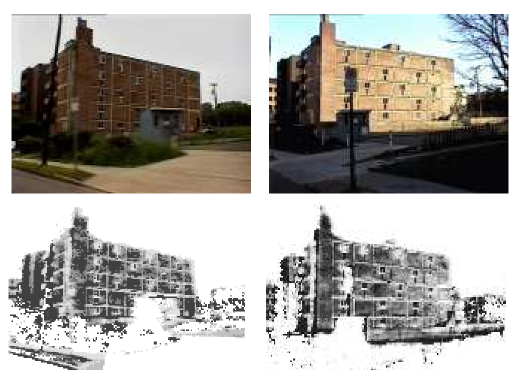

Figure 1: Color normalization; (top) Original images; (bottom) Normalized images.

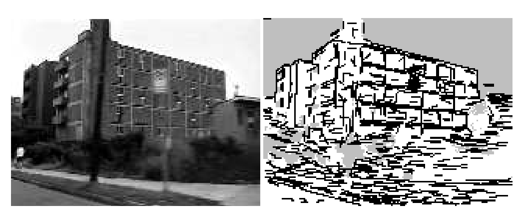

Figure 2: Segment distributions in a typical training image.

In the results presented below, sixteen sub-images were used from a regular 4x4 subdivision of the original image. This choice is a compromise between using a large number of small sub-images, which runs the risk of not having enough information in each sub-image, and using a small number of large sub-images, which may not have any advantage over using the entire image. This choice is also based on the average variation between images in the type of video sequences used in the test outdoor scenes.

3 Comparing Images

3.1 Registration

Given two images I_1 and I_2, the registration problem consists of finding a transformation $\mathbf{H}$ such that $\mathbf{H}(I_1)$ is as close as possible to I_2. This registration problem can be made tractable through a few domain assumptions. First of all, we are interested in landmarks that are far from the observer. Second, we assume that the images come from video sequences taken from a moving vehicle looking forward or transverse from the direction of motion, in a way similar to a human observer. As a result, the problem can be approximated by using an affine transformation $\mathbf{H}$ and by concentrating on the top half of the image, since the bottom typically contains more of the ground plane and little information about the landmark of interest.

Given those approximations, image registration is implemented in a two-step approach. In the first step, an initial estimate $\mathbf{H_0}$ is computed by using only the data at the top of the images. For each column x, pixels that have similar values are grouped into a connected interval of length $S(x)$. A direct least-squares solution is used to minimize the sum of squared differences $(S_1(ax+b)+c - S_2(x))^2$ taken over all the columns x. The resulting parameters a,b, and c correspond to the scale and translation along the rows of the image, and the translation along the columns, respectively. Those three parameters are by far the largest contribution to image change in typical video sequences taken from a moving vehicle.

This simple approach to the partial estimation of $\mathbf{H}$ takes advantage of the fact that the objects of interest do not generally extend all the way to the top of the image and that, consequently, the top contour is well-defined. More general algorithms based on feature matching could be used if this assumption were not valid. This approach is similar to other registration algorithms based on skyline matching (see [3] for a review.)

The initial estimate of $\mathbf{H}$ is built by placing a,b,c in the first row of the matrix. $\mathbf{H}$ is then refined by comparing the images directly. In this refinement algorithm, the SSD between a fixed set of points of I_1 and the corresponding set of points on I_2 is computed using the current estimate of $\mathbf{H}$. A change in the assignment of corresponding pixels that reduces the error is computed by moving pixels by one pixel. $\mathbf{H}$ is then estimated based on the updated correspondences. This algorithm iterates until a stable value of $\mathbf{H}$ is reached. This algorithm converges as long as the initial estimate $\mathbf{H_0}$ is close to the correct value. In particular, the algorithm performs well as long as large in-plane rotations do not contribute substantially to $\mathbf{H}$. More general registration algorithms can be used in those situations.

Figure 3 shows an example of two images of the training sequence Craig3. As expected, the registration degrades from top to bottom in the image. As will be shown in the next sections, the registration is sufficient for correctly recognizing this building from a variety of viewpoints.

In the remainder of the paper, whenever two images are compared, it is understood that this registration procedure has been applied and that the comparison is performed only on the overlapping part of the two images.

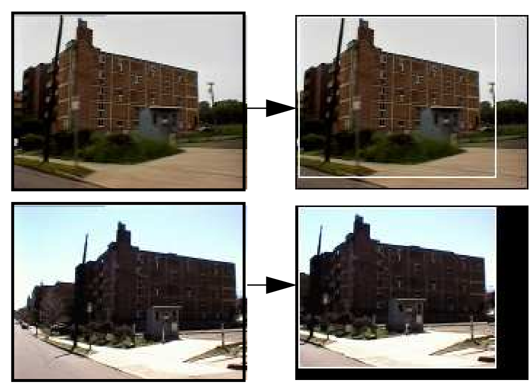

Figure 3: Approximate image registration; (top) Reference image; (bottom) registered image.

3.2 Image Distance

Given a "model" image, I_M, and an observed image, I_O, a distance can be computed by comparing the image attributes. More precisely, I_O is first registered with I_M and the attributes are computed from the new, registered, image I_O'. The global distance is defined as a sum of distances between the attributes of the sub-images of I_M and I_O' in the area in which they overlap. The distance between attributes is defined as follows. For the color transition matrix C, the distance is computed by computing the SSD of the entries of C from the two images. In computing this distance, it is natural to give more weight to regions in which there is more color variation rather than to uniform regions. This is implemented by giving more weight to the off-diagonal elements of C, which correspond to pixels with large variations of color, rather than to the elements close to the diagonal, which lie in uniform regions.

For single-values attributes, e.g., N_s, the distance is simply the squared-difference between the model and observed values. For histograms, e.g., H_l, the distance is the sum of squared difference between the elements of the histograms after a correction step. The correction step is used to compensate for mis-registration. Specifically, if the differences between the two histograms at x and $x+1$, $\Delta(x)$ and $\Delta(x+1)$ are of opposite sign and large magnitude, then, assuming $\Delta(x)$ is positive, it is decreased by a small amount while $\Delta(x+1)$ is increased by the same amount. This procedure is repeated over the entire histogram until no further adjustment is needed.

This procedure can be viewed as a coarse approximation of the earth-mover distance in which a cost is assigned in moving an entry from one histogram to another entry and the set of such motions for which cost is minimal is computed. By comparison, in the algorithm described above, a single cost of 1 is associated with a change of one entry in the histogram, and the final motion is determined using a sub-optimal algorithm. In practice, this approach effectively reduces the effect of mis-registration.

The distances between individual attributes are combined into a single distance $D(I_M^i, I_O^i)$ for each sub-image i by using a weighted sum. Finally the distances for all the sub-images are combined into a global distance, denoted by $D(I_M, I_O)$, which reflects the similarity of the images in appearance (color) and shape (edge distribution). In order to account for the particular geometry of our sequences, the weight of $D(I_M^i, I_O^i)$ decreases as i becomes closer to the bottom of the image.

4 Grouping Images Into Models

The discussion above focused on comparing individual images. Because of the large variations in appearance, multiple images must be used for representing a given landmark. That is, groups of images that represent a single object of interest must be extracted from training sequences. In this section, we briefly describe the algorithm used for extracting a small number of discriminating groups of images from a training sequence and how to use those groups as models for recognition. In the example used throughout the paper, the training sequence is called Craig3 (MOVIE 2.) An overview of the grouping algorithms is given in the section since a more formal description of grouping algorithms was included in an earlier paper [ ].

4.1 Grouping

Let us denote the training sequence by I_i, $i=1..N$. The mutual distance between images in the sequence can be computed as: $d_{ij} = D(I_i, I_j)$, where D is the image distance defined above (in particular, it is implicit in this definition that I_i and I_j are registered as part of the computation of D.) A pictorial representation of the set of values d_{ij} is shown in Figure 4 for the training sequence Craig3. In this representation, the d_{ij}'s are displayed as a two-dimensional image in which the dark pixels correspond to large values. The diagonal, i.e., the values $d_{ii,}$ is not shown since d_{ii} is always 0. Images for which the mutual distances are small are grouped into a number of clusters. Unlike general clustering problems in which the data can be arranged in clusters in an arbitrary manner, the training set has a sequential nature that can be used to make the grouping problem tractable. Specifically, images whose sequence numbers are far from each other cannot be grouped together because they were taken from locations distant from each other. Graphically, this means that we need only consider the values of d_{ij} for ij near the diagonal $i = j$. The exact meaning of "near" depends on the extent of the object of interest, the frequency at which images were digitized, and the rate of motion of the camera in the environment. In practice, a band of 16 images around the diagonal is used for a digitization rate of 5 images/second for the images presented in this paper.

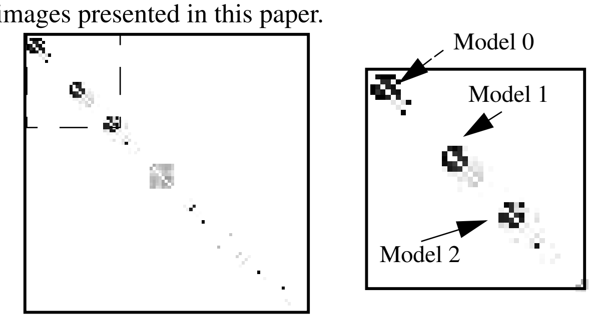

Figure 4: Distance matrix for a 145-images training sequence (Craig3); darker points correspond to lower distances; the right images shows the distance matrix for the first 50 images.

Given this sequentiality constraint, many groups can be found using standard clustering approaches. For a recognition system to be useful, however, only a small number of groups is relevant. More precisely, we are interested in the groups that have enough information, i.e., a large number of images with sufficient variation, and that are discriminating with respect to the other images. The second criterion is important because it is often the case that many groups look alike in a typical urban sequence. This is addressed by comparing the initial groups with each other and discarding those with high similarity to other groups.

The right side of Figure 4 shows a magnified version of the distance graph in the neighborhood of the three models extracted from Craig3. Example images from each of the three models are shown in Figure 5.

Figure 5: Three models extracted from training sequence Craig3; three example images are shown for each model.

4.2 Model Representation

Each of the groups extracted from the training sequence corresponds to a distinguishable landmark. Before being used for recognition, each group must be collapsed into a model suitable for comparison with test images. There are two aspects to this. First, a reference image must be created as the core image representation of the model. Second, image attributes from all the images in the group must be collapsed into a single set of attributes.

Given a group $\{I_i\}$, $i_{min} < i < i_{max}$, the first part is addressed by selecting a reference image I_o in the group -- usually the median image in the sequence. All the other images are registered to I_o using the approximate registration procedure described above, yielding new images $I_i^{'}$. The second part is addressed by computing the attributes of each $I_i^{'}$. For each attribute, the average value over all the images in the group is computed. In order to capture the variation of the attributes within the group, the variation of each attribute within the group is computed, also over all the images in the group.

In summary, given a group $\{I_i\}$, the corresponding model M consists of a reference image I_M, the average value of each attribute (average single-value attributes and histograms), and the variation of each attribute (variance of single-value attributes, and covariance matrix of vector-valued attributes.) The average value of an attribute P will be denoted by $\overline{P}$ and its variance by C_P.

5 Comparing Images to Models

Run-time recognition involves comparing a new image I with all the models in order to determine the best match. The first step in comparing I to a model M is to register I with I_M and to compute the attributes of the registered image I' over the overlapping region between I' and I_M. The attributes P^I_i from I' are compared with the attributes P_i of M by using a sum of distance weighted by the variation of the parameters in the model: $D(I, M) = \Sigma_i \alpha_i (P^I_i - \overline{P}_i)^t C_i^{-1} (P^I_i - \overline{P}_i)$. In this definition, the coefficient α_i are fixed and represent the relative importance of the different types of attributes. Those weights are computed using a principal component analysis technique on the set of attributes from the training sequence, as described in an earlier paper. The sum in $D(I, M)$ is evaluated over all the attributes and all the sub-images in common between I' and I_M. It implements a maximum likelihood estimate of the distance between image and models.

The definition of distance given above does not take into account the fact that there might be little overlap between I' and I_M. In fact, $D(I, M)$ could be small and could force a match if the registration is poor, i.e., the overlap is small. This situation is addressed in two ways. First, I is not matched with M if the registration area $R(I,M)$ is below a threshold T_R. This threshold is computed automatically from the training set by computing the mean registration area between images of the same model and adding $3\sigma_R$, where σ_R is the variation of registration area over all the images of the model. Second, the actual distance used for matching is modified to: $D'(I, M) = D(I, M)/R(I,M)$. This weighting penalizes images with low overlap with the model.

The model that realizes the minimum of $D'(I, M)$ over all the models is taken as the best match to the image. Simply using the minimum would lead to a high rate of false positives in cases in which $D'(I, M)$ is low for all the models, i.e., the image matches poorly all the models, and in cases in which the distances to two models are of similar magnitude, i.e., the image is ambiguous. Both cases are addressed in standard ways. The first case is addressed by rejecting the image if $D'(I, M)$ is below a threshold T_D. The second case is addressed by rejecting the image if the margin between D' for the best model and for the next best model is lower than a threshold T_m. Both T_D and T_m are computed automatically from the training sequences. T_m is computed from the average difference between distances in the training data set. As in any recognition problem, there is a compromise between recognition rate and rate of false positives. Because of the application, the thresholds are set conservatively to minimize the rate of false positives, i.e., the algorithm favors rejecting images that are marginal matches over running the risk of mis-classifying them.

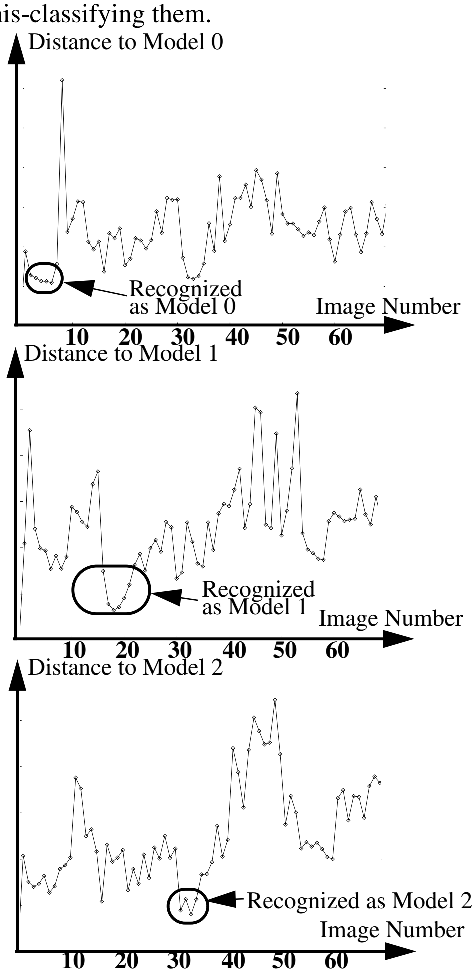

Figure 6: Graph of the distance to three models for a test sequence (Craig4).

Figure 6 shows the distances between the images of a test sequence called Craig4 (MOVIE2) and three models from training sequence Craig3. The same scale is used in all three graphs. The recognized models are indicated in the graphs. For reasons of space, results on all the images of Craig4 cannot be included here but are shown separately (MOVIE1.) The graphs show that the images are recognized by a substantial margin.

6 Performance

The image recognition algorithm was evaluated over a dozen data sets, using different models. Results are reported here using the same training sequence Craig3 that was used throughout the paper. All the data sets were annotated manually in order to provide ground truth. The composition of twelve data sets is shown in Table 1. The data sets are divided into three groups. The first group consists of five video sequences of urban areas similar to the one used in Craig3 but which do *not* contain any of the three models. Those data sets are useful for verifying that the algorithm reports a small number of false positives. The second group consists of video sequences taken in the same area as Craig3, thus containing images of the model objects and of other areas. This group of sequences is typical of the data that would be used in a real situation. Finally, the third group of data sets contains only images of the models.

Name	Model 0	Model 1	Model 2	None	Total
Fifth	0	0	0	54	54
Bayard	0	0	0	90	90
Center	0	0	0	35	35
Mwood	0	0	0	35	35
Negley	0	0	0	32	32
Craig4	4	4	3	49	60
Craig1s	3	2	1	36	42
Craig2s	3	2	2	52	59
Craig3s	2	2	0	48	52
Craig2c	5	5	5	0	15
Craig1c	7	4	6	0	17
Craig3c	5	6	5	0	16

Table 1: Description of seven test sequences.

	Recognized	Rejected	Mis-Classified
Model Images	77.4%	22.6%	0%
Non-Model Images		99.7%	0.3%

Table 2: Summary performance statistics.

The performance of the algorithm is summarized in Table 2. The two rows of the Table show statistics separately for the images that belong to the models, and those that do not, according to the manual annotation of the sequences. The "Recognized" column is the percentage of images that are matched to the correct model; the "Rejected" column is the percentage of images that are not matched with any model; finally, the "Mis-Classified" column is the percentage of the total number of images that are either matched to the wrong model (first row), or that are matched to a model when they should not be (second row) -- what would normally be called "false positives".

To interpret those results, it is important to keep in mind that false positives must be avoided as much as possible for the class of applications under considerations. As mentioned above, this means that the parameters generated from the training sequence are such that the rate of false positive is minimized, possibly to the detriment of the recognition rate. Indeed, the experiments show that the rate of false positives is close to 0% in all cases and that, in particular, no image from the models is mis-classified.

Figure 7 and show examples of images recognized as Model 0 under substantial variations. Figure 8 and Figure 9 show examples of images from Model 0 rejected because of viewpoint and illumination variations, respectively. Again, it is important to note that the images are rejected, but they are not matched to the wrong model. Additional examples of recognized and non-recognized images are provided separately (MOVIE3).

Figure 7: Example images from three different test sequences recognized as Model 0 under different illumination and viewpoints.

Figure 8: Two examples of images not recognized because of large variations in aspect.

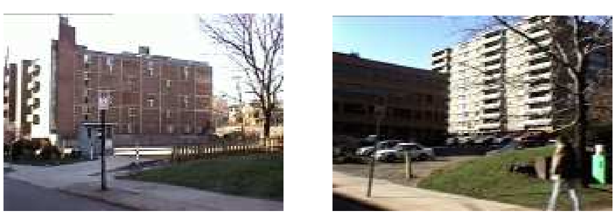

Figure 9: Two examples of images not recognized because of extreme illumination conditions.

7 Conclusion

Results on image sequences in real environment show that visual learning techniques can be used for building image-based models suitable for recognition of landmarks in complex scenes. The approach performs well, even in the presence of significant photometric and geometric variations, provided that the appropriate domain constraints are used.

Several limitations of the approach need to be addressed. First of all, rejection of unreliable groups needs to be improved. In particular, the selection of the parameters controlling the grouping needs to be implemented in a principled manner. Second, images that do not contribute information should be filtered out of the training sequences. For example, images with high density of vegetation lead to great variations in many of the feature distributions and thus make grouping and recognition difficult. This is currently addressed by the threshold on registration area.

References

[1] Bach et al. The Virage Image Search Engine:An Open Framework for Image Management. *SPIE Proc. Image Storage and Retrieval*. 1996.

[2] S. Carlsson. Combinatorial Geometry for Shape Indexing. *Proc. Workshop on Object Representation for Comp.Vision*. Cambridge. 1996.

[3] F. Cozman. Position Estimation from Outdoor visual landmarks. *Proc. WACV'96*. 1996.

[4] P. Gros, O. Bournez and E. Boyer. *Using Local Planar Geometric Invariants to Match and Model Images of Line Segments*. To appear in Int. J. of Comp. Vision and Image Underst.

[5] R. Horaud, T. Skordas and F. Veillon. Finding Geometric and Relational Structures in An Image *Proc. of the 1st ECCV*. Antibes, France pages 374--384, April 1990

[6] B.Lamiroy and P.Gros. Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing. *Proc. of the 5th ECCV*, Cambridge, England, pages 59--70, vol. 1, April 1996.

[7] R.W. Picard. A Society of Models for Video and Image Libraries. *IBM Systems Journal*, 35(3-4):292-312. 1996.

[8] D.A. Pomerleau. Neural network-based vision processing for autonomous robot guidance. *Proc. Appl. of Neural Networks II*. 1991.

[9] Y. Rubner, L. Guibas, C. Tomasi. The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval. *Proc. IU Workshop*. 1997.

[10] C. Schmid and R. Mohr. Combining Greyvalue Invariants with Local Constraints for Object Recognition. *Proc. CVPR*. San Francisco, California, USA. pages 872--877, June 1996.

[11] M.J. Swain, D.H. Ballard. Color Indexing. Int. J. of Comp.Vision, 7(1):11-32.1991.