Localization and Positioning using Combinations of Model Views

Elad Rivlin and Ronen Basri

Abstract

A method for localization and positioning in an indoor environment is presented. Localization is the act of recognizing the environment, and positioning is the act of computing the exact coordinates of a robot in the environment. The method is based on representing the scene as a set of 2D views and predicting the appearance of novel views by linear combinations of the model views. The method accurately approximates the appearance of scenes under weak perspective projection. Analysis of this projection as well as experimental results demonstrates that in many cases this approximation is sufficient to accurately describe the scene. When an orthographic approximation is invalid, either a larger number of models can be acquired or an iterative solution to account for the perspective distortions can be employed.

The presented method has several advantages over existing methods. It uses relatively simple representations, the representations are 2D rather than 3D, and localization can be done by a single 2D view only. The same principal method is applied both for the localization as well as the positioning problems, and a simple algorithm for repositioning, the task of returning to a previously visited position defined by a single view, is derived from this method.

©Massachusetts Institute of Technology (1992)

This report describes research done at the Massachusetts Institute of Technology with support from the McDonnell-Pew Center for Cognitive Neuroscience as well as support from the Defense Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-91-J-4038. Ronen Basri is supported by the McDonnell-Pew and the Rothschild postdoctoral fellowships. Elad Rivlin is at the University of Maryland, College Park, MD.
1 Introduction

Basic tasks in autonomous robot navigation are localization and positioning. Localization is the act of recognizing the environment, that is, assigning consistent labels to different locations. Positioning is the act of computing the coordinates of the robot in the environment. Positioning is a task complementary to localization, in the sense that position (e.g., “1.5 meters north of table T”) is often specified in a place-specific coordinate system (“in room 911”). In this paper we suggest a method of both localization and positioning using vision alone. A variant of the positioning problem referred to as repositioning, involving the return to a previously visited place is also discussed.

Previous studies have examined the problems of localization and positioning under a variety of conditions, defined by the kind of sensor(s) employed, the nature of the environment, the representations used. We can distinguish between active and passive sensing, indoor and outdoor navigation tasks, and metric and topological representations. The metric approach attempts to utilize a detailed geometric description of the environment, while the topological approach uses a more qualitative description including a graph with nodes representing places and arcs representing sequences of actions that would result in moving the robot from one node to another.

In the paper we consider a robot that uses a passive sensor, vision, in an indoor environment. The environment cannot be changed by the robot to improve its performance; neither beacon, nor floor or wall markings are employed. The paper addresses both the localization and the positioning problems. Solutions to these problems are presented based on object recognition techniques. The method, based on the linear combinations scheme of [17], represents scenes by sets of their 2D images. Localization is achieved by comparing the observed image and the linear combinations of model views, and the position of the robot is computed by analyzing the coefficients of the linear combination that aligns the model to the image. Also, a simple “qualitative” solution to the repositioning problem using the linear combinations scheme is presented.

The rest of the paper is organized as follows. The next section describes the localization and positioning problems and surveys previous solutions. The method of localization and positioning using linear combinations of model views is described in Section 3. The method assumes a perspective projection. An iterative scheme to account for perspective distortions is presented in Section 4. An analysis of the error resulting from the projection assumption is presented in Section 5. Constraints imposed on the motion of the robot as a result of special properties of indoor environments can be used to reduce the complexity of the method presented here. This topic is covered in Section 6. Experimental results follow.
2 The Problem

Localization and positioning from visual input are defined in the following way: Given a familiar environment, identify the observed environment, and then find your position in that environment. Localization resembles the task of object recognition, with objects replacing scenes. Once localization is accomplished, positioning can be performed.

One problem a system for localization and positioning should address is the variability of images due to viewpoint changes. The inexactness of practical systems makes it difficult for a robot to return to a specified position on subsequent visits. The visual data available to a robot between visits varies in accordance with the viewing position of the robot. A local positioning system should be able to recognize scenes from different positions and orientations.

Another problem is that of changes in the scene. At subsequent visits the same place may look different due to changes in the arrangement of the objects, the introduction of new objects, and the removal of others. In general, some objects tend to be more static than others. Wires, chairs and books are often moved, tables, closets, and pictures tend to change their position much less, and walls are almost guaranteed to be static. Static cues naturally are more reliable than mobile ones. Confining the system to static cues, however, may in some cases result in failure to recognize the scene due to insufficient cues. The system should therefore attempt to rely on static cues, but should not ignore the dynamic cues.

Solutions to the problem of localization from visual data require a large memory and heavy computation. Existing systems often try to reduce this cost by using sparse representations and by exploiting contextual information. Sparse representations are introduced in [1]. Moravec [10] represents scenes as sequences of landmarks (such as walls, doors, etc.) extracted by tracing the boundaries of the scene using a sonar and a compass. Metric information, and between the landmarks is not stored. Sarachik [14] recognizes a room by its dimensions, which are measured by identifying and locating the top corners of the room using stereo data (obtained from four cameras). In both cases the representation is very sparse, and the system therefore often ambiguous.

Hierarchical representations are used in [3, 5] where higher success rates are reported. Barrow [3] represents the scene by an occupancy table, a 2D bit array which contains a 1 at every location occupied by some object. The table is constructed by taking stereo pictures covering a 360° view of the middle of the room and projecting the obtained 3D data onto the floor. The method suffers from loss of information due to the projection onto the floor.

Engelson et al. [5] represent the scene by a set of invariant “signatures”. A signature is usually composed of low-resolution gray-level or range data obtained by blurring an image set of signatures taken from different viewpoints are stored. A scene is recognized if the robot encounters a signature similar to one of the stored signatures.

Systems that use the full information provided by the image (e.g., [6, 12]) usually rely on contextual information to avoid scanning all the models in the memory and to reduce the computational cost of comparing a model to the image. The system follows a predetermined
path, so that the identity of each visited location is known in advance, and localization by a verification problem. Path continuity in many cases is essential, and the so-called “drop-off problem is not addressed. The emphasis in these systems is on positioning, which is used to keep the robot on the path. It is typical for these systems (e.g., [1, 6, 12]) to use a model of the environment.

Oguchi et al. [12], among others, represent the environment by a set of landmarks selected from pairs of stereo images by a human operator. These landmarks are transformed by an image processing program which is designed so as to identify the specific landmark using specific extraction instructions (such as what features to look for and at what locations). Location is achieved by applying the extraction procedure specified for the next landmark. Once a landmark is identified, the position of the robot relative to that landmark is determined by comparing the dimensions of the observed landmark with those of the stored model.

The method presented in this paper represents the environment using a set of edge maps. Localization and positioning are achieved by comparing images of the environment to linear combinations of the model views. The method uses rich visual information to represent the scene. The system is flexible. In many cases it is capable of recognizing its location from one image only (360° coverage is not required). When one image is not sufficient, additional images can be acquired to solve the localization problem. Context can be used to determine the order of comparison of the models to the observed image and to increase the confidence in a given match, but context is not essential; the system can also, by performing more extensive computations, solve the “drop-off” problem.

3 The Method

The problems of localization and object recognition are similar in many ways. Both problems require the matching of visual images to stored models, either of the environment or of observed objects. Both problems face similar difficulties, such as varying illumination conditions and changes in appearance due to viewpoint changes. Similar methodologies therefore can be used for solving both problems.

A particular application of an object recognition scheme, the Linear Combinations (LC scheme [17], to the problems of localization and positioning is discussed below. The environment is represented in this scheme by a small set of views obtained from different viewpoints and the correspondence between the views. A novel viewpoint is recognized by comparing it to linear combinations of the stored views. Positioning is achieved by recovering the position of the camera relative to its position in the model views from the coefficients of the aligning combination. In the rest of this section we review the linear combinations approach and describe its application to both localization and positioning. The section concludes with a solution to the problem of repositioning, that is, the problem of returning to a previously visited position by “locking” into an image acquired in that position.
3.1 Localization

The problem of localization is defined as follows: given \( P \), a 2D image of a place, and \( \mathcal{M} \), a set of stored models, find a model \( M \) such that \( P \) matches \( \mathcal{M} \). Localization is the recognition of a place. It can therefore potentially benefit from using an object recognition method.

A common approach to handling the problem of recognition from different viewpoints is by comparing the stored models to the observed environment after the viewpoint is recovered or compensated for. This approach, called alignment, is used in a number of studies of object recognition [2, 7, 8, 9, 15, 16]. We apply the alignment approach to the problem of localization.

The system described below uses the “Linear Combinations” (LC) scheme, which was suggested by Ullman and Basri [17].

We begin with a brief review of the LC scheme. LCs are defined as follows. Given an image, we construct two view vectors from the feature points in the image, one contains the \( x \)-coordinate of the points, and the other contains the \( y \)-coordinate of the points. An object (in our environment) is modeled by a set of such views, where the points 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 object is recognized if the two match. The advantage of this method is twofold. First, viewer-centered representations are used rather than object-centered ones; namely, models are composed of 2D views of the observed scene; second, novel appearances are predicted in a simple and accurate way (under weak perspective projection).

Formally, given \( P \), a 2D image of a scene, and \( \mathcal{M} \), a set of stored models, the objective is to find a model \( M \in \mathcal{M} \) such that \( P = \sum_{j=1}^{k} a_j M_j \) for some constant \( a_j \in \mathcal{R} \). It has been shown that this scheme accurately predicts the appearance of rigid objects under weak perspective projection (orthographic projection and scale). The limitations of this projection model are discussed later in this paper.

More concretely, let \( \mathcal{R} = \{ (x_i, y_i, z_i) \}, 1 \leq i \leq n \), be a set of \( n \) object points. Under weak perspective projection, the points \( \{ (x'_i, y'_i) \} \) of these points in the image are given by

\[
\begin{align*}
x'_i &= s r_{11} x_i + s r_{12} y_i + s r_{13} z_i + t_x \\
y'_i &= s r_{21} x_i + s r_{22} y_i + s r_{23} z_i + t_y
\end{align*}
\]

(1)

where \( r_{ij} \) are the components of a \( 3 \times 3 \) rotation matrix, and \( s \) is a scale factor. Rewriting the \( x \)-vector equation forms we obtain

\[
\begin{align*}
x' &= s r_{11} x + s r_{12} y + s r_{13} z + t_x 1 \\
y' &= s r_{21} x + s r_{22} y + s r_{23} z + t_y 1
\end{align*}
\]

(2)

where \( x, y, z, x', y' \in \mathcal{R} \) are the vectors \( \{ x_i, y_i, z_i \}, \), \( i \) and \( y' \) coordinates respectively, and \( 1 = (1, 1, \ldots, 1) \). Consequently,

\[
x', y' \in \text{span}\{x, y, z, 1\}
\]

(3)
or, in other words, and \( y' \) belong to a four-dimensional linear subspace (Δν of course that \( z' \), the vector of depth coordinates of the projected points, also belongs to this subspace: this fact is used in Section 4 below.) A four-dimensional space is spanned by any four linearly independent vectors of the space. Two views of the scene supply four such vectors [13, Δν]. Denote by \( x_1, y_1 \) and \( x_2, y_2 \) the location vectors of the \( n \) points in the two images; then there exist coefficients \( a_1, a_2, a_3, b_1, b_2, b_3, b_4 \) such that

\[
\begin{align*}
x' &= a_1 x_1 + a_2 y_1 + a_3 x_2 + a_4 1 \\
y' &= b_1 x_1 + b_2 y_1 + b_3 x_2 + b_4 1
\end{align*}
\]

(Note that the vector \( x' \) already depends on the other four vectors.) Since \( R \) is a rotation matrix, the coefficients satisfy the following two quadratic constraints:

\[
\begin{align*}
a_1^2 + a_2^2 + a_3^2 - b_1^2 - b_2^2 &= 2(b_1 b_3 - a_1 a_3)\gamma_1 + 2(b_2 b_3 - a_1 a_2)\gamma_2 \\
a_1 b_1 + a_2 b_2 + a_3 b_3 + (a_1 b_3 + a_2 b_1)\gamma_1 + (a_2 b_3 + a_3 b_1)\gamma_2 &= 0
\end{align*}
\]

To derive these constraints the transformation between the two model views should be recover. 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 generally scenes are fairly from one another.

A LC scheme for the problem of localization is as follows: The environment is modeled by a set of images with correspondence between the images. For example, a spot 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 the 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 LC scheme uses viewer-centered models, that is, 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 areas of the scene, we avoid the need to handle occlusions in the scene. If from some viewpoint 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 models. 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.

One problem with using the LC scheme for localization is due to the weak perspective approximation. In contrast with the problem of object recognition, where we can generally as
that objects are small relative to their distance from the camera, in localization the environment surrounds the robot and perspective distortions cannot be neglected. The limitations of weak perspective modeling are discussed both mathematically and empirically in the next two sections. It is shown that in many practical cases weak perspective is sufficient to ensure accurate localization. The main reason is that the problem of localization does not require accurate measurements in the entire image; it only requires identifying a sufficient number of spots to guarantee accurate naming. If these spots are relatively close to the center of the image, or if the depth differences they create are relatively small (as in the case of a wall when the line of sight is nearly perpendicular to the wall), the perspective distortions are relatively small, and the system can identify the scene with high accuracy. Also, it relates to a translation parallel to the image plane for a linear space even when perspective distortions are large. This case and other simplifications are discussed in Section 6.

By using weak perspective we avoid stability problems that frequently occur in perspective computations. We can therefore compute the alignment coefficients by looking at a relatively narrow field of view. The entire scene can be viewed as an accumulative process. Rather than acquiring images of the entire scene and comparing them all to a full scene model (as in [2]), we recognize the scene image by image, spot by spot, until we accumulate sufficient convincing information that indicates the identity of the place.

When perspective distortions are relatively large and weak perspective is insufficient, we model the environment, two approaches can be used. One possibility is to construct a larger number of models so as to keep the possible changes between the familiar and the novel views small. Alternatively, an iterative computation can be applied to compensate for these distortions. Such an iterative method is described in Section 4.

### 3.2 Positioning

Positioning is the problem of recovering the exact position of the robot. This position is 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 discuss an application of the LC scheme to the positioning problem.

The idea is the following. We assume a model composed of two images $P_1$ and $P_2$; their relative position is given. Given a novel image, we first align the model with the image (i.e., localization). By considering the coefficients of the linear combination of the robot relative to the model images is recovered. To recover the absolute position of the robot, we use the absolute positions of the model views. The absolute positions of the model points in the following way:

$$x' = a_1 x_1 + a_2 y_1 + a_3 x_2 + a_4$$
\[ y' = b_1 x_1 + a_2 y_1 + b_3 x_2 + b_4 \]  

Substituting for we obtain

\[
\begin{align*}
x' &= a_1 x_1 + a_2 y_1 + a_3 (s \eta_1 x_1 + s \eta_2 y_1 + s r_{12} z_1 + t_x) + a_4 \\
y' &= b_1 x_1 + b_2 y_1 + b_3 (s \eta_1 x_1 + s \eta_2 y_1 + s r_{12} z_1 + t_x) + b_4
\end{align*}
\]  

and rearranging these equations we obtain

\[
\begin{align*}
x' &= (a_1 + a_3 s \eta_1) x_1 + (a_2 + a_3 s \eta_2) y_1 + (a_3 s \eta_3) z_1 + (a_3 t_x + a_4) \\
y' &= (b_1 + b_3 s \eta_1) x_1 + (b_2 + b_3 s \eta_2) y_1 + (b_3 s \eta_3) z_1 + (b_3 t_x + a_4)
\end{align*}
\]  

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 \( v \), and scaling \( s \).

Using the orthonormality constraint we can first derive the scale factor

\[
s_n^2 = (a_1 + a_3 s \eta_1)^2 + (a_2 + a_3 s \eta_2)^2 + (a_3 s \eta_3)^2 = a_1^2 + a_2^2 + a_3^2 s^2 + 2 a_3 s (a_1 \eta_1 + a_2 \eta_2)
\]  

From Equations (8) and (9), by deriving the components of the translation vector, \( t \), we obtain the position of the robot in the image relative to its position in the model view.

\[
\begin{align*}
\Delta x &= a_3 t_x + a_4 \\
\Delta y &= b_3 t_y + b_4 \\
\Delta z &= f \left( \frac{1}{s_n} - \frac{1}{s} \right)
\end{align*}
\]

Note that \( \Delta z \) is derived from the change in scale of the object. The rotation matrix \( U \) between \( P_1 \) and \( P_2 \) is given by

\[
\begin{align*}
u_{11} &= \frac{a_1 + a_3 \eta_1}{s_n} & u_{12} &= \frac{a_2 + a_3 \eta_2}{s_n} & u_{13} &= \frac{a_3 \eta_3}{s_n} \\
u_{21} &= \frac{b_1 + b_3 \eta_1}{s_n} & u_{22} &= \frac{b_2 + b_3 \eta_2}{s_n} & u_{23} &= \frac{b_3 \eta_3}{s_n}
\end{align*}
\]

As was already mentioned, the position of the robot is computed here relative to the position in the camera when the first model image is acquired. \( \Delta x \) and \( \Delta z \) represent the motion of the robot from \( P \) to \( P_2 \), and the rest of the parameters represent its 3D rotation and elevation.

To obtain the relative position the transformation parameters between the model views, \( P_1 \) and \( P_2 \), are required.

### 3.3 Repositioning

An interesting variant of the positioning problem referred to as repositioning, is defined as follows. Given an image, called the target image, position yourself in the location from
The image was observed to solve this problem. To extract the exact position from which the target image was obtained and direct the robot to that position. In this section, we are interested in a more qualitative approach. Under this approach position is not computed. Instead, the robot observes the environment and extracts only the direction to the target location. Unlike the exact approach, the method presented here does not require the recording of the transformation between the model views.

We assume we are given with a model of the environment together with a target image. The robot is allowed to take new images as it moves towards the target. We assume a horizontally moving platform (In other words, we assume three degrees of freedom rather than six; the robot is allowed to rotate around the vertical axis and translate horizontally. The validity of this constraint is discussed in Section 6.) Below we give a simple computation that determines a path which terminates in the target location. At each time step the robot acquires a new image and aligns it with the model. By comparing the alignment coefficients with the coefficients for the target image the robot determines its next step. The algorithm is divided into two stages. In the first stage the robot fixates on one identifiable point and moves along a circular path around the fixation point until the line of sight to this point coincides with the line of sight to the corresponding point in the target image. In the second stage the robot advances forward or retreats backward until it reaches the target location.

**Given a model composed of two images,** \( P_1 \) and \( P_2 \), \( P_1 \) is obtained from \( P_2 \) by a rotation about the Y-axis by an angle \( \alpha \), horizontal translation factor \( s \). Given a target image \( P_t \), \( P_t \) is obtained from \( P_2 \) by a similar rotation by an angle \( \theta \), translation factor \( s_t \). Using Eq. (4) the position of a target point \( P_t \) can be expressed as

\[ x_t = a_1 x_1 + a_3 x_2 + a_4 \\
\]

\[ y_t = b_2 y_1 \]

(12)

(The rest of the coefficients are zero since the platform moves horizontally.) In fact, the coefficients are given by

\[ a_1 = \frac{s_t \sin(\alpha - \theta)}{\sin \alpha} \]

\[ a_3 = \frac{s_t \sin \theta}{\sin \alpha} \]

\[ a_4 = t \cdot \frac{s_t \sin \theta}{s \sin \alpha} \]

\[ b_2 = s_t \]

(13)

(\text{The derivation is given in the Appendix.})

At every time step the robot acquires an image and aligns it with the above model. Assume that image \( P_i \) is obtained as a result of a rotation by an angle \( \phi \), translation \( s \).
The position of a point \( p_i(x) \) is expressed by
\[
\begin{align*}
x_p &= c_1 x_1 + c_3 x_2 + c_4 \\
y_p &= d_2 y_1
\end{align*}
\] (14)

where the coefficients are given by
\[
\begin{align*}
c_1 &= \frac{s_p \sin(\alpha - \phi)}{\sin \alpha} \\
c_3 &= \frac{s_p \sin \phi}{s \sin \alpha} \\
c_4 &= \frac{t_p - t_p s_p \sin \phi}{s \sin \alpha} \\
d_2 &= s_p
\end{align*}
\] (15)

The step performed by the robot is determined by
\[
\delta = \frac{c_1}{c_3} - \frac{a_1}{a_3}
\] (16)

That is,
\[
\delta = \frac{s \sin(\alpha - \phi)}{s \sin \phi} \frac{s \sin(\alpha - \theta)}{s \sin \theta} = s \sin \alpha (\cot \phi - \cot \theta)
\] (17)

The robot should now move so as to reduce the absolute value of \( \delta \). The direction of motion depends on the sign of \( \omega \). The robot can deduce the direction by moving slightly to the side and checking if this motion results in an increase or decrease of \( \delta \). The notion is defined as follows. The robot moves to the right (or to the left), depending on which direction reduces \( \delta \) by a step \( \Delta \).

A new image \( P_n \) is now acquired, and the fixated point is located in this image. Denote its new position by \( x \). Since the motion is parallel to the image plane the depth values of the point in the two views, \( a_n \) and \( P_n \), are identical. We now want to rotate the camera so as to return the fixated point to its original position. The angle of rotation, \( \beta \), can be deduced from the equation
\[
x_p = x_n \cos \beta + \sin \beta
\] (18)

This equation has two solutions. We choose the one that counters the translation (namely, translation is to the right, the camera should rotate to the left), and that keeps the angle of rotation small. In the next step, the new position \( P_{n+1} \) and the procedure is repeated until \( \delta \) vanishes. The resulting path is circular around the point of focus.

Once the robot arrives at a position for which \( \delta = 0 \) (namely, its line of sight coincides with that of the target image, and \( \phi = \theta \)) it should now advance forward or retreat backward to adjust its position along the line of sight. Several measures can be used to determine the direction of motion; one example is the
\[
\frac{c_1}{a_1} = \frac{s_p}{s_t}
\] (19)

when the two lines of sight coincide. The objective at this stage is to bring this measure
4 Handling Perspective Distortions

The linear combination scheme presented above accurately handles changes in viewpoint assuming the images are obtained under weak perspective projection. Error analysis and experimental results demonstrate that in many practical cases this assumption is valid. In cases where perspective distortions are too large to be handled by a weak perspective approximation, matching between the model and the image can be facilitated in two ways. One possibility is to avoid cases of large perspective distortion by augmenting the library of stored models with additional models. In a relatively dense library there usually exists a model that is related to the view by a sufficiently small transformation avoiding such distortions. The second alternative improves the match between the model and the image using an iterative process. In this section we consider the second option.

The suggested iterative process is based on a Taylor expansion of the perspective coordinates. As described below, this expansion results in a polynomial consisting of terms of which can be approximated by linear combinations of views. The first term of this series represents the orthographic approximation. The process resembles a method of matching points with 2D points described recently by Dementhon and Davis [4]. In this case, however, the method is applied to 2D models rather than 3D ones. In our application the 3D coordinates of the model points are not provided; instead they are approximated from the model views.

An image point $(x, y) = (fX/Z, fY/Z)$ is the projection of some object point $(X, Y, Z)$ in the image, where $f$ denotes the focal length. Consider the following Taylor expansion of $x$ around some depth value $Z_0$:

$$
\frac{1}{Z} = \sum_{k=0}^{\infty} \frac{f^{(k)}(Z_0)}{k!} (Z - Z_0)^k
$$

$$
= \frac{1}{Z_0} + \sum_{k=1}^{\infty} \frac{(-1)^k}{(k - 1)!} Z_0^{k+1}(Z - Z_0)^k
$$

The Taylor series describing the position of a point $x$ is therefore given by

$$
x = \frac{fX}{Z} = \frac{fX}{Z_0} \left[1 + \sum_{k=1}^{\infty} \frac{(-1)^k}{(k - 1)!} \left(\frac{Z - Z_0}{Z_0}\right)^k\right]
$$

The $k$th term of the series:

$$
\Delta^{(k)} = \frac{fX}{Z_0} \frac{(-1)^k}{(k - 1)!} \left(\frac{Z - Z_0}{Z_0}\right)^k
$$

Notice that the zero term contains the orthographic approximation for $x$. The $k$th term of the series:

A recursive definition of the above series is given below.
Initialization
\[ x^{(0)} = \Delta^{(0)} = \frac{f X}{Z_0} \]

Iterative step:
\[ \Delta^{(k)} = -\frac{Z - Z}{(k - 1)Z} \Delta^{(k-1)} \]
\[ x^{(k)} = x^{(k-1)} + \Delta^{(k)} \]

where \( x^{(k)} \) represents the \( k \)th order approximation for \( x \), and \( \Delta^{(k)} \) represents the highest order term in \( \Delta^{(k)} \).

According to the orthographic approximation both \( X \) and \( Z \) can be expressed as linear combinations of the model views (Eq. (4)). We therefore apply the above procedure, approximating \( X \) and \( Z \) at every step using the linear combination that best aligns the model points with image points. The general idea is therefore the following. First, \( x \) and \( \Delta^{(k)} \) are obtained by solving the orthographic case. Then at each step of the iteration we improve the estimate of \( x \) by seeking the linear combination that best estimates the factor
\[ \frac{Z - Z}{(k - 1)Z} \approx \frac{x - \Delta^{(k)}}{\Delta^{(k-1)}} \]
(23)

Denote by \( x \in \mathcal{R} \) the vector of image point coordinates, and denote by
\[ P = [x_1, x_2, x_3, 1] \]

an \( n \times 4 \) matrix containing the position of the points in the two model images. Denote \( P^+ = (P^TP)^{-1}P^T \) the pseudo-inverse of \( P \) (we assume \( P \) is overdetermined). Also denote by \( a^{(k)} \) the coefficients computed for the \( k \)th step. \( a^{(k)} \) represents the linear combination computed at that step to approximate the \( X \) or the \( Z \) values. Since at every step \( k \) are constant they can be merged into the linear combination. Denote by \( \Delta^{(k)} \) the vectors of computed values of \( x \) and \( \Delta \) at the \( k \)th step. An iterative procedure to align a model to the image is described below.

Initialization
Solve the orthographic approximation, namely
\[ a^{(0)} = P^+ x \]
\[ x^{(0)} = \Delta^{(0)} = Pa^{(0)} \]

Iterative step
\[ q^{(k)} = (x - x^{(k-1)}) + \Delta^{(k-1)} \]
\[ a^{(k)} = Pq^{(k)} \]
\[ \Delta^{(k)} = (Pa^{(k)}) \oplus \Delta^{(k-1)} \]
\[ x^{(k)} = x^{(k-1)} + \Delta^{(k)} \]
where the vector operations $\otimes$ and $\div$ are defined as
\[
\mathbf{u} \otimes \mathbf{v} = (u_1 v_1, \ldots, u_n v_n)
\]
\[
\mathbf{u} \div \mathbf{v} = \left( \frac{u_1}{v_1}, \ldots, \frac{u_n}{v_n} \right)
\]

5 Projection Model – Error Analysis

In this section we estimate the error obtained by using the linear combination method. The method assumes a weak perspective projection model. We compare this assumption with the more accurate perspective projection model.

A point \((X, Y, Z)\) is projected under the perspective model to \((x, y) = (fX/Z, fY/Z)\) in the image, where \(f\) denotes the focal length. Under our weak perspective model the same point is approximated by \((\hat{x}, \hat{y}) = (sX, sY)\) where \(s\) is a scaling factor. The best estimate for the scaling factor, is given by \(s_0 = \frac{1}{Z_0}\) where \(Z_0\) is the average depth of the observed environment. Denote the error by

\[
E = |\hat{x} - x|
\]

The error is expressed by

\[
E = \left| fX(\frac{1}{Z_0} - \frac{1}{Z}) \right|
\]

Changing to image coordinates

\[
E = \left| xZ(\frac{1}{Z_0} - \frac{1}{Z}) \right|
\]

or

\[
E = \left| x \right| \left| \frac{Z}{Z_0} - 1 \right|
\]

The error is small when the measured feature is close to the optical axis, or when our estimate for the depth, \(Z_0\), is close to the real depth, \(Z\). This supports the basic intuition that images with low depth variance and for fixed regions in the image, the obtained perspective distortions are relatively small, and the system can therefore recover the scene with high accuracy. Figures 1 and 2 show the depth ratio \(Z/Z_0\) as a function of \(x\) for \(\epsilon = 10\) and 20 pixels, and Table 5 shows a number of examples for this function. The allowed depth variance, \(Z/Z_0\), is computed as a function of \(x\) and the tolerated error, \(\epsilon\). For example, a 10 pixel error tolerated in a field of view of up to \(\pm 50\) pixel is equivalent to allowing variations of \(20\%\). From this discussion it is apparent that when a model is aligned to the scene, the results of this alignment should be judged differently at different points of the image. Farther away a point is from the center the more discrepancy should be tolerated between prediction and the actual image. A five pixel error at position \(x = 50\) is equivalent to a 10 pixel error at position \(x = 100\).

So far we have considered the discrepancies between the weak perspective and the perspective projections of points. The accuracy of the LCS scheme depends on the validity of the values...
Figure 1: $\frac{x}{x_0}$ as a function of $x$ for $\epsilon = 10$ pixels.

Figure 2: $\frac{x}{x_0}$ as a function of $x$ for $\epsilon = 20$ pixels.
<table>
<thead>
<tr>
<th>$x \setminus \epsilon$</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1.2</td>
<td>1.4</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>50</td>
<td>1.1</td>
<td>1.2</td>
<td>1.3</td>
<td>1.4</td>
</tr>
<tr>
<td>75</td>
<td>1.07</td>
<td>1.13</td>
<td>1.2</td>
<td>1.27</td>
</tr>
<tr>
<td>100</td>
<td>1.05</td>
<td>1.1</td>
<td>1.15</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 1: Allowed depth ratios $\hat{Z}_0$, as a function of $x$ (half the width of the field considered) and the error allowed ($\epsilon$, in pixels).

perspective projection both in the model views and for the incoming image. In the rest of this section we develop an error term for the LC scheme assuming that both the model views and the incoming image are obtained by perspective projection.

The error obtained by using the LC scheme is given by

$$E = | x - a \hat{x} - b \hat{y} - c \hat{z} - \hat{d} |$$  \hspace{1cm} (29)

Since the scheme accurately predicts the appearances of points under weak perspective projection, it satisfies

$$\hat{x} = a \hat{x}_1 - b \hat{\psi} - c \hat{z} - d$$  \hspace{1cm} (30)

where accented letters represent orthographic approximations. Assume that in the two model pictures the depth ratios are roughly equal:

$$\frac{Z_0^M}{Z^M} = \frac{Z_{01}}{Z_1} = \frac{Z_{02}}{Z_2}$$  \hspace{1cm} (31)

(Th is condition is satisfied, for example, when between the two model images the camera translates along the image plane.) Using the fact that

$$x = \frac{f X}{Z} = \frac{f X Z_0}{Z_0} = \frac{\hat{x} Z_0}{Z}$$  \hspace{1cm} (32)

we obtain

$$E = | x - a \hat{x} - b \hat{y} - c \hat{z} - \hat{d} |$$

$$\approx | \frac{Z_0}{Z} \hat{x} - a \hat{\psi} \frac{Z_0^M}{Z^M} - b \hat{y} \frac{Z_0^M}{Z^M} - c \hat{z} \frac{Z_0^M}{Z^M} - \hat{d} |$$

$$= | \frac{Z_0}{Z} \hat{x} - ( a \hat{\psi} - b \hat{\psi} - c \hat{z} ) \frac{Z_0^M}{Z^M} - \hat{d} |$$

$$= | \frac{Z_0}{Z} \hat{x} - \frac{Z_0^M}{Z^M} - \hat{d} |$$  \hspace{1cm} (33)
\[
= \left| z \left( \frac{Z_0}{Z} - \frac{Z_0^M}{Z^M} \right) - \frac{Z_0^M}{Z^M} - 1 \right|
\]
\[
\leq |\hat{z}| \left| \frac{Z_0}{Z} - \frac{Z_0^M}{Z^M} \right| + |d| \left| \frac{Z_0^M}{Z^M} - 1 \right|
\]

The error therefore depends on two terms. The first gets smaller as the image points get close to the center of the frame and as the difference between the depth ratios of the model and the image gets smaller. The second gets smaller as the translation component gets smaller and the model gets close to orthographic.

Following this analysis, weak perspective can be used as a projection model when the depth variations in the scene are relatively low and when the system concentrates on the center of the image. We conclude that, by fixing on distinguished parts of the environment, a linear combinations scheme can be used for localization and positioning.

6 Imposing Constraints

Localization and positioning require a large memory and a great deal of on-line computation. A large number of models must be stored to enable the robot to navigate and manipulate in relatively large and complicated environments. The computational cost of model-in-comparison is high, and if context (such as path history) is not available the number of recoimparisons may get very large. To reduce this computational cost a number of constraints may be employed. These constraints take advantage of the structure of the robot, the properties of indoor environments, and the natural properties of the navigation task. This section examines some of these constraints.

One thing a system may attempt to do is to build the set of models so as to reduce the effect of perspective distortions in order to avoid performing iterative computations. When the environment obtained when the system looks relatively deep into the scene usually satisfies this condition. When perspective distortions are large the system may consider more subsets of views related by a translation parallel to the image plane (perpendicular to the line of sight). In this case the depth values of the points are roughly equal across all subsets considered, and it can be shown that novel views can be expressed by linear combinations of two model views even in the presence of large perspective distortions. This becomes apparent from the following derivation. Let \( \mathbf{X} = Z_i \), \( 1 \leq i \leq n \) be a point projected in the image to \((x, y) = (f X_i/Z_i, f Y_i/Z_i)\), and let \( f_{ij}x'_j \) be the projected point after applying a rigid transformation. Assuming the condition \( Z_i \), we obtain

\[
Z_i x'_i = r_{11} X_i + r_{12} Y_i + r_{13} Z_i + t_x
\]
\[
Z_i y'_i = r_{21} X_i + r_{22} Y_i + r_{23} Z_i + t_y
\]

(34)

where

\[
x'_i = r_{11} x_i + r_{12} y_i + r_{13} + t_x \frac{1}{Z_i}
\]

(35)
\[
\begin{align*}
  y'_i &= r_1 x_i + r_2 y_i + r_3 + t_y z_i \\
  y'_i &= r_1 x_i + r_2 y_i + r_3 + t_y z_i
\end{align*}
\] (35)

Rewriting this in vector equation form gives

\[
\begin{align*}
  \mathbf{x}' &= r_{11} \mathbf{x} + r_{12} \mathbf{y} + r_{13} \mathbf{1} + t_x \mathbf{z} \\
  \mathbf{y}' &= r_{21} \mathbf{x} + r_{22} \mathbf{y} + r_{23} \mathbf{1} + t_y \mathbf{z} \quad (36)
\end{align*}
\]

where \( \mathbf{x}, \mathbf{y}, \mathbf{z} \) and \( \mathbf{y}' \) are the vectors of \( x, y, z \), and \( \mathbf{z} \) is a vector of all 1's, and \( \mathbf{1} \) is a vector of \( 1 \)'s values respectively. \( \mathbf{1} \) is a vector of all 1's, and \( \mathbf{z} \) is a vector of \( 1 \)'s values. Consequently, as in the weak perspective case, novel views obtained by a translation parallel to the image plane can be expressed by linear combinations of four vectors.

An indoor environment usually provides the robot with a flat, horizontal support. Consequently, the motion of the camera is often constrained to rotation about the vertical \((Y)\) axis and to translation in the \(XZ\) plane. Such motion has only three degrees of freedom in the general case. Under this constraint fewer correspondences are required to align the model with the image. For example, in Eq. (4) (above) the coefficient \( a_2 = b_1 = b_3 = b_4 = 0 \). Three points rather than four are required to determine the coefficients of \( \mathbf{x}' \) and \( \mathbf{y}' \) by solving a linear system. Two, rather than three, are required if the quadratic constraints are also considered. Another advantage of considering only horizontal motion is the fact that it constrains the possible epipolar lines between images. This fact can be used to reduce the task of correspondence seeking.

Objects in indoor environments sometimes appear in roughly planar settings. In particular, objects placed on the floor, such as shelves, pictures, closets and tables, are generally flat. If the assumption of orthographic projection is made (for example, when the floor is relatively distant from the robot, or when the line of sight is roughly perpendicular to the wall) the transformation between any two views can be described by a 2D affine transformation. The dimension of the space of views of the scene is then reduced to three (rather than four), and Eq. (4) becomes

\[
\begin{align*}
  \mathbf{x}' &= a_1 \mathbf{x} + a_2 \mathbf{y} + a_3 \mathbf{1} \\
  \mathbf{y}' &= b_1 \mathbf{x} + b_2 \mathbf{y} + b_3 \mathbf{1}
\end{align*}
\] (37)

\( a_1 = a_2 = a_3 = 0 \). Only one view is therefore sufficient to model the scene.

Most office-like indoor environments are composed of rooms connected by corridors. Navigating in such an environment involves maneuvering through the corridors, entering and exiting the rooms. Not all points in such an environment are equally important. Junctions, places where the robot faces a number of options for changing its direction, are more important than other places for navigation. In an indoor environment these places include the thresholds of rooms and the beginnings and ends of corridors. A navigation system would therefore tend to store more models for these points than for others.

One important property shared by many junctions is that they are confined to relatively small areas. Consider for example the threshold of a room. It is a relatively narrow
that separates the room from the adjacent corridor. When a robot is about to enter a room, a common behavior includes stepping through the door, looking into the room, and identifying it before a decision is made to enter the room or avoid it. The set of interesting images for this task includes the set of views of the room from its entrance. Provided that thresholds are set where these views are related to each other almost exclusively by rotation around the vertical axis. Under perspective projection, such a rotation is relatively easy to recover. The set of points in novel views can be recovered from one model view only. This is apparent from the following derivation. Consider a point \( p = (X, Y, Z) \). Its position in a model view is given by \((x, y) = (fX/Z, fY/Z)\). Now, consider another view obtained by a rotation \( R \) around the camera. The location of \( p \) in the new view is given by (assuming \( f = 1 \))

\[
(d', \ y') = \left( \frac{r_{11}X + r_{12}Y + r_{13}Z}{r_{31}X + r_{32}Y + r_{33}Z}, \frac{r_{21}X + r_{22}Y + r_{23}Z}{r_{31}X + r_{32}Y + r_{33}Z} \right)
\]

(38)

implying that

\[
(d', \ y') = \left( \frac{r_{11}x + r_{12}y + r_{13}}{r_{31}x + r_{32}y + r_{33}}, \frac{r_{21}x + r_{22}y + r_{23}}{r_{31}x + r_{32}y + r_{33}} \right)
\]

(39)

Depth is therefore not a factor in determining the relation between the views. Eq. (39) becomes even simpler if only rotations about the \( Y \)-axis are considered:

\[
(d', \ y') = \left( \frac{x \cos \alpha + \sin \alpha}{-x \sin \alpha + \cos \alpha} \frac{y}{x \sin \alpha + \cos \alpha} \right)
\]

(40)

where \( \alpha \) is the angle of rotation. In this case \( \alpha \) can be recovered merely from a single correspondence.

7 Experiments

The ILC method was implemented and applied to images taken in an indoor environment. Images of two offices, A and B that have similar structures were taken using a Panasonic camera with a focal length of 700 pixels. Semi-static objects, such as heavy furniture and pictures, were used to distinguish between the offices. Figure 3 shows two model views of office A. The views were taken at a distance of about 4m from the wall. Correspondences were picked manually. The results of aligning the model views to images of the two offices are presented in Figure 4. The left image contains an overlay of a predicted image (the thick white lines), constructed linearly combining the two views, and an actual image of office A. A good match between the two was achieved. The right image contains an overlay of a predicted image constructed from a model of office B and an image of office A. Because the offices share a similar structure the static cues (the wall corners) were perfectly aligned. The semi-static cues, however, still match any features in the image.

Figure 5 shows the matching of the model of office A with an image of the same office obtained by a relatively large motion forward (about 2m) and to the side (about 1.5m). Although...
Figure 3: Two model views of office A

Figure 4: Matching a model of office A to an image of office A (left), and matching a model of office B to the same image (right).

Figure 5: Matching a model of office A to an image of the same office obtained by a relatively large motion forward and to the right.
Figure 6: Two model views of a corridor.

Figure 7: Matching the corridor model with two images of the corridor. The right image was obtained by a relatively large motion forward (about half of the corridor length) and the right.

The distances are relatively short most perspective distortions are negligible, and a good match between the model and the image is obtained.

Another set of images was taken in a corridor. Here, because of the deep structure of the corridor, perspective distortions are noticeable. Nevertheless, the alignment reasonably well demonstrates an accurate match in large portions of the image. Figure 6 shows two model views of the corridor. Figure 7 (left) shows an overlay of a linear combination of the model with an image of the corridor. It can be seen that the parts that are relatively distant align relatively well. Figure 7 (right) shows the matching of the corridor model with an image obtained with a relatively large motion (about half of the corridor length). Because of perspective distortion, the relatively near features no longer align (e.g., the near door edges). The relatively far features, however, still match.

The next experiment shows the application of the iterative process presented in Section 5.
in cases where large perspective distortion were noticeable. Figure 8 shows two model views and Figure 9 shows the results of matching a linear combination of the model views to an image of the same office. In this case, because the image was taken from a relatively close distance, perspective distortions cannot be neglected. The effects of perspective distortion be noticed on the right corner of the board, and on the edges of the hanger on the top right. Perspective effects were reduced by using the iterative process. The results of applying this procedure after one and three iterations are shown in Figure 10.

The experimental results demonstrate that the LCM method achieves accurate localization in many cases, and that when the method fails because of large perspective distortions an iterative computation can be used to improve the quality of the match.

8 Conclusions

A method of localization and positioning in an indoor environment was presented. The method is based on representing the scene as a set of 2D views and predicting the appearance of the views by linear combinations of the model views. The method accurately approximates the appearances of scenes under weak perspective projection. Analysis of this projection as experimental results demonstrate that in many cases this approximation is sufficient to accurately describe the scene. When the weak perspective approximation is invalid, either a larger number of models can be acquired or an iterative solution can be employed to account for the perspective distortions.

The method presented in this paper has several advantages over existing methods. It uses remarkably rich representations; the representations are 2D rather than 3D, and localization can be done from a single 2D view only. The same basic method is used in both the localization and positioning problem, and a simple algorithm for repositioning is derived from this method. Future work includes handling the problem of acquisition and maintenance of models, developing efficient and robust algorithms for solving the correspondence problem, and building models using visual input.

Appendix

In this appendix we derive the explicit values of the coefficients of the linear combinations in case of horizontal motion. Consider a point \( p = (x, y, z) \) that is projected by weak perspective to three images, \( P_1 \), \( P_2 \), and \( P_3 \). \( P_1 \) is obtained from a rotation about the \( Y \)-axis by an angle \( \alpha \), translation \( t \), and scale factor \( s \). \( P_2 \) is obtained from a rotation about the \( Y \)-axis by an angle \( \theta \), translation \( t \), and scale factor \( s \). The position of \( p \) in the three images is given by

\[
(x_1, y_1) = (x, y)
\]
Figure 8: Two model views of office C.

Figure 9: Matching the model to an image obtained by a relatively large motion. Perspective distortions can be seen in the table, the board, and the hanger at the upper right.

Figure 10: The results of applying the iterative process to reduce perspective distortions, one (left) and three (right) iterations.
\[
(x_2', y) = (s_m x \cos \alpha + s_m z \sin \alpha + l_m, \ s_n y)
\]
\[
(x', y') = (s_p x \cos \theta + s_p z \sin \theta + t_p, \ s_p y)
\]

The point \((x', y')\) can be expressed by a linear combination of the first two points:

\[
x' = a_1 x_1 + a_2 x_2 + a_3
\]
\[
y' = b y_1
\]

Rewriting these equations we get

\[
s_p x \cos \theta + s_p z \sin \theta + t_p = a_1 x + a_2 (s_m x \cos \alpha + s_m z \sin \alpha + l_m) + a_3
\]
\[
s_p y = b y
\]

Equating the values for the coefficients in both sides of these equations we obtain

\[
s_p \cos \theta = a + a_2 s_m \cos \alpha
\]
\[
s_p \sin \theta = a_2 s_m \sin \alpha
\]
\[
t_p = a_2 t_m + a_3
\]
\[
s_p = b
\]

and the coefficients are therefore given by

\[
a_1 = \frac{s_p \sin (\alpha - \theta)}{\sin \alpha}
\]
\[
a_2 = \frac{s_p \sin \theta}{s_m \sin \alpha}
\]
\[
a_3 = t_p - \frac{t_m s_p \sin \theta}{s_m \sin \alpha}
\]
\[
b = s_p
\]

References


