Navigation Based on a Network of 2D Images

D. Wilkes
Dept. of Comp. Sci.
U. of Toronto
Toronto, Canada

S. Dickinson
Dept. of Comp. Sci.
U. of Toronto
Toronto, Canada

E. Rivlin
Dept. of Comp. Sci.
Technion
Haifa, Israel

R. Basri
Dept. of Appl. Math.
Weizmann Inst.
Rehovot, Israel

Abstract

This paper describes the integration of 2D stimulus-driven robot localization and positioning with a token-based correspondence method in a practical robot navigation system. The approach is interesting because it allows for modular acquisition and update of world knowledge for navigation, and robustness of navigation to low-level errors. No special marking of the world is necessary, so the robot may operate in quite general environments. Tests in a real industrial environment confirm the potential of the method.

1 Introduction

Many approaches have been taken to solving the problem of robot positioning in a known environment. By *positioning* we mean the act of computing the coordinates of the robot in the environment. The approaches may be divided into two broad classes: those that modify the environment in order to provide position cues that are easy to sense, and those that do not. The former class, including wire and tape guidance, corner-cube and bar-code-based approaches, is surveyed by Tsumura [22].

There is a great variety of techniques that do not modify the environment. These are typically applied by using sensors continuously or intermittently to correct dead-reckoning position estimates. One such method operates by following pre-planned, fixed paths parallel to walls, and corrects dead-reckoning errors with sonar range measurements from the walls [7]. Use of sonar in more general situations is complicated by specular reflection of the sonar signal from many surfaces, although a variety of techniques have met with some success provided that the environmental structure is relatively simple and that distances to the structure are relatively small [1, 4, 8, 12, 14, 18]. Use of other active ranging technologies shows promise, but is still somewhat limited by the cost of the sensors and eye-safety concerns for laser devices.

There are many computer vision-based methods for robot positioning. Some are based on estimation of the three-dimensional position and orientation of fixed environmental structures with respect to the robot, using multiple frames from different positions [11], or from a single image [3, 10, 19]. Other techniques, such as [9, 21] take a homing approach where a test image (or signature) is compared to a model image. The discrepancy between the views is used to guide the robot to the position where the model view was acquired. The main drawback of these approaches is that the robot must be near to the position where the model view was acquired for the homing to be effective. Furthermore, the robot is constrained to move to a deterministic set of positions as it navigates its environment.

We are exploring navigation strategies based on the use of 2D images of the environment. However, unlike the above image-based methods described which home the robot through a sequence of known positions, our approach uses correspondence between test and model images to determine the robot's position from where the test image was acquired. Earlier work by two of the authors [2] has demonstrated the possibility of using sets of 2D model views for localization and positioning based on the appearance of a set of model points, in relation to the appearance of the set from known model positions. The approach is attractive for a number of reasons. First, the approach admits the possibility of automatic construction of the representation of the environment, since the representation may consist only of a set of 2D image tokens and corresponding robot positions. Second, the approach adapts easily to local changes in the environment, since only memory concerning the changed regions needs to be modified. Detection of environmental change is possible based on failure of the scheme at a particular model view. Finally, the method may be applied with redundant information, allowing robustness to failures in the low-level sensing.

There are two difficulties to be addressed in order to apply the earlier work in a real navigation system. The first is the determination of point correspondences among feature points in stored model frames and feature points in frames from unknown positions. Techniques developed by another of the authors are applicable to this problem. The second problem is to integrate this method with additional techniques for global navigation, so that an extended environment, covering many locations, may be represented. The second problem is solved by maintaining a graph-like representation of the neighbour relationships among locales defined by model views, and will be treated elsewhere. In this paper, we demonstrate the feasibility of combining the model view and correspondence work to solve real robot navigation problems, using experimental equipment and environment provided by the ARK project [15].

2 Overview of the approach

Figure 1 illustrates our approach to navigation. The robot's environment is decomposed into regions within which a set of model views of a particular piece of the environment may be used to determine the position of the robot. Figure 2 illustrates the components necessary to the approach. Low-level vision extracts tokens from the image, that are put into correspondence with model tokens. The correspondences are checked for feasibility using Basri and Rivlin's solution to the localization problem [2]. The correctly corresponded tokens then allow solution for the position of the robot relative to the known positions from which the model views were acquired. High-level navigation determines steering commands to move from one model-view region to the next.

Correspondence among tokens from various views is achieved by the method discussed in detail in [20]. Each token T is described by its set of neighbours in T-based coordinates. Each neighbour is described by a probability distribution in a normalized, uniform feature space. The method provides robustness to variation among the views in the relative appearance of each token. A brief description of the correspondence method is presented in section 4.

The scheme for localization is as follows. Given an image, we construct two view vectors from the feature points in the image; one contains the x-coordinates of the points and the other contains the y-coordinates of the points. A section of the environment is modeled by a set of such views, where the points

1051-4651/94 $04.00 © 1994 IEEE

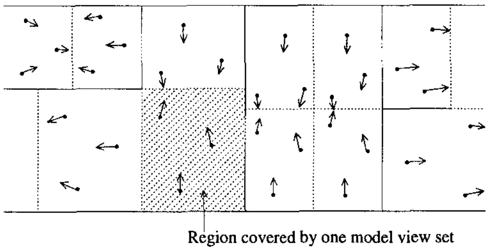

Figure 1: Overview of the approach to navigation. The environment is decomposed into regions within which particular pieces of the environment may be used to determine the robot's position. The dots are model view positions, with the arrows indicating the directions of stored model views.

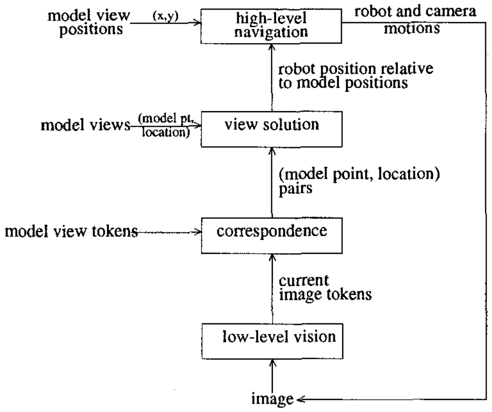

Figure 2: Components necessary to the approach.

in these views are ordered in correspondence. The appearance of a novel view of the object is predicted by applying linear combinations to the stored views. The predicted appearance is then compared with the actual image, and the correspondences between tokens in the novel and model views are accepted if the two match. Section 3 below outlines the methods both for determining the best linear combination of the model views to express the novel view (localization), and the computation for the robot's position (positioning).

3 Model Views

3.1 Localization

The scheme for localization is the following. Given P, a 2D image of a scene, and $\mathcal{M}$, a set of stored models, the objective is to find a model $M^i \in M$ such that $P = \sum_{j=1}^{k} \alpha_j M_j^i$ for some constants $\alpha_j \in \mathcal{R}$. This scheme accurately predicts the appearance of rigid objects under weak-perspective projection (orthographic projection and scale) [23]. The limitations of this projection model are discussed in [2] in more detail.

More concretely, let $p_i = (x_i, y_i, z_i)$, $1 \le i \le n$, be a set of n object points. Under weak-perspective projection, the position $p_i' = (x_i', y_i')$ of these points in the image are given by

$$\begin{aligned} x_i' &= sr_{11}x_i + sr_{12}y_i + sr_{13}z_i + t_x \\ y_i' &= sr_{21}x_i + sr_{22}y_i + sr_{23}z_i + t_y \end{aligned} \tag{1}$$

where r_{ij} are the components of a 3×3 rotation matrix, and s is a scale factor. Rewriting this in vector equation form, we obtain

$$\begin{aligned} \mathbf{x}' &= sr_{11}\mathbf{x} + sr_{12}\mathbf{y} + sr_{13}\mathbf{z} + t_x\mathbf{1} \\ \mathbf{y}' &= sr_{21}\mathbf{x} + sr_{22}\mathbf{y} + sr_{23}\mathbf{z} + t_y\mathbf{1} \end{aligned} \tag{2}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{x}', \mathbf{y}' \in R^n$ are the vectors of x_i, y_i, z_i, x_i' and y_i' coordinates respectively, and $\mathbf{1} = (1, 1, \ldots, 1)$. Consequently,

$$\mathbf{x}', \mathbf{y}' \in span\{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{1}\} \tag{3}$$

or, in other words, $\mathbf{x}'$ and $\mathbf{y}'$ belong to a four-dimensional linear subspace of $\mathcal{R}^n$. A four-dimensional space is spanned by any four linearly independent vectors of the space. Two views of the scene supply four such vectors [23, 16]. Denote by $\mathbf{x}_1$, $\mathbf{y}_1$ and $\mathbf{x}_2$, $\mathbf{y}_2$ the location vectors of the n points in the two images; then there exist coefficients a_1, a_2, a_3, a_4 and b_1, b_2, b_3, b_4 such that

$$\begin{aligned} \mathbf{x}' &= a_1\mathbf{x}_1 + a_2\mathbf{y}_1 + a_3\mathbf{x}_2 + a_4\mathbf{1} \\ \mathbf{y}' &= b_1\mathbf{x}_1 + b_2\mathbf{y}_1 + b_3\mathbf{x}_2 + b_4\mathbf{1} \end{aligned} \tag{4}$$

(Note that the vector $\mathbf{y}_2$ already depends on the other four vectors.) Since R is a rotation matrix, the coefficients satisfy the following two quadratic constraints:

$$\begin{gathered} a_1^2 + a_2^2 + a_3^2 - b_1^2 - b_2^2 - b_3^2 = \\ 2(b_1b_3 - a_1a_3)r_{11} + 2(b_2b_3 - a_2a_3)r_{12} \end{gathered} \tag{5}$$

and

$$\begin{gathered} a_1b_1 + a_2b_2 + a_3b_3 + \\ (a_1b_3 + a_3b_1)r_{11} + (a_2b_3 + a_3b_2)r_{12} = 0 \end{gathered} \tag{6}$$

To derive these constraints, the transformation between the two model views should be recovered. This can be done under weak-perspective using a third image. Alternatively, the constraints can be ignored, in which case the system would confuse rigid transformations with affine ones. This usually does not prevent successful localization since scenes are generally fairly different from one another.

To summarize, we model the environment by a set of images with correspondence between the images. For example, a section of the environment can be modeled by two of its corresponding views. The corresponding quadratic constraints may also be stored. Localization is achieved by recovering the linear combination that aligns the model to the observed image. The coefficients are determined using four model points and their corresponding image points by solving a linear set of equations. Three points are sufficient to determine the coefficients if the quadratic constraints are also considered. Additional points may be used to reduce the effect of noise.

The scheme uses viewer-centered models, i.e., representations that are composed of images. It has a number of advantages over methods that build full three-dimensional models to represent the scene. First, by using viewer-centered models that cover relatively small transformations, we avoid the need to handle occlusions in the scene. If from some viewpoints the scene appears different because of occlusions, we utilize a new model for these viewpoints. Second, viewer-centered models are easier to build and to maintain than object-centered ones. The models contain only images and correspondences. By limiting the transformation between the model images, one can find the correspondence using motion methods. If large portions of the environment are changed between visits, a new model can be constructed by simply replacing old images with new ones.

3.2 Positioning

Positioning is the problem of recovering the exact position of the robot. This position can be specified in a fixed coordinate system associated with the environment (i.e., room coordinates), or it can be associated with some model, in which case location is expressed with respect to the position from which the model views were acquired. In this section we derive the position of a robot from the alignment coefficients.

We assume a model composed of two images, P_1 and P_2; their relative position is given. Given a novel image P', we first align the model with the image (i.e., localization). By considering the coefficients of the linear combination the robot's position relative to the model images is recovered. To recover the absolute position of the robot in the room the absolute positions of the model views should also be provided. Note that the computation is done in "image coordinates" (that is, assuming a unit focal length). Positions should be normalized if world coordinates are used.

Assume P_2 is obtained from P_1 by a rotation R, translation $\mathbf{t} = (t_x, t_y, t_z)$, and scaling s. (Denote the average distance of the camera in P_1 to the scene by Z_0, s is given by $Z_0/(Z_0+t_z)$.) The coordinates of a point in P', (x', y'), can be written as linear combinations of the corresponding model points in the following way:

$$\begin{aligned} x' &= a_1x_1 + a_2y_1 + a_3x_2 + a_4 \\ y' &= b_1x_1 + b_2y_1 + b_3x_2 + b_4 \end{aligned} \tag{7}$$

Substituting for x_2 we obtain

$$\begin{aligned} x' &= a_1x_1 + a_2y_1 + \\ &\quad a_3(sr_{11}x_1 + sr_{12}y_1 + sr_{13}z_1 + t_x) + a_4 \\ y' &= b_1x_1 + b_2y_1 + \\ &\quad b_3(sr_{11}x_1 + sr_{12}y_1 + sr_{13}z_1 + t_x) + b_4 \end{aligned} \tag{8}$$

and rearranging these equations we obtain

$$\begin{aligned} x' &= (a_1 + a_3sr_{11})x_1 + \\ &\quad (a_2 + a_3sr_{12})y_1 + (a_3sr_{13})z_1 + (a_3t_x + a_4) \\ y' &= (b_1 + b_3sr_{11})x_1 + \\ &\quad (b_2 + b_3sr_{12})y_1 + (b_3sr_{13})z_1 + (b_3t_x + a_4) \end{aligned} \tag{9}$$

Using these equations we can derive all the parameters of the transformation between the model and the image. Assume the image is obtained by a rotation U, translation $\mathbf{t}_n$, and scaling s_n. Using the orthonormality constraint we can first derive the scale factor

$$\begin{aligned} s_n^2 &= (a_1 + a_3sr_{11})^2 + (a_2 + a_3sr_{12})^2 + (a_3sr_{13})^2 \\ &= a_1^2 + a_2^2 + a_3^2s^2 + 2a_3s(a_1r_{11} + a_2r_{12}) \end{aligned} \tag{10}$$

Note that we can also extract the scale factor by applying the same constraint to the b's:

$$s_n^2 = b_1^2 + b_2^2 + b_3^2s^2 + 2b_3s(b_1r_{11} + b_2r_{12}) \tag{11}$$

We can use the two equations to verify that the weak-perspective approximation is valid. The orthogonality constraint (Eq. 6) can also be used for the this purpose. From Equations (9) and (10), by deriving the components of the translation vector, $\mathbf{t}_n$, we can obtain the position of the robot in the image relative to its position in the model views:

$$\begin{aligned} \Delta x &= a_3t_x + a_4 \\ \Delta y &= b_3t_y + b_4 \\ \Delta z &= t_z\left(\frac{1-\frac{1}{s_n}}{1-\frac{1}{s}}\right) \end{aligned} \tag{12}$$

Note that Δz is derived from the change in scale of the object. The rotation matrix U between P_1 and P' is given by

$$\begin{aligned} u_{11} &= \frac{a_1 + a_3sr_{11}}{s_n} & u_{21} &= \frac{b_1 + b_3sr_{21}}{s_n} \\ u_{12} &= \frac{a_2 + a_3sr_{12}}{s_n} & u_{22} &= \frac{b_2 + b_3sr_{22}}{s_n} \\ u_{13} &= \frac{a_3sr_{13}}{s_n} & u_{23} &= \frac{b_3sr_{23}}{s_n} \end{aligned} \tag{13}$$

As has already been mentioned, the position of the robot is computed here relative to the position of the camera when the first model image, P_1, was acquired. Δx and Δz represent the motion of the robot from P_1 to P', and the rest of the parameters represent its 3D rotation and elevation. To obtain this relative position the transformation parameters between the model views, P_1 and P_2, are required. Consequently, positioning, unlike localization, requires calibration of the model images.

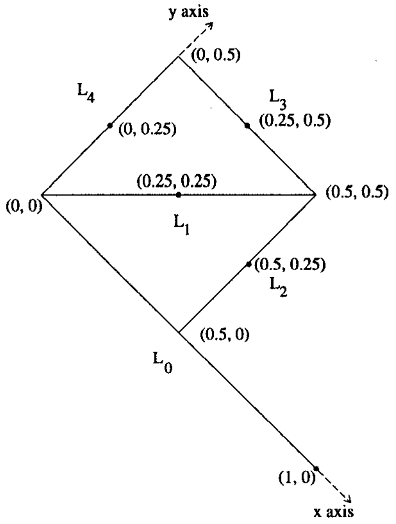

Figure 3: The definition of a token for line L_0. The parameters of the four neighbouring lines are expressed in a coordinate system with L_0 as the unit x axis. Thus, the token is the set $\{L_1, L_2, L_3, L_4\}$ which has the value { (0.25, 0.25, 0.25, 0.25), (0.5, 0.25, 0, 0.25), (0.25, 0.5, 0.25, 0), (0, 0.25, 0, 0.25) } (before mapping to a uniform normalized feature space).

4 Correspondence

Our algorithm for correspondence among views exploits the idea of image-independent tokens as it appears in [6]. We use the term ***token*** to refer to the view-independent data type representing a scene feature to be corresponded. The term ***primitive*** is used to refer to the parameterized object extracted at a particular view, corresponding to the scene feature as seen at the particular view. The purpose of the tokens is to provide robustness to missing primitive data for a scene feature at a particular novel view. Several authors have described solutions to the problem of motion correspondence that reduce the search space for correspondences by first attempting to match whole groups of primitives, in a hierarchy, to each other [17, 24, 13, 5]. This has the desirable effect of matching objects to one another that are more distinguishable from each other than the constituent primitives.

We also exploit this idea that groupings of individually unreliable primitives are easier to match unambiguously to each other. We describe each primitive by its context. Specifically, the token for each primitive P consists of the set of parameter vectors for each neighbouring primitive Q within some image distance of P, ***expressed in P-centered coordinates.*** The coordinate system used is one of the two possible Cartesian right-handed systems defined with P as the unit x axis. Use of P-centered coordinates has the advantage of giving a token representation that is invariant to scale and camera rotation. This is useful, because it reduces the degree of token variation due to camera rotation about any axis, or translation along the optical axis. Use of orthogonal coordinate axes rather than a more general affine coordinate system has the advantage of depending on one fewer basis points. In our trials, the primitive parameter vectors are four-dimensional. The four dimensions are parameters giving the centre position (x_c, y_c) and length and orientation information (dx, dy) for each line segment extracted from the image data. Figure 3 illustrates the definition a typical token.

Figure 4 shows a cross section through the normalized four-dimensional feature space describing each token. The raw parameters are mapped into a unit hypercube by performing a histogram equalization on each dimension, replacing each parameter with one one hundredth of its percentile in a large

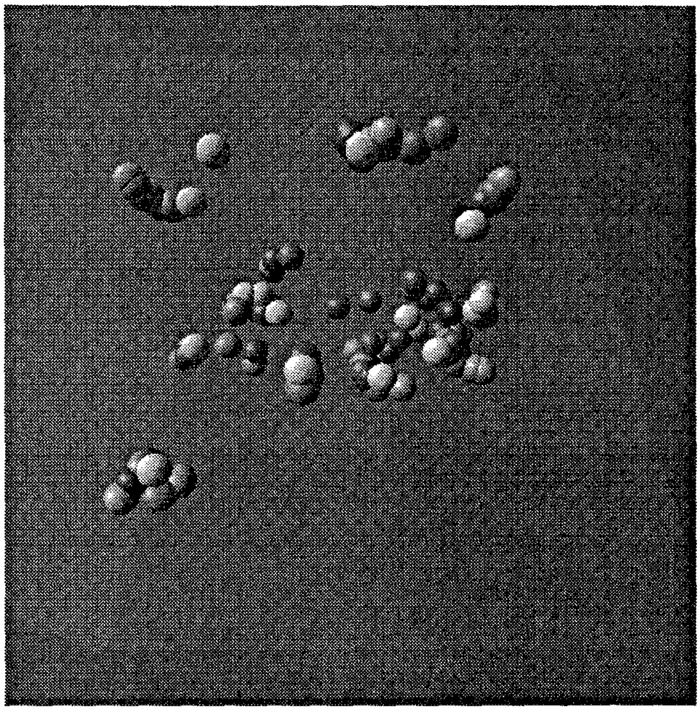

Figure 4: A cross section through a line-segment feature space, showing the line segments recovered from each of eight views, expressed in P-centered coordinates. Spheres of the same shade represent feature vectors recovered from the same frame. The unit cube in feature space is also shown.

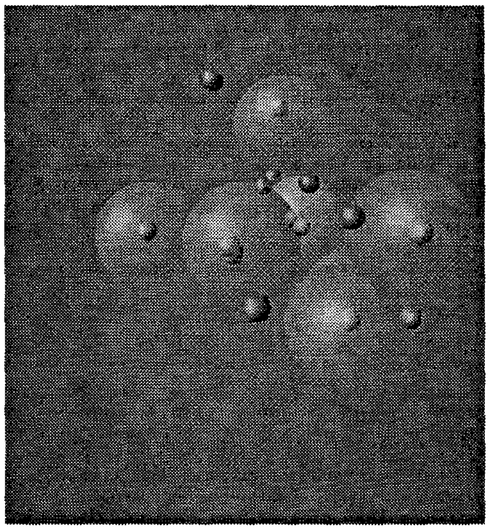

Figure 5: Edges of a new frame showing unrelated coordinates (dark) and correct coordinates (light), with token superimposed. This demonstrates how a context of multiple edges gives strong correspondences in spite of variable, extra or missing data.

data set. The data set used for this computation can be any large sample of line segments from the application domain, expressed in each other's coordinate systems. The purpose of the computation is to spread the vectors as evenly as possible in the feature space, so that distance metrics in the space are as uniform and isotropic as possible. This is useful for purposes of display of the dataset, and so that simple distributions can be expected to provide reasonable models of parameter variation for individual line segments. If the feature vector dimensions had shown significant correlation, a re-expression of the vectors using principal components may have been desirable. The shaded balls in figure 4 show the feature vectors of primitives neighbouring the primitive P to be corresponded, in P-centered coordinates, for sample data from Wilkes' active object recognition application [25]. Balls of the same shade show primitive positions for a single view. Each cluster in the figure represents a single primitive. The objective of the correspondence algorithm is to describe each cluster with a simple probability distribution. Our model describes each cluster with a separate probability density function, with an additional distribution describing the outlier points.

Letting k be the number of neighbouring primitives to be used in the description of each tracked primitive, we model each token as a set of $k+1$ probability density functions $f_0, f_1, ...f_k$ and $k+1$ probabilities $p_0, p_1, ...p_k$. Our algorithm sets the probabilities and parameterizes the distributions described by the f_i so that a primitive extracted for the token will fall at a position $\mathbf{x}$ in the feature space with probability

$$\sum_{i=0}^{k} p_i f_i(\mathbf{x}) \tag{14}$$

Thus, we may think of the probabilities p_i as providing the relative frequency with which a primitive is generated with the distribution f_i.

There are two different approaches to determining a correspondence between primitives in a new model view or unknown image and the tokens maintained by the algorithm. Both maximize the function $p(P|T)$, where

$$p(P|T) = \prod_{\substack{\textit{primitives } Q \textit{ in} \\ P-\textit{centered coordinates}}} \left(\sum_{\textit{distributions } i \in T} p_i f_i(Q) \right) \tag{15}$$

One possibility is to find, for each token T, the best primitive P. Another is to find, for each primitive P, the best token T. Which method is chosen depends on whether one is attempting to account for all of the image data, or all of the tokens. In our application, we are concerned about reliable correspondence of only the most prominent tokens, so the first method is more appropriate.

Figure 5 shows the 8 longest edges from one view represented in two coordinate systems. The dark points are for a coordinate system based on a line that does *not* correspond to the token shown. The lighter points are for the correct coordinate system. The top eight distributions of the token are shown. The correct coordinate system clearly provides the better match, and in fact gives the correct correspondence over all other possible coordinate systems.

5 Trials of the integrated system

Trials of the system integrating the correspondence and model-view-based self-location are being conducted in the AECL industrial bay, shown in figure 6. This is a large, open area in an engineering laboratory used for testing and design of components for the Canadian nuclear industry. The open area is approximately 130 metres long and 15 metres high. It accommodates test rigs of various sizes, mockups of reactor components, a machine shop, fabrication and assembly areas. The environment is ideal for tests of robotic equipment designed for industrial use, because it contains both wide open and very cramped spaces. The large distances encountered in the bay are difficult for many active sensors, but favour the

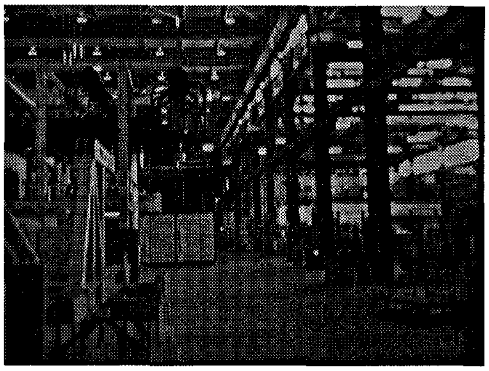

Figure 6: The AECL Industrial Bay

use of vision-based navigation. The open areas are ideal for the model view-based approach, since good opportunities exist in many places to view shallow structure from a reasonably large distance, approximating a weak-perspective projection of the structure. There are a few cramped areas in which nothing but a forest of pipes and beams is visible. These areas are problematic in that the weak perspective approximation breaks down.

We are still early in the testing and refinement process. Our low-level edge extraction has some difficulty coping with the extremes of high and low contrast encountered in the bay. As a result, we have been applying the model view solution technique to only four to six of the most reliably corresponded points in the model and unknown views. The reliability of the correspondence is given quantitatively by the scaled probabilities returned by the correspondence technique. The solutions we obtain are as accurate as our independent measurements for camera orientation, but less accurate for camera position, giving the correct betweenness relationships among the views, but fairly large error. The error can be improved with a finer spacing of model views, better low-level processing, or use of more image points.

Figure 7 shows an example, giving the model and novel views and their relative positions. The model views used in a view pair are taken from the limits of region of floor that the views are to cover. The intention is that the model view-pair bracketing the robot position (as determined by the robot's dead-reckoning capability) will be used to determine the robot's position more precisely, in order to correct the dead-reckoning estimate. Figure 8 shows the positions from which each view was taken, and the estimated position of the unknown view.

6 Discussion

By combining work on token-based correspondence with model view-based localization, we have demonstrated the potential of the method for navigation in large, open environments. There is much work yet to do. First, additional tests will be conducted as soon as assembly of some remaining components for the test robot is complete. Second, we need to enhance the low-level vision in order to provide more stable positions for line endpoints in the presence of clutter and contrast extremes. Finally, there are remaining issues to be addressed in order to automate completely the acquisition of the representation of the environment.

Acknowledgements Funding for this work was provided, in part, by the ARK Project, which receives its funding from PRECARN Associates, Inc., the Department of Industry, Science and Technology Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited.

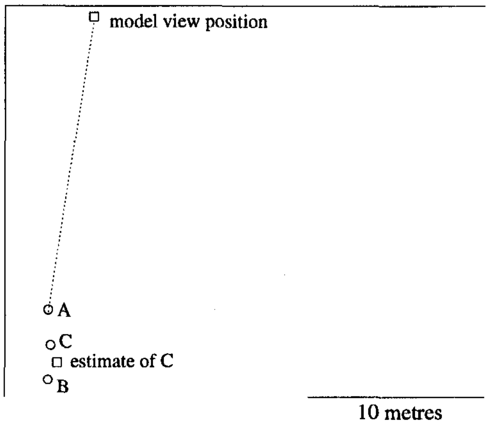

Figure 8: The positions for each view.

References

[1] Audenaert, K., Peremans, H., Kawahara, Y., Van Campenhout, J., "Accurate Ranging of Multiple Objects using Ultrasonic Sensors," *Proc. 1992 IEEE International Conference on Robotics and Automation*, pp. 1733-38.

[2] Basri, R., and Rivlin, E., "Localization using Combinations of Model Views," *Proc. IEEE 4th International Conference on Computer Vision (ICCV-93), Berlin*, 1993, pp. 226–230.

[3] Beveridge, J.R., and Riseman, E.M., "Optimal Geometric Model Matching Under Full 3D Perspective," *Proceedings of the Second IEEE Workshop on Directions in Automated CAD-Based Vision*, IEEE Computer Society Press, 1994.

[4] Bozma, O., Kuc, R., "Building a Sonar Map in a Specular Environment Using a Single Mobile Sensor," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 13, No. 12. December 1991.

[5] Chung, R.C.K., and Nevatia, N., "Use of Monocular Groupings and Occlusion Analysis in a Hierarchical Stereo System," *Proc. IEEE Conference on Computer Vision and Pattern Recognition '91*, IEEE Computer Society Press, Washington, 1991.

[6] Crowley, J.L., Stelmaszyk, P. and Discours, C., "Measuring Image Flow by Tracking Edge-Lines," *Proc. Second International Conference on Computer Vision*, IEEE Computer Society Press, Washignton, 1988.

[7] Holland, J.M., "Programming the Navmaster (TM) for Autonomous Navigation," Product tutorial documentation, Cybermotion, Inc., Roanoke, VA. 1989.

[8] Holenstein, A.A., Müller, M.A., Badreddin, E., "Mobile Robot Localization in a Structured Environment Cluttered with Obstacles," *Proc. 1992 IEEE International Conference on Robotics and Automation*, May 1992, Nice, France, pp. 2576-81.

[9] Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E., "Image-Based Navigation using 360deg Views", Proceedings, Image Understanding Workshop, Morgan Kaufmann, San Mateo, CA, 1990, pp 782–791.

[10] Krotkov, E., "Mobile Robot Localization Using a Single Image," *Proc. 1989 IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, Washington, 1989, pp. 978-983.

[11] Kumar, R., "Model Dependent Inference of 3D Information from a Sequence of 2D Images," *COINS TR92-04*, Department of Computer Science, University of Massachusetts at Amherst, 1992.

[12] Leonard, J., and Durrant-Whyte, H., "Mobile Robot Localization by Tracking Geometric Beacons," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp. 376-82. June 1991.

[13] Lim, H.S., and Binford, T.O., "Structural Correspondence in Stereo Vision," *Proc. DARPA Image Understanding Workshop*, Cambridge, MA, 1988.

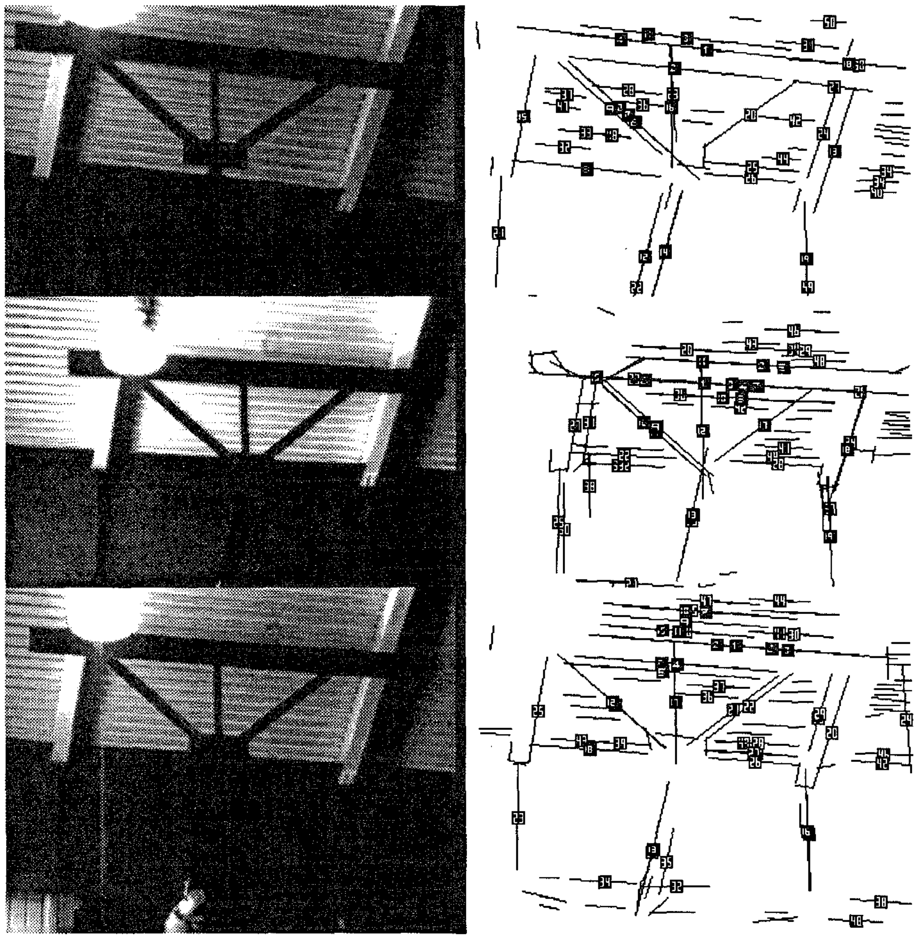

Figure 7: An example trial. Two model views (top two images) and an unknown view (bottom image). Camera orientation was recovered to within our own position measurement errors. Camera position errors are larger, due to the availability of only a few reliable feature points. The positions from which each was taken are shown on the map, along with the position estimate for the unknown view. The positioning was done using the endpoints of the three best-corresponded lines: A21/B25/C23, A15/B27/C25 and A8/B22/C18.

[14] Lim, J.H., and Cho, D.W., "Physically Based Sensor Modelling for a Sonar Map in a Specular Environment," *Proc. 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992. pp. 1714-19.

[15] Nickerson, S.B., Jenkin, M., Milios, E., Down, B., Jasiobedzki, P., Jepson, A., Terzopoulos, D., Tsotsos, J, Wilkes, D., Bains, N., Tran, K., "ARK: Autonomous Navigation of a Mobile Robot in a Known Environment," *Proc. IAS-3*, Pittsburgh, 1993.

[16] Poggio, T., "3D Object Recognition: On a Result by Basri and Ullman," Technical Report 9005-03, ISRT, Povo, Italy, 1990.

[17] Quan, L., and Mohr, R., "Matching Perspective Images Using Geometric Constraints and Perceptual Grouping," *Proc. Second International Conference on Computer Vision*, IEEE Computer Society Press, Washington 1988.

[18] Wilkes, D., Dudek, G., Jenkin, M., and Milios, E., "Multi-transducer sonar interpretation", *IEEE Int. Conf. on Robotics and Automation*, Atlanda, GA, 1993.

[19] Sugihara, K., "Some Location Problems for Robot Navigation using a Single Camera," *Computer Vision, Graphics, and Image Processing*, Vol. 42, No. 1, April 1988.

[20] Wilkes, D., *Active Object Recognition*, Ph.D. Thesis, Department of Computer Science, University of Toronto, 1994.

[21] Tsuji, S. and Zheng, J., "Qualitative Representation of Scenes Along a Route", Proceedings, AAAI Workshop on Qualitative Vision, July, 1990, pp 67–71.

[22] Tsumura, T., "Survey of Automated Guided Vehicle in Japanese Factory," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, Los Angeles, 1986.

[23] Ullman, S., and Basri, R., "Recognition by Linear Combination of Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp 992–1006.

[24] Venkateswar, V., and Chellappa, R., "Hierarchical Feature Based Matching for Motion Correspondence," *Proc. IEEE Workshop on Visual Motion 1991*, IEEE Computer Society Press, Los Alamitos, CA, 1991.

[25] Wilkes, D., and Tsotsos, J., "Active Object Recognition," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition '92*, IEEE Computer Society Press, Urbana, Illinois, June 1992.