MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

TRANSFORM CLUSTERING FOR MODEL–IMAGE FEATURE CORRESPONDENCE

Raj Talluri and J. K. Aggarwal

Computer and Vision Research Center
The University of Texas at Austin
Austin, Texas 78712, USA

ABSTRACT

In this paper we present a novel technique for establishing a robust and accurate correspondence between a 3d model and a 2d image. We present a transform clustering approach to isolate the transformation that maps the model features to the image features. It is shown that this transform clustering technique alleviates the problems with using the traditional Hough transform techniques used by previous researchers. We demonstrate the effectiveness of our approach in estimating the position and and pose of an autonomous mobile robot navigating in an outdoor urban environment. We present experimental results of testing this approach using a model of an airport scene.

INTRODUCTION

The task of establishing a reliable and accurate correspondence between an image of a scene and a stored model of it occurs in a large number of computer vision problems. Autonomous navigation of a mobile robot given *a priori* model of the environment and model-based object recognition are two examples of computer vision tasks in which the model-image correspondence needs to be addressed. In the context of autonomous navigation, the robot is provided with a preloaded world model of the environment. The world model could be in different forms, such as a Digital Elevation Map (DEM), a CAD description, or a floor map. The robot uses an onboard camera to image the environment. Once we establish a correspondence between the image and the model, the robot's position and pose can be determined. This position information can be used by the robot to successfully navigate in its environment. In the context of model-based object recognition, we are given a geometric description of the object to be recognized and an image of the scene in which the object is present. The task is to isolate the object in the scene by using the image. Model-image correspondence are particularly difficult because the image and the model are usually in different formats, different co-ordinate frames and of different dimensions.

A popular approach to solving this problem is to extract features from the image and search the model description for the corresponding set of features. The type of features required and the number of features used depends on the model description and what is assumed to be known about the scene. For example, in navigating the robot in an indoor structured environment with a given CAD model of the environment, it is common practice to use line segments as features [3]. On the other hand, in navigating the robot in an outdoor mountainous terrain given a DEM of the environment, using curves may be a logical choice [9].

Typically, in these problems the model and the camera (robot) are specified in two different co-ordinate systems. Once we extract the relevant features from the image and identify the corresponding features in the model, we can computer the transformation $\mathcal{T}$ that maps the model features into the image features. The parameters of this transformation are the required position and pose of the camera (robot) with respect to the model. Solving for the parameters of $\mathcal{T}$, once a set of model-image feature correspondences is established, is a very well studied problem [2]. Therefore, the crucial task to be accomplished is that of establishing a reliable and accurate correspondence. Noise, occlusions, errors in feature detection and inaccurate model descriptions further complicate this correspondence problem.

Transform Clustering: Previous researchers have considered the technique of matching a key model feature, such as a long edge or a set of lines in specific orientations, to establish an initial transformation [1, 6]. Subsequent assignments are then used to refine this transformation. New assignments are selected on predictions of a model feature, projected into the image using the current transformation. However, these techniques assume that it is possible to initially select a correct key model feature, which may not always be possible.

Some researchers used the generalized Hough transform and its related parameter hashing techniques to perform *transform clustering* to isolate the transformation mapping the model features into the image features [6, 5, 10]. The generalized Hough transform works by first quantizing the n-dimensional parameter space into discrete buckets or bins. The parameters are the components describing $\mathcal{T}$. From the given image, features are extracted using a feature extractor. Then all the possible model-image feature correspondences are hypothesized and, for each hypothesis, the parameter vector is computed. For each parameter vector so computed, its n components are quantized and used as indices to vote in one of the n-dimensional buckets. Searching for large clusters is then accomplished by finding the buckets with a large numbers of entries. Sometimes it is possible that one correspondence may not give explicitly all the components of the parameter vector, but may only give a range of possible values for each component. In this case, entries are

This research was supported by in part by Army Research Office contract DAAL03-91-G-0050 and in part by Air Force Office of Scientific Research (AFSC) contract F49620-89-C-0044.

made into all the buckets within range. The advantage of this approach is that clustering provides a robust criterion for selecting valid model feature assignments. The effects of missing or incorrect features due to occlusion, shadows, or low contrast, are not felt.

The problems associated with using the Hough transform approach to transform clustering are that large transform clusters may occur randomly. If these clusters are as large or larger than those due to the correct transform, the estimation procedure that relies only on the Hough transform will be erroneous. If the number of buckets is increased, then the possibility of random large clusters is alleviated but the number of computations grows rapidly. Grimson [4] summarizes these problems with the generalized Hough transform.

This paper presents a method to reduce the problems associated with the Hough transform approach to transform clustering by using a partition of the parameter space, which is not necessarily uniform. The partition is, in fact, *intelligent* and uses *a priori* model information. Due to the geometric constraints imposed by the model and the camera geometry, not all model features may be *visible* in all camera positions. Typically occlusions between the model features affect their visibility at various positions. However, since we know the 3d descriptions of the model features, these geometric constraints can be pre-computed and used to partition the parameter space to reduce the probability of the occurrence of random transform clusters.

We demonstrate the effectiveness of our approach in estimating the position and and pose of an autonomous mobile robot. The robot is assumed to be navigating in an outdoor, urban environment. The 3d description of the lines that constitute the rooftops of the buildings is given as a world model. The position and pose of the robot are estimated by establishing a correspondence between the lines extracted from the image (image features) and the lines that constitute rooftops of the buildings (model features). By exploiting the visibility constraints imposed by the 3d world model and the camera geometry, we partition the parameter space into into distinct, non-overlapping regions called *Edge Visibility Regions* (EVRs) [7]. In each of these regions, we also store the list of model features that are visible from within that region. We then hypothesize a correspondence between all pairs of model and image features and compute the range of possible transformations for each hypothesis. We vote in all the regions in the parameter space where this transformation is valid. After considering all the pairings, we select the regions in the parameter space with the large numbers of votes as the candidate EVRs for position estimation. The actual correspondence and position estimation are then performed by a constrained search process within these EVRs using a interpretation tree search paradigm.

PARTITIONING THE PARAMETER SPACE

Consider the world coordinate system $OXYZ$ and the robot coordinate system $O'X'Y'Z'$ shown in Figure 1. Generally, the transformation $\mathcal{T}$ that transforms $OXYZ$ into $O'X'Y'Z'$ has six degrees of freedom: three rotational and three translational. Sometimes, depending on the application, some of these degrees of freedom can be elimi-

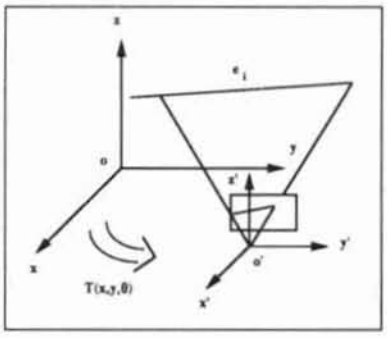

Figure 1: The world and robot co-ordinate systems

nated. Most mobile robot self-location tasks make the assumption that the robot is on the ground (OXY plane), so the Z-translation (the height of the robot above the ground) is assumed to be known or to be zero. The camera on the robot is assumed to have zero roll (rotation about X-axis), and the tilt angle of the camera, (rotation about the Y-axis) is assumed to be measurable. So, there are effectively three parameters in the transformation: two translational (X, Y) and one rotational θ (the pan angle of the camera, which is a rotation about the Z-axis). Likewise, in this paper we have only three parameters of $\mathcal{T}$: X, Y and θ. The parameter space of the transformation is thus the entire OXY plane and the range of robot orientation θ is 0 through 360 degrees.

In this section, we briefly we describe a method for partitioning the OXY plane into regions called Edge Visibility Regions (EVRs) using the given world model description. For more details see [7]. Associated with each EVR is a list of the world model features *visible* in that region, called the *visibility list* (VL). No two adjacent EVRs have the same VL. Also stored for each entry in the VL of an EVR is the range of robot orientations from which the feature is visible. Thus, each EVR is a region of space which has the topological property that from its points, the same set of edges of the model are visible through a complete circular scan. The EVR representation partitions the entire parameter space of (X,Y,θ) and captures the visibility constraints between the world model features.

The algorithm that divides the OXY plane into the desired EVRs, along with their associated VLs, uses three subprocesses called *Split*, *Project*, and *Merge*. The algorithm's basic idea is to start with the entire OXY plane as one EVR with a NULL visibility list. Each of the polygons that makes up the building's rooftop in the world model is considered in turn by extending its edges, and the EVRs that are intersected are divided into two new ones. The new EVRs then replace the old one, and the VLs of the new EVRs are updated to account for the visibility of this edge by considering it to be visible in one half-plane, say the half-plane into the left of the edge, and invisible in the other. The *Split* process handles this updating. For each new rooftop considered, the mutual occlusion of the rooftop's edges with the other existing rooftops is handled by forming the *shadow region* of these edges on the other existing rooftops. The *Project* process handles the forming of these shadow regions. Finally, the *Merge* process concatenates all the adjacent EVRs with identical VLs into one EVR. After partitioning the OXY plane into EVRs,

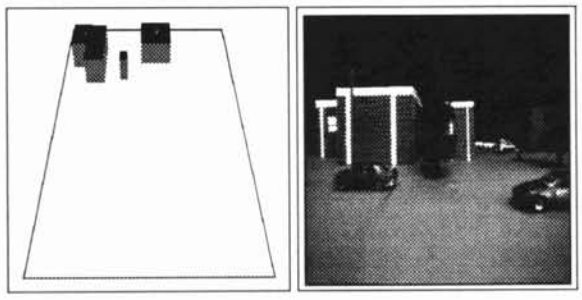

Figure 2: (a) World model (b) Robot view

the range of the robot's orientations for which each model feature in the VL of an EVR is visible, is also computed and stored. An efficient method to compute these ranges is also developed. Figure 2(a) shows the world model and Figure 5(b) shows the EVR description computed from this world model.

FEATURE EXTRACTION

In this research, we used a scale model of the Austin Executive Park Airport to test the position estimation algorithms developed. The world model thus consists of the 3d descriptions of the rooftops of the three buildings in this airport. Figure 2(a) shows this world model. A calibrated camera is placed in this environment and used to acquire the images of the model. These are then used as the robot's views. Figure 2(b) shows one such view. We use a Canny edge detector to extract the edges from this image. Contiguous edges are then linked using a pixel chaining algorithm. We then use a line fitting technique to form line segments from these pixel chains. These line segments are then thresholded by length to remove all the lines shorter than 20 pixels. Figure 3(a) shows these lines. We use a *rooftop extraction* technique to select the lines that correspond to the rooftops only. The technique scans each column of the line segment image from top to bottom and selects the *topmost* lines only in each column. All the lines that lie below, completely within the projection of a selected line, are then discarded. The lines isolated using this technique are then considered as the image features. Figure 3(b) shows these lines. Notice that the image feature extraction procedure is far from perfect. Some of the lines that correspond to rooftops are not extracted and, due to noise and occlusion, some of the extracted lines do not arise from the rooftops but from extraneous objects such as trees and telephone poles. The task is thus to use the transform clustering and the search technique to correctly isolate the model features and the noise features from these image features and accurately estimate the robot's position and pose in the environment.

MODIFIED HOUGH TRANSFORM

Having formed the EVR description of the environment and extracted the features from the images, we use a modified Hough transform to isolate a small set of EVRs likely to contain the robot's location. The EVRs are used as a partitioning of the parameter space (X, Y, θ) of the transformation. We find that this partitioning alleviates the problems of traditional Hough transform, namely, the random occurrence of large clusters and the resulting need for the large amounts of memory required to perform the

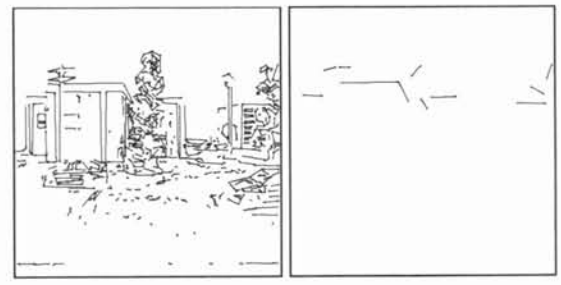

Figure 3: (a) Detected lines (b) Image features

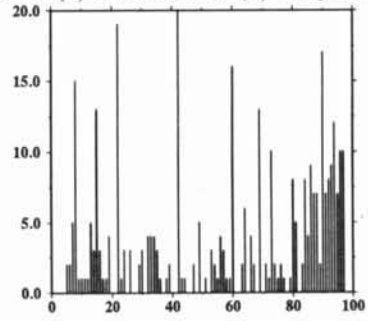

Figure 4: EVR no. vs the number of votes

fine partitioning of the parameter space to eliminate this problem. Since it is difficult to accurately extract the end points of the rooftops, we use infinite lines and not line segments as the image features. The image features are 2d lines and the model features are 3d lines. Using one 2d to 3d line correspondence, we can compute the orientation of the robot θ and get a constraint on the position of the robot of the form aX + bY + c = 0, where a,b, and c are constraints. This constraint describes a line **L** in the OXY plane. See [8] for details of the derivation.

We hypothesize all the possible model-image feature correspondences, and for each hypothesis compute the θ and get the constraint line **L** on (X, Y). We now vote in all the EVRs where: 1) the line **L** intersects the EVR; and 2) the θ lies within the range of possible robot orientations in the visibility list of the EVR. We finally select the EVR with a largest numbers of votes as the candidate EVRs most likely to contain the robot's location. Figure 4 shows a plot of the EVR number vs. the number of votes. Figure 5(a) shows the complete EVR description and Figure 5(b) shows the selected candidate with a large number of votes.

INTERPRETATION TREE SEARCH

Having isolated the candidate set of EVRs most likely to

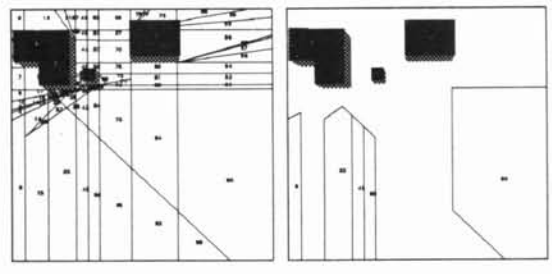

Figure 5: (a) EVR description (b) EVRs isolated by the Hough transform

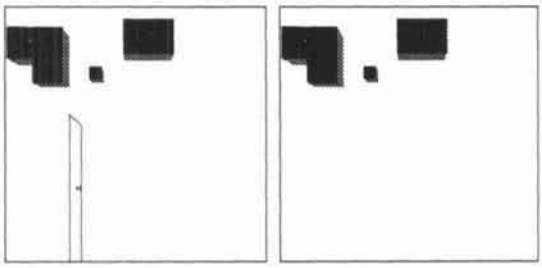

Figure 6: (a) Final EVR (b) Estimated robot location

Actual Position	Actual Pose deg	Estimated Position	Estimated Pose deg	EVR No.
(875,410)	0	874.93,411.29	1.03	42
(750,1010)	-30	752.45,1016.21	-32.02	90
(900,50)	5	901.74,49.84	4.87	8
(800,800)	-5	798.63,800.12	-4.58	84
(1320,1350)	-15	1324.27,1347.46	-16.14	89
(350,1400)	-90	352.23,1404.67	-92.27	94
(460,1400)	-75	460.15,1406.14	-78.04	91
(425,1450)	-85	428.57,1454.21	-88.13	93
(1350,900)	-15	1351.12,896.14	-14.21	83
(1475,600)	0	1480.11,605.26	1.02	69

Table 1: The Search results

contain the robot's location using the modified Hough transform, we now wish to isolate the robot's location more precisely among these EVRs. For each of the candidate EVRs we form an *interpretation tree* of all the possible model-image feature correspondences and then search this trees to isolate the correct set of correspondences. Note that these tree are very short since we only need to consider those model features that are present in each EVR's VL. Also by using the geometric constraints established by the EVR, that is, its extent in the OXY plane and the range of possible θ values, we can prune large parts of this interpretation tree.

This search process finally isolates the correct EVR containing the robot's location and a set of model-image feature correspondences. Using all of these correspondences in a least squares framework, the robot's position and pose are accurately estimated. Figure 6(a) shows the EVR isolated as containing the robot's location and Figure 6(b) shows the final estimated robot's position. We find that the estimated position and pose obtained by these techniques are quite close to their true values. Table 1 compares the estimated and the actual values obtained from the test runs using the world model shown in Figure 2(a).

CONCLUSIONS

This paper presented a novel and efficient transform clustering technique for establishing a robust and accurate correspondence between a 3d model and a 2d image. We demonstrate the effectiveness of this technique in estimating the position and pose of an autonomous mobile robot in an outdoor urban environment consisting of polyhedral buildings. It is shown that this transform clustering technique alleviates the problems associated with the traditional Hough transform techniques used by previous researchers.

Although we have demonstrated the utility of the technique for the mobile robot self-location problem, the approach can be easily extended to other computer vision tasks such as model-based object recognition. One possible approach is to precompute the *characteristic views* or *aspects* of the object to be recognized and use these to partition the parameter space. By imposing suitable and practical restrictions on the number of degrees of freedom in the transformation between the model and the image [6], the number of aspects can be kept tractable. By selecting an appropriate set of features from the image and using a similar transform clustering approach as described in this paper, it is possible to isolate a small set of aspects of the object corresponding to the given image. Using a tree search technique it is then possible to establish a more accurate correspondence between the image features and the model features and isolate the correct aspect, and thereby recognize the object from the given set of models.

References

[1] N. Ayache and O. Faugeras. HYPER: a new aproach for the recognition and positioning of 2d objects. *IEEE Trans. on PAMI*, 8(1):44–54, 1986.

[2] R. M. Haralick et. al. Pose estimation from corresponding point data. *IEEE Tran. on Sys., man and cybernetics*, 19(6):1426–1445, November/December 1989.

[3] C. Fennema and A. R. Hanson. Experiments in autonomous navigation. In *Proc. of the tenth Int. Conf. on Pattern Recognition*, pages 24–31, Atlantic City, 1990.

[4] W. E. L. Grimson and D. P. Huttenlocker. On the sensitivity of the hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):255–274, Mar 1990.

[5] W.E.L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on PAMI*, 9(4):469–482, 1987.

[6] Teresa M. Silberberg, David A. Harwood, and Larry S. Davis. Object recognition using oriented model points. *Computer Vision, Graphics, and Image Processing*, 35:47–71, 1986.

[7] R. Talluri and J. K. Aggarwal. Edge visibility regions - a new representation of the environment of a mobile robot. In *IAPR Workshop on Machine Vision Applications, MVA '90*, pages 375–380, Tokyo, Japan, November 1990.

[8] R. Talluri and J. K. Aggarwal. Mobile robot self-location using constrained search. In *Proc. IEEE Workshop on Intellligent Robots and Systems, IROS '91*, Japan, November 1991.

[9] R. Talluri and J. K. Aggarwal. Position estimation for a mobile robot in an outdoor environment. To appear in the *IEEE Transactions on Robotics and Automation*, 1992.

[10] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proc. of IEEE Intl. Conf on Robotics and Automation*, pages 208–220, Raleigh, March 1987.