Learning Visual Landmarks for Pose Estimation

Robert Sim and Gregory Dudek
Centre for Intelligent Machines
McGill University
3480 University St, Montreal, Canada H3A 2A7

Abstract— **We present an approach to vision-based mobile robot localization, even without an *a-priori* pose estimate. This is accomplished by *learning* a set of visual features called image-domain *landmarks*. The landmark learning mechanism is designed to be applicable to a wide range of environments. Each landmark is detected as a local extremum of a measure of uniqueness and represented by an appearance-based encoding. Localization is performed using a method that matches observed landmarks to learned prototypes and generates independent position estimates for each match. The independent estimates are then combined to obtain a final position estimate, with an associated uncertainty. Quantitative experimental evidence is presented that demonstrates that accurate pose estimates can be obtained, despite changes to the environment.**

I. Introduction

In order for a mobile robot to perform its assigned tasks, it often requires a representation of its environment, knowledge of how to navigate in its environment, and a method for determining its position in the environment. These problems have been characterized by the three fundamental questions of mobile robotics, that is "Where am I?", "Where am I going?" and "How can I get there?". This paper addresses the first question, that of position estimation for a robot located in a previously explored region of the environment. The robot is equipped with a single achromatic camera, and does not require an *a priori* estimate of its position. An accurate position estimate is desired without any motion on the part of the robot. One might imagine that the robot must consistently re-localize itself after periodic shutdowns for maintenance. We build on previous work by Sim and Dudek which demonstrated that position estimation could be accurately performed in a more constrained environment using a similar technique[1]. In this paper we extend that work to show more accurate results, an approach to orientation recovery, and robustness to modifications to the environment.

Our approach to the problem at hand uses visual features, referred to as *landmarks*, to perform position estimation, extracting these landmarks from a preliminary traversal of the environment. In this work, landmarks are *image-domain* features, as opposed to interpreted characteristics of the scene. *Candidate landmark* selection is based on a local distinctiveness criterion; this is later validated by verifying the appearance of the candidate landmarks against a set of landmark prototypes. In contrast to methods such as Markov localization[2], this method avoids an *a priori* discretization of the state space and the associated tradeoff between accuracy and high computational costs. Rather, our method delivers highly accurate pose estimates with low computational cost in both space and time. We also obtain partial illumination invariance.

A. Outline of the Paper

Section II presents a general discussion of existing solutions and other related work. Section III introduces our method for position estimation with a general overview. The model of visual attention that is employed for feature extraction is presented in Section IV. Section V presents our method for learning landmarks for the purposes of localization and Section VI presents the online method for employing the learned landmarks in order to obtain a position estimate. Finally, the experimental results are presented in Chapter VII. Section VIII concludes with a discussion of the experimental results and possible directions for future work.

II. Previous Work

Early solutions to the localization problem employed geometric triangulation methods first developed by cartographers and navigators. Sugihara, Krotkov, and Avis and Imai each consider the problem of achieving landmark correspondence given a map of known landmarks and a set of bearings to observed landmarks [3], [4], [5]. Sutherland and Thompson and Betke and Gurvits approach triangulation methods from the perspective that the landmark correspondence problem has been solved, but the bearings to the observed landmarks cannot be precisely known [6], [7]. In all of these works, the problem of reliably extracting landmarks from sensor data is ignored.

Given that it is often difficult to reliably extract landmarks from sensor data, a number of methods have been proposed which employ *Kalman filters* for achieving locally optimal correspondence between a map and ubiquitous, but often noisy, sensor readings. Of such methods, those employed by Smith and Cheeseman, and Leonard and Durrant-Whyte are perhaps the best known [8], [9]. One difficulty with Kalman filtering techniques is that they tend to rely on a good *a priori* estimate and therefore can fail to converge to the correct solution. Other methods for achieving optimal correspondence between sensor measurements and a map include Lu and Milios, Beveridge, Weiss and Riseman, and Boley, Steinmetz and Sutherland [10], [11], [12]. Of particular note is work by Thrun, Fox and Burghard which derives a Markov-based solution which subsumes the Kalman filter [2]. These works all compute a globally optimal pose estimate which presents issues of computational efficiency and tractability.

A number of researchers have developed methods which avoid the use of explicit features or maps. These methods express the sensor data as a function of the pose of the robot, and attempt to invert this function. One such technique, employed in work by Nayar, Murase and Nene, Belhumeur, Hespana and Kriegman, and Jepson and Black is Principal Components Analysis (PCA) [13], [14], [15]. These methods are similar to the Kalman Filter in that they rely on a linear approximation to the underlying behaviour of the data, yet they differ in that they do not rely on explicitly interpreted features but linearize the statistical variation of the data in order to choose maximally discriminating features, which are unlikely to hold any explicit semantic value.

Other work which employs sensor inversion include Dudek and Zhang, and Oore, Hinton and Dudek [16], [17], which employ neural networks to invert edge statistics and sonar charaterizations as a function of position. While neural networks have been shown to give good results for highly nonlinear or complex input, they can be difficult to tune, and tend to fail in the presence of outliers. Another significant difficulty associated with the problem of sensor inversion in general is that the function to be inverted may not be not one-to-one, a situation which may not be easily detected *a priori*.

III. Overview

Rather than relying on unstable interpretations of sensor data or confronting issues of non-invertibility and outlier sensor readings, our method seeks to characterise independent observations of significant portions of what the robot senses, and later recover those portions for pose estimation. We achieve this by employing a model of visual attention aimed at extracting the parts of a scene which are distinctive, and characterizing those parts as a function of the robot's pose. In so doing, we can exploit an assumption of local linear variation in the sensor data– an assumption which is far less constraining than one of global linear variation. Furthermore, such a characterization should be robust to local changes in the environment, or partial sensor occlusion.

Our method consists of two phases. In an initial, offline *learning* or *exploration* phase, a set of landmarks is extracted from image data and grouped for future recognition. The learned groups, referred to as *tracked landmarks*, are encoded using a principal components representation of appearance, which is later exploited for charaterising the landmark as a function of position. The on-line phase, which is employed whenever the robot requires a pose estimate, consists of detecting and classifying landmarks from the robot's current observations, and thereby computing a pose estimate from the characterization of the landmark. An outline of the method is depicted in Figure 1, and described below.

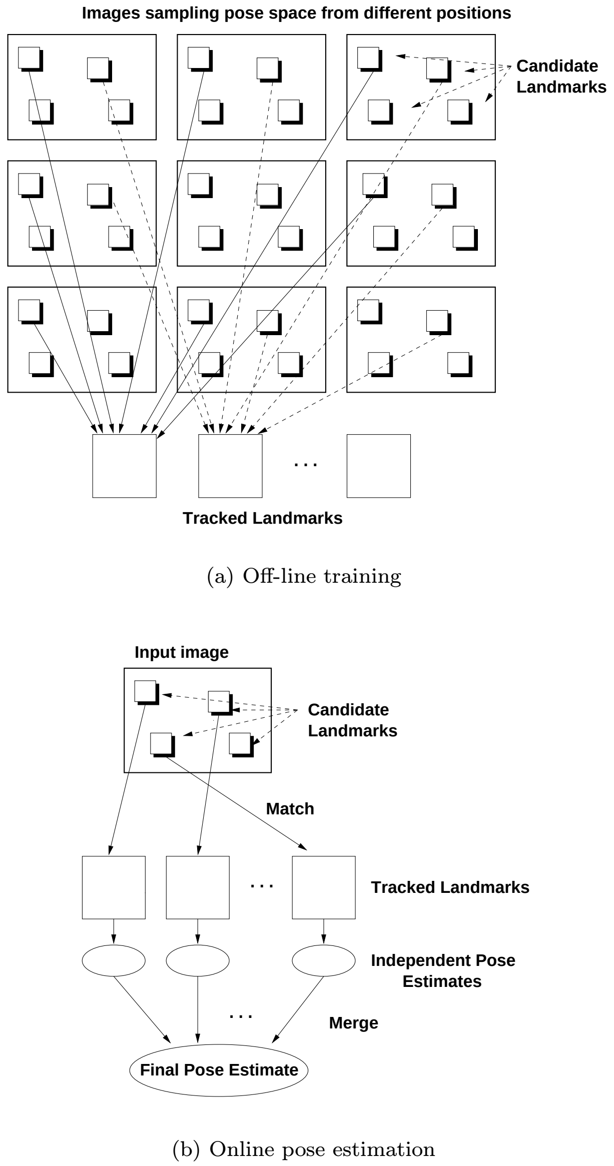

Fig. 1. An overview of the method.

- Off-line "Map" construction (Figure 1(a)):
  1. Training images are collected sampling a range of poses in the environment.
  2. *Landmark candidates* are extracted from each image using a model of visual attention.
  3. *Tracked Landmarks* are extracted as sets of candidate landmarks over the configuration space. Tracked landmarks are each represented by a characteristic prototype, obtained by encoding an initial set of candidate landmarks by their principal components decomposition. For each image, a local search is performed in the neighbourhood of the candidate landmarks in order to locate optimal matches to the templates.
  4. The set of tracked landmarks is stored for future retrieval.
- On-line localization (Figure 1(b)):
  1. When a position estimate is required, a single image is acquired from the camera.
  2. Candidate landmarks are extracted from the input

image using the same model of visual attention used in the off-line phase.

3. The candidate landmarks are matched to the learned templates using the same method used for tracking in the off-line phase.
4. A position estimate is obtained for *each* matched candidate landmark. This is achieved by computing a reconstruction of the candidate based on the decomposition of the tracked candidates and their known poses in the tracked landmark. The result is a position estimate obtained as a linear combination of the views of the tracked candidates in the tracked landmarks.
5. A final position estimate is computed as the robust average of the individual estimates of the observed candidates.

In practice we use a statistical measure of local image content for candidate landmark extraction. Good candidates for a statistical measure include saliency measures such as edge density, or local symmetry, or the output of a matched filter. For the purposes of this work we employ a measure of edge density. Such a measure has strong local structure in the sense that the output tends to vary smoothly under local changes in camera pose. The objective of this definition is to produce observed landmarks which are reasonably stable and repeatable image features, distinctive in appearance and containing a rich body of information concerning the structure of the image as a whole. Furthermore, such characteristics can be rapidly and easily extracted from an image, with the added benefit that they tend to be *robust to variations in illumination conditions*.

IV. Visual Landmarks

Edge data is often employed in computational vision to extract geometric information from a scene while providing robustness to illumination effects. It is well known, however, that the interpretation of the putative edge elements in an image is complex and subject to instability [18], [19], [20]. The *distribution* of edge elements in a scene, however, is closely related to basic scene structure, and yet can offer greater stability for tracking. Furthermore, the edge element distribution shares similar advantages with the underlying edge map, such as robustness to variations in illumination. One can also expect that a local description of the edge distribution will vary smoothly with changes in camera pose.

To this end, we formulate our *landmark detector* as a filter that extracts local maxima from the edge density D of the image. The purpose of the landmark detector is to locate *candidate landmarks* in an image. $D(x, y)$ is measured as the sum of the edge magnitudes over a small subwindow (on the order of 20 by 20 pixels) centred at position (x, y) in the image. Figure 2 shows the results obtained from running the landmark detector on an image obtained in our lab, with the image depicted on the left and the density function D depicted on the right. The landmark candidates are superimposed as squares. This idea is presented in greater detail in Bourque, Dudek and Ciaravola, and Sim and Dudek [21], [1].

Fig. 2. Detected Landmarks in an Image. The left image is the original and the density function D is depicted on the right, where darker intensity represents large values of D. In each image, candidate landmarks are drawn as squares.

V. Tracking

We have developed the notion of an image-domain landmark as a local maximum of edge density. A landmark represents the basic feature which we employ for localisation, a task which will be accomplished using a characterisation of the landmark's appearance as a function of the camera's position in configuration space. In order to achieve this characterisation, however, the landmark must first be *tracked*.

Our technique for landmark tracking operates as follows. Given an initial set of *prototypes*, that is, observations of a set of unique candidate landmarks, a *tracked landmark*, is constructed for each prototype by identifying matches to the prototype amongst the set of all observed landmark candidates. In practice, since landmark candidates can demonstrate local variation in position as the camera moves, a local search in the image neighbourhood of a candidate may be required. We refer to the task of matching a single candidate landmark to a prototype as landmark *recognition*, and the task of building tracked landmarks as landmark *tracking*. Figure 3 provides an overview of the training process presented thus far; candidate landmarks are detected as local maxima of edge density and then tracked into sets of tracked landmarks.

We represent the appearance of landmarks (both candidates and prototypes) using a principal components representation of the intensity image in the neighbourhood of the candidate [22], [13], [23]. For the purposes of matching and tracking, recognition is achieved by selecting the prototype with least Euclidean distance in the subspace from the candidate under consideration. Figure 4 shows a set of landmark prototypes (top), and the corresponding eigenvectors, or *eigenlandmarks* constructed from the prototypes (bottom).

In order to describe the environment, images must be obtained from representative viewpoints. For the purposes of this discussion, let us assume that we select viewpoints that cover the configuration space in a uniform grid. This is by no means a requirement or constraint, but rather a simplifying assumption. In order to achieve computational efficiency, viewpoints are selected such that the camera is

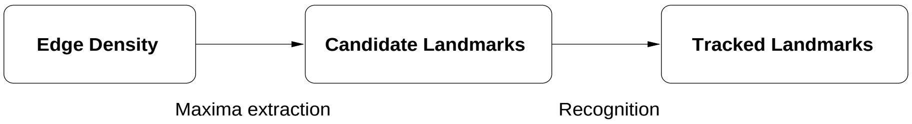

Fig. 3. The training process: Candidate landmarks are detected as local maxima of edge density and then tracked into sets of tracked landmarks.

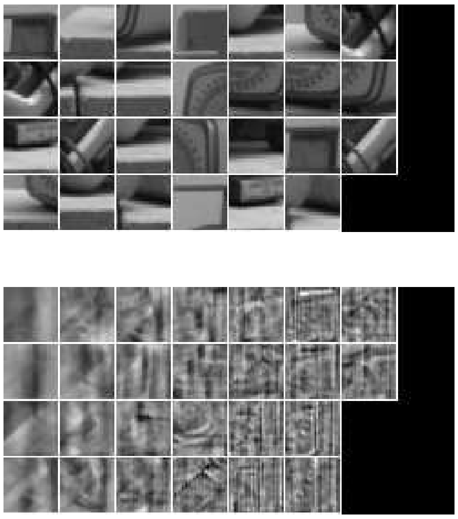

Fig. 4. Landmark Prototypes (top) and Eigenlandmarks (bottom).

facing in a consistent orientation[1]. Once the sample images have been acquired, they are used to automatically learn a suitable set of tracked landmarks for subsequent positioning.

The set of tracked landmarks is initially defined by the set of single candidate landmarks observed in a selected *bootstrap* image from the database. These candidate landmarks, which become prototypes for matching, are selected in this manner in order to guarantee uniqueness. Typically, we select the initial bootstrap image to be the one that is taken from a camera position closest to the centroid of all visited camera positions. Given this initial set of prototypes, the candidate landmarks in each of the remaining images are considered for inclusion in one of the tracked landmarks. Consideration for inclusion in a set is based on the following methodology:

Algorithm 1 (Tracking algorithm for a single image.)

1. For each landmark l_i in the image, and
   (a) for each prototype t_j in the database,
      i. perform a local search in the neighbourhood of l_i in the image for a better match to t_j. If a better match l' is found, it replaces l_i as a candidate match to t_j.
   (b) Select the prototype t_j for which the best match to l_i was found in step 1a.
2. If l_i is the best match to t_j over all other landmarks in the image and l_i matches t_j within a reasonable threshold, add it to the tracked landmark represented by t_j, otherwise, create a new tracked landmark with l_i as the prototype.

[1]While this constraint can be readily relaxed, we will later demonstrate a method for estimating orientation under the conditions that the database orientation is fixed.

The goal of this method is to grow landmark sets as much as possible in configuration space so that a candidate landmark can be matched to the correct target over a large portion of the space. The local search in the neighbourhood of l_i is performed in order to counter the effects of any instabilities in the underlying landmark detector. Figure 5 shows a typical landmark set. Each thumbnail image corresponds to the landmark as detected in the image taken at the corresponding grid position in configuration space. Grid positions with no corresponding thumbnail image indicate positions in the configuration space where no landmark candidate was found that matched the prototype.

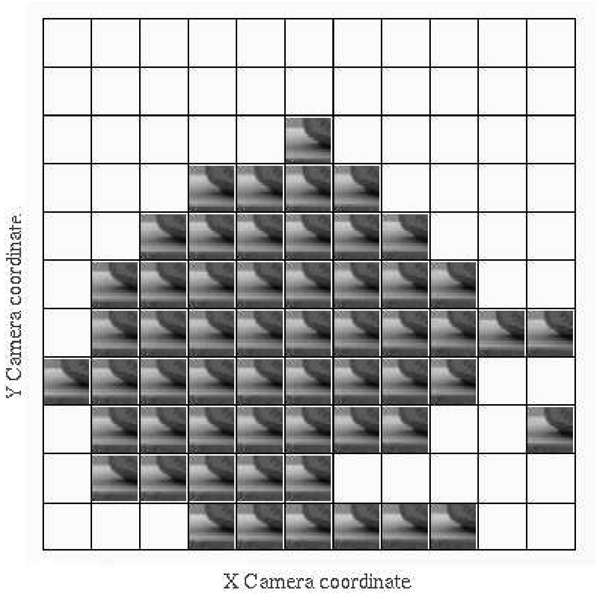

Fig. 5. A typical landmark set. Each thumbnail corresponds to the landmark as detected in the image taken at the corresponding grid position in camera space.

A *tracked landmark* is the essential modelling primitive that defines the "map" and which is used for subsequent correspondence and position estimation. It should be noted that the tracking method makes no assumptions regarding position within the image, which somewhat relaxes some constraints that could be imposed on the pose of the camera.

VI. Pose Estimation

On-line localisation is performed by matching candidate landmarks from the robot's current view to the tracked landmarks, and interpolating a parameterization of the set of tracked candidates. This section discusses the position estimation procedure given that the association between a candidate landmark and a tracked landmark is known. We then consider a method for combining the individual position estimates from several matches to obtain a robust estimate.

When a position estimate is required, an image is obtained and landmarks are extracted by selecting the local maxima of edge density, and the extracted candidates are then matched to the *tracked landmarks* in the database, using the procedure outlined above. Once landmark matching is accomplished, we exploit an assumption of local linear variation in the landmark characteristics with respect to camera pose in order to obtain a position estimate. If this assumption is true, then the encoding of the landmark observed from an unknown camera position is a linear combination of the encodings of the tracked models, allowing us to *interpolate* between the sample positions in the database. We will later present a method for quantitatively evaluating the reliability of the linearity assumption, which will allow us to obtain a measure of confidence in the results.

Let us define the *encoding* $\mathbf{k}_l$ of a landmark candidate l as the projection of the intensity distribution in the image neighbourhood represented by l into the subspace defined by the principal components decomposition of the set of all tracked landmark prototypes.

$$\mathbf{k}_l = \mathbf{U}^T \mathbf{l} \tag{1}$$

where $\mathbf{l}$ is the local intensity distribution of l normalised to unit magnitude and $\mathbf{U}$ is the set of principal directions of the space defined by the tracked landmark prototypes.

Let us now define a *feature-vector* $\mathbf{f}$ associated with a landmark candidate l as the principal components encoding $\mathbf{k}$, concatenated with two vector quantities: the image position $\mathbf{p}$ of the landmark, and the camera position $\mathbf{c}$ from which the landmark was observed:

$$\mathbf{f} = \left| \mathbf{k} \quad \mathbf{p} \quad \mathbf{c} \right| \tag{2}$$

where, in this particular instance alone, the notation $|\mathbf{a}\,\mathbf{b}|$ represents the concatenation of the vectors $\mathbf{a}$ and $\mathbf{b}$.

Given the associated feature vector $\mathbf{f}_i$ for each landmark l_i in the tracked landmark $T = \{l_1, l_2, \ldots, l_m\}$, we construct a matrix $\mathbf{F}$ as the composite matrix of all $\mathbf{f}_i$, arranged in column-wise fashion, and then take the singular values decomposition of $\mathbf{F}$,

$$\begin{aligned} \mathbf{F} &= \left[\begin{array}{ccc} \mathbf{f}_1 & \cdots & \mathbf{f}_n \end{array} \right] \\ &= \mathbf{U}_F \mathbf{W} \mathbf{V}^T \end{aligned} \tag{3}$$

to obtain $\mathbf{U}_F$, representing the set of eigenvectors of the tracked landmark T arranged in column-wise fashion. Note that since $\mathbf{c}_i$ is a component of each $\mathbf{f}_i$, $\mathbf{U}_F$ encodes camera position along with appearance. Now consider the feature vector $\mathbf{f}_l$ associated with l, the observed landmark for which we have no pose information - that is, the $\mathbf{c}$ component of $\mathbf{f}_l$ is undetermined. If we project $\mathbf{f}_l$ into the subspace defined by $\mathbf{U}_F$ to obtain

$$\mathbf{g} = \mathbf{U}_F^T \mathbf{f}_l \tag{4}$$

and then reconstruct $\mathbf{f}_l$ from $\mathbf{g}$ to obtain the feature vector

$$\mathbf{f}'_l = \mathbf{U}_F \mathbf{g} \tag{5}$$

then the resulting reconstruction $\mathbf{f}'_l$ is augmented by a camera pose estimate that interpolates between the nearest eigenvectors in $\mathbf{U}_F$. In practice, the initial value of the undetermined camera pose, $\mathbf{c}$ in $\mathbf{f}_l$ will play a role in the resulting estimate and so we substitute the new value of $\mathbf{c}$ back into $\mathbf{f}_l$ and repeat the operation, reconstructing $\mathbf{f}'_l$ until the estimate converges to a steady state. This repeated operation, which constitutes the recovery of the unknown $\mathbf{c}$ is summarised in Figure 6.

Figure 7 illustrates a set of estimates obtained for the landmarks detected in a single image. While most of the estimates are reasonably accurate, one observation is clearly an outlier, most likely produced by nonlinearities in the tracked landmark, poor tracking, or a match that is altogether incorrect. We now consider a method for combining the individual estimates obtained from each observed candidate landmark, taking into consideration the presence of outliers and the reliability of the tracked landmarks.

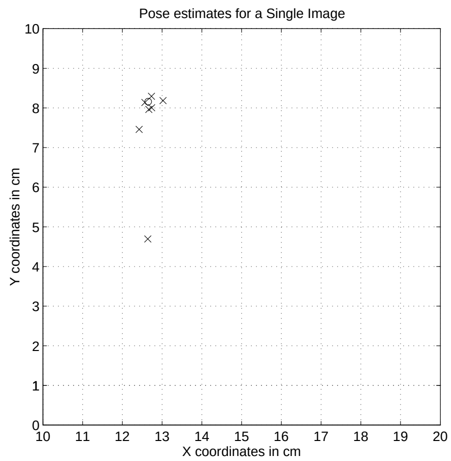

Fig. 7. Position estimate for a single test image. Each 'x' marks an estimate as obtained from a single landmark in the image. The 'o' at position $(12.7, 18, 2)$ represents the true position. The training images were obtained at the locations of the grid intersections.

We employ the approach used by Smith and Cheeseman for combining estimates with associated error models [8]. An error model for a particular tracked landmark T is constructed using *cross-validation*[24]. That is, we measure how well each observed candidate landmark in T is predicted by the rest of the candidate landmarks in T. This is a quantity which is fixed for a given tracked landmark, and hence can be computed *a priori*. The error model E for T is then described as an *approximate transform* (AT)

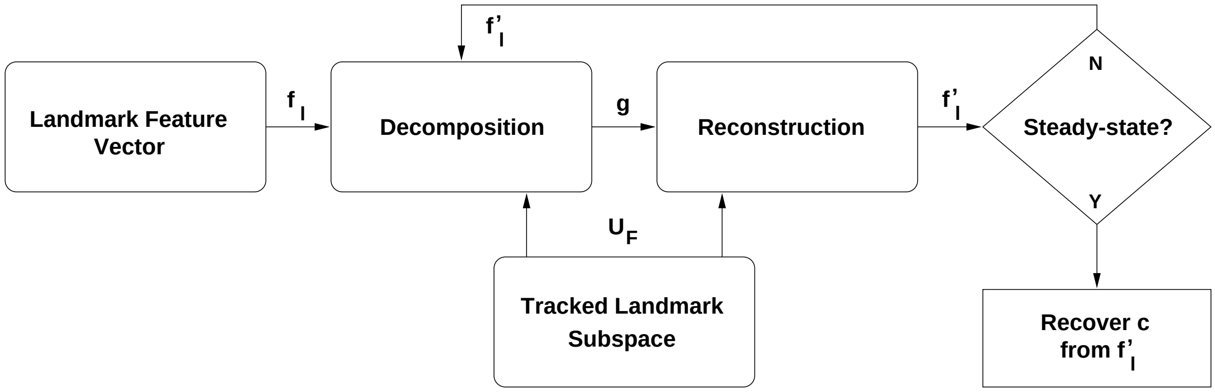

Fig. 6. The recovery operation. The unknown camera position **c** associated with a landmark l is recovered by repeatedly reconstructing the landmark feature vector in the subspace defined by the matching tracked landmark.

with two components, $\hat{\mathbf{X}}$ being the the average displacement from the true position $\mathbf{c}_t(i)$ for all l_i of T, and $\mathbf{C}$ being the total covariance of the set of displacements.

$$\mathbf{E} = \left\{ \hat{\mathbf{X}}, \mathbf{C} \right\} \tag{6}$$

Outlier detection is performed by finding the median position estimate $\hat{\mathbf{X}}_m$, and computing a median covariance, $\mathbf{C}_m$ from the set of predictions and their associated covariances (recall that the set of predictions is defined by the predictions computed for each candidate landmark observed in the image). $\mathbf{C}_m$ defines an ellipsoidal region of configuration space, the scale of which is controlled by the user, centred at $\hat{\mathbf{X}}_m$, within which predictions can be considered to be acceptable. Figure 8 depicts a set of position estimates (the set of all diamonds), the median estimate (the ellipse) and those estimates which are considered acceptable for merging, (the solid diamonds). The '+'s represent locations at which training images were obtained.

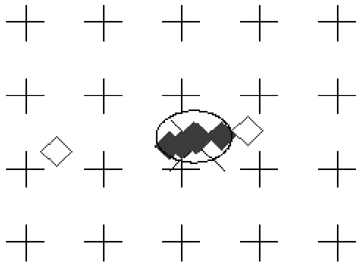

Fig. 8. A set of filtered predictions. The ellipse corresponds to the covariance of the median AT. Solid diamonds represent retained predictions whereas hollow diamonds represent rejected predictions. The '+'s represent a portion of the locations at which training images were obtained.

Once outliers have been filtered, the final step in obtaining a position estimate is to merge the individual estimates using the merging operation defined for ATs, which derives the Kalman gain of the associated covariances in order to compute a weighted average of the input estimates. For greater detail, refer to Smith and Cheeseman [8].

A. Recovering Orientation

Throughout our presentation, we have constrained the pose of the observer such that it faces in a consistent orientation. While one could conceivably train the robot in a higher dimensional configuration space, the computational and storage costs would be too high. We propose instead that orientation can be recovered given a database that is trained for only one orientation. This is accomplished by measuring the degree to which the set of independent pose estimates are consistent with one another. to this end, we employ a *consistency* measure,

$$M = \frac{C}{GPR} \tag{7}$$

where

$$C = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{8}$$

is the square-root of the sum of the variances (one for each axis of the trained configuration space) of the set of independent pose estimates obtained for each matched landmark candidate in the image, G is the percentage of independent pose estimates which are not rejected as outliers, P is the percentage of 'matched' candidate landmarks - that is, the ratio of the number of successful candidate-tracked landmark matches out of all detected landmark candidates, and finally, R is the raw number of retained independent pose estimates. Clearly, lower values of M indicate that there is good consistency between the measurements obtained from the image and the training database.

Given our consistency measure, M, we can recover the robot's orientation by rotating the robot through 360°, taking an image at each orientation (or a set of sample orientations) and finding M. The orientation at which M is minimised is considered to be the correct orientation.

Figure 9 plots M for a series of orientations taken at 10° increments from the scene considered in the experimental results. The correct orientation is correctly predicted to be 0°.

The results in Figure 9 indicate that the measure is useful for recovering the orientation of the robot when it is unknown. This result greatly increases the utility of the

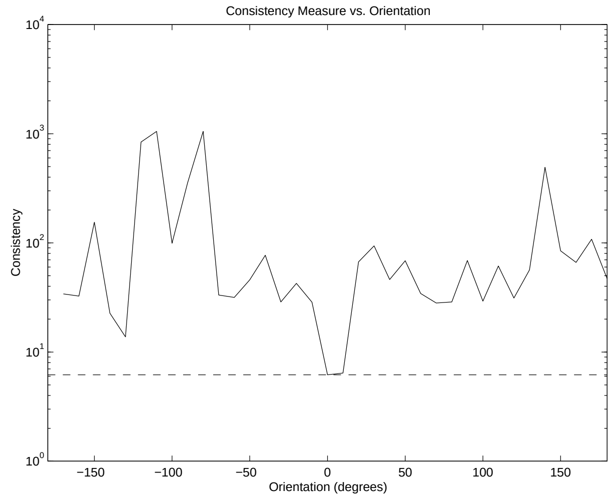

Fig. 9. The consistency measure plotted as a function of orientation. The correct orientation is 0°.

method, since the robot pose need not be constrained while online (provided that it is constrained during the training phase, which is supervised), and dead-reckoning errors in orientation can be corrected.

VII. Experimental Results

Fig. 10. An Indoor Environment

Our technique has been tested using a variety of different robots, both static and mobile. We present here results obtained using a mobile robot platform.

An indoor scene is depicted in Figure 10. In this scene, a camera was mounted on an RWI B-12 mobile robot (Figure 11). In addition, a split-beam laser was mounted on the back of the robot, and pointed at the floor, in order to obtain ground truth by accurately positioning the robot by hand to within 0.5cm of the desired pose, and oriented to within 1.0°. Training images were taken at 20.0cm intervals over a 2.0m by 2.0m grid. Despite the good dead reckoning, the unevenness of the floor led to some variation in image alignment.

Once training images were collected, a series of 30 test images were taken from random positions in order to test the method. Figure 12 presents the set of estimates obtained from the method, plotted against their ground-truth. The mean error in position is 6.3cm or 31% of the sample spacing.

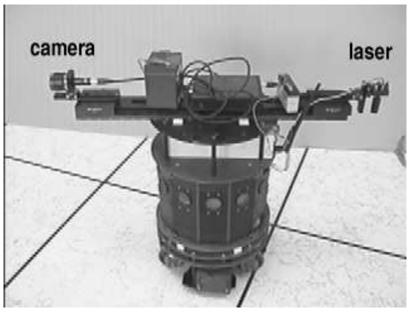

Fig. 11. The robot with mounted camera.

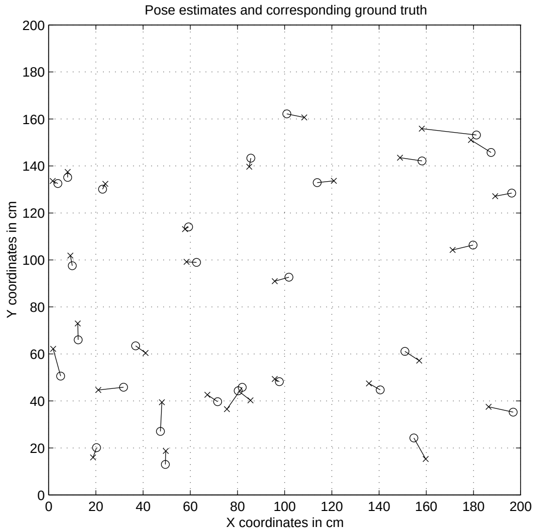

Fig. 12. The set of pose estimates obtained using the method. The mean estimation error is 6.3cm.

In order to test the claim that the method is robust under minor changes to the environment, five more test images were taken of the scene, with one of the foreground chairs moved back against the wall (Figure 13).

Figure 14 depicts the set of results obtained for the five test images. The mean error is 9.4cm. Clearly, the method works very well in the face of a change which would pose serious difficulties for many existing localisation solutions.

VIII. Conclusions

This paper has presented a method for estimating the position of a mobile robot, without an *a priori* estimate. This is accomplished by learning a set of visual features, known as *landmarks*, candidates for which are detected as local maxima of a measure of distinctiveness. Landmark candidates are then grouped into *tracked landmarks*: sets of candidates which correspond to the same visual region of the environment, as observed from different viewpoints. Grouping is achieved by matching subspace encodings of the candidates, perhaps with adjustments in position in the image in order to improve matching. Online position estimation is performed by detecting candidates and matching

Fig. 13. Altered Scene

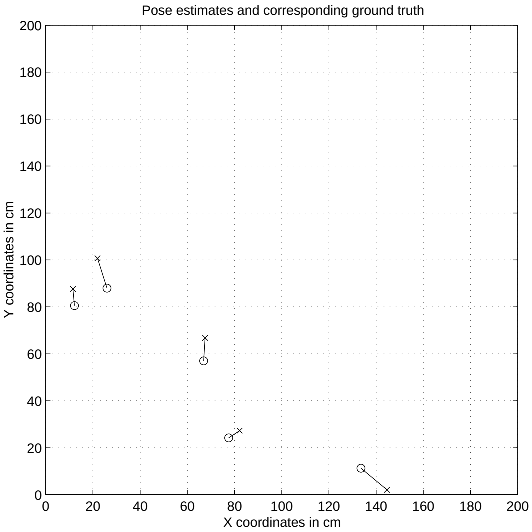

Fig. 14. Results from the altered scene. The mean estimation error is 9.4cm.

them to the tracked landmarks. Each match is used to generate a pose estimate by employing a principal components reconstruction of a feature vector which encodes both appearance and image geometry. The experimental results indicate that the method is robust for a typical indoor environment. In previous work, we demonstrated the robustness of a similar method for a more constrained environment [1].

To conclude, we have presented a method for image-based mobile robot localization which exhibits many advantages over both traditional triangulation and optimization methods and recent feature-based and principal components methods. This was achieved by exploiting the strengths of both solution domains. Experimental results indicate that the method is suitable for practical, real-world implementation.

References

[1] R. Sim and G. Dudek, "Mobile robot localization from learned landmarks", in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems(IROS)*, Victoria, Canada, October 1998, IEEE Press.

[2] S. Thrun, D. Fox, and W. Burghard, "A probabilistic approach to concurrent mapping and localization for mobile robots.", *Autonomous Robots*, vol. 5, pp. 253–271, 1998.

[3] K. Sugihara, "Some location problems for robot navigation using a single camera", *Computer Vision, Graphics, and Image Processing*, vol. 42, pp. 112–129, 1988.

[4] Eric Krotkov, "Mobile robot localization using a single image", in *Proceedings 1989 IEEE International Conference on Robotics and Automation*, pp. 978–983. 1989.

[5] D. Avis and H. Imai, "Locating a robot with angle measurements", *Journal of Symbolic Computation*, , no. 10, pp. 311–326, 1990.

[6] K.T. Sutherland and W.B. Thompson, "Inexact navigation", in *Proceedings of the IEEE*, 1993, pp. 1–7.

[7] Magrit Betke and Leonid Gurvits, "Mobile robot localization using landmarks", *IEEE Trans. on Robotics and Automation*, vol. 13, no. 2, pp. 251–263, April 1997.

[8] Randall C. Smith and Peter Cheeseman, "On the representation and estimation of spatial undertainty", *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[9] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons", *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[10] J. R. Beveridge, R. Weiss, and E. M. Riseman, "Combinatorial optimization applied to variable scale 2d model matching", in *Proceedings of the 10th International Conference on Pattern Recognition*, June 1990, pp. 18–23.

[11] F. Lu and E. E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans", in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, June 1994, pp. 935–938, IEEE Computer Society Press.

[12] D.L. Boley, E.S. Steinmetz, and K.T. Sutherland, "Robot localization from landmarks using recursive total least squares", in *Proceedings of the IEEE International Conference on Robotics and Automation, 1996*, Minneapolis, April 1996, IEEE.

[13] S.K. Nayar, H. Murase, and S.A. Nene, "Learning, positioning, and tracking visual appearance", in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994, pp. 3237–3246.

[14] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, July 1997.

[15] M. J. Black and A. D. Jepson, "Eigen tracking: robust matching and tracking of articulated objects using a view-based representation", *Lecture Notes in Computer Science*, vol. 1064, pp. 329, 1996.

[16] G. Dudek and C. Zhang, "Vision-based robot localization without explicit object models", in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996.

[17] Sageev Oore, Geoffrey Hinton, and Gregory Dudek, "A mobile robot that learns its place", *Neural Computation*, vol. 3, no. 9, pp. 683–699, April 1997.

[18] J.H. Elder and S. W. Zucker, "Computing contour closure", in *Proc. 4th European Conference on Computer Vision*, Cambridge, UK, 1996, vol. 2, pp. 399–412.

[19] Lee A. Iverson, *Toward discrete geometric models for early vision*, PhD thesis, McGill University, 1993.

[20] Lance Williams and David Jacobs, "Stochastic completion fields: A neural model of illusory contour shape and salience", in *International Conference on Computer Vision*, June 1995.

[21] Eric Bourque, Gregory Dudek, and Philippe Ciaravola, "Robotic sightseeing - a method for automatically creating virtual environments", in *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[22] Matthew Turk and Alex Pentland, "Face processing: Models for recognition", *Mobile Robotics IV*, Nov. 1989.

[23] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition", in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, June 1994, pp. 84–90, IEEE Press.

[24] Grace Wahba, "Convergence rates of 'thin plate' smoothing splines when the data are noisy", *Smoothing Techniques for Curve Estimation*, pp. 233–245, 1979.