2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

Two-Stage Visual Localisation: Landmark-Based Pose Initialisation and Model-Based Pose Refinement *

Zezhi Chen, Philip Pe, John McDermid and Nick Pears
Department of Computer Science, University of York, YO10 5DD, UK
(chen,philippe,jam,nep)@cs.york.ac.uk

***Abstract*— We show that landmark based localisation (LBL) and Lowe's model-based localisation (MBL) are complementary in that LBL provides a pose initialisation to MBL, which is a necessary input to the algorithm, and MBL can then refine that pose to a give more accurate pose estimate than LBL alone can provide. For LBL, we extend Betke and Gurvit's method, such that it can be used with standard perspective cameras (their original proposal was for omnidirectional cameras) in order to get a useful initial value as an input to Lowe's method. Intensive experiments have been carried out to analyse how camera parameters (intrinsic and extrinsic) affect the LBL position and orientation errors in the initial pose estimate. In error propagation experiments, we show that the position and orientation of a robot are sensitive to focal length and errors in imaged feature positions respectively. In the MBL pose refinement phase, we find that MBL is able to refine the position estimate, but the error in orientation estimate remains the same.**

***Index Terms*— Mobile robots, localization, visual landmarks, navigation.**

I. Introduction

Localization has attracted much attention in the Robotics and Computer Vision communities recently. Se and Lowe et al. [1][2][3] have proposed a global robot localization and map-building method using scale invariant visual landmarks. They have later refined this method using distinctive visual features [4]. The museum tour guide robot RHINO [6][5] utilizes a metric version of the Markov localization algorithm employing laser sensors with success. However, it needs to be supplied with a manually derived map, and cannot learn maps. Fox et al. [8][7] proposed the Monte Carlo Localization method based on CONDENSATION algorithm [9]. Their algorithm relies on vision-based Bayesian filtering methods using sampling density to represent multi-modal probability distributions. Given a visual map of the ceiling obtained by mosaicing, localization can be achieved using scalar brightness measurement. Betke and Gurvits gave an efficient method for localizing a mobile robot using landmarks [10] by making efficient use of the geometry of the problem, especially the representation of the landmarks using complex numbers. Their method runs in time linear with respect to the number of landmarks.

Pioneering work by Lowe [11] and Gennery [12] addresses the issue of camera pose computation, given a known (modelled) 3-D object and its corresponding image. It assumes that the imaging process is a projective transformation. The translation and orientation (with respect to the camera coordinate system) of a local coordinate system affixed to an imaged rigid object is computed. The recovery process is based on the application of Newton's method, which assumes that the function relating image appearance and object parameters is locally linear. In general the imaging equations are nonlinear and are locally linearized. Successful application of Newton's method requires starting with an appropriate initial value for the unknown parameters and, even in this case, there is still the risk of convergence to a false local minimum.

Lowe's algorithm [11] is attractive because of its elegant simplicity and powerful generality. Araújo et al. [13] proposed a fully projective formulation for Lowe's tracking algorithm resulting in dramatic improvement in accuracy with minimal increase in computation cost per iteration. Lowe's agorithm [11] and it's variants [13] are often collectively known as *model-based* pose estimation techniques, as they rely on 3D model of the viewed environment. However, their results are dependent on an accurate initial estimate of the pose, which leads us to a fundamental question: "How do we find an appropriate initialisation for pose?". Our method extends the localization method [10] to a standard perspective camera to get an initial value for Lowe's method, and use the fully projective formulation method [13] (http://www.cs.rochester.edu/u/carceron/publs/matlab /index.html) to improve on the results. Note that we do not address the often difficult problem of how the correspondence between mapped scene points and their image positions is achieved.

The novel elements of this paper are as follows: (1) the two methods of landmark-based localisation (LBL) and model-based localisation (MBL) used together, gives better localisation system than either method alone. The methods are complementary in the sense that MBL improves on the accuracy of the LBL method by refining the pose estimate, whilst the LBL method provides the necessary initial pose estimate for MBL, which would otherwise have to provided manually or from some other non-visual sensory mode, such as an odometry system, inertial navigation system or GPS system. (2) We have used a standard perspective camera instead of the "omnidirectional" camera as used in the Betke and Gurvit's work [10]. In order to use a standard camera, we need to project 3D landmarks onto a plane parallel with the ground plane. These features may be any

*The authors acknowledge the support of the UK DTI Aeronautics Research Programme.

reliably detected visual features such as a corner, line or blob. (3) We determine how the various camera parameters influence the accuracy of the pose estimate of the robot. Based on our experimental results, we can track problems and adjust parameters related to navigational systems easily and efficiently since it is difficult to get highly accurate parameters using a self-calibrated hand-held camera. (4) Since sensitive parameters have been identified which are vital to safety analysis of robot sensing systems, we can anticipate and prevent serious consequences from happening due to these parameters variations.

Briefly, the structure of this paper is as follows: in the following section, we describe the LBL method of how to determine camera/robot pose from a minimum of three viewed landmarks, whose horizontal bearings are known in a global coordinate frame. In III, we describe how LBL is used to initialise MBL. This only requires a simple coordinate transformation to match up the different reference frames used in the two systems. In the section IV, we present our experimental results in both simulated and real environments. A final section presents our conclusions.

II. LBL: Extending Betke and Gurvit's method to a standard camera

A. Input Data Organization

In this section we describe the problem of estimating a robot's position and orientation given a global map of the environment and the measured horizontal bearings between landmarks at the robot's viewpoint. We do not solve the problem of automatically identifying the landmarks and in our work this is done manually. In a fully automatic system, there are several possibilities to do this, including viewpoint invariant feature extraction and graph matching techniques. Note that Betke and Gurvit's original method [10] was applied to omnidirectional cameras. Here we adapt the system to work with a conventional camera structure, since we then want to use conventional images in a model-based pose refinement stage.

We designate the world coordinate system using a right handed system denoted by X_w, Y_w and Z_w with the origin at O_w, and the plane $X_wO_wY_w$ representing the floor. We distinguish it from the camera coordinate system which is a left handed coordinate system denoted as X_c, Y_c and Z_c with the origin at O_c (optical center of the camera). The robot's orientation angles are described by pitch (rotated along X_w), roll(rotated along Y_w) and yaw(rotated along Z_w) relative to the world coordinate system. To be able to get an initial value for Lowe's method, we assume that the retinal plane is parallel to the Z_w axis and the horizontal plane bounded by the u-axis and the camera center O_c should be parallel to the ground plane $X_wO_wY_w$ with Z_c piercing through the principal point of the image plane from the optical center O_c (Fig. 1). In this condition, pitch and roll angles are equal to zero. Likewise if we project X_c and Z_c axes and the optical center O_c along Z_w axis to the floor, we get X'_c, Y'_c and O'_c respectively since $X'_c//X_c$ and $Y'_c//Z_c$. The orientation of the robot in 2D is subtended by the X'_c and the X_w axes and is represented by the yaw angle denoted by the symbol θ.

We represent P_{w0}, ..., P_{wi}, ..., as the 3D landmark positions in the world coordinate system and P'_{w0}, ..., P'_{wi}, ..., their projected points in 2D on the floor. The robot's position is described by vector $p = (p_x, p_y, p_z)$ in the world coordinate system. Vector p links the origins of both coordinate systems. $p = O_wO_c = O_wO'_c + O'_cO_c$. If the position O'_c is found and since we know the height of the camera, the position of the robot can then be obtained.

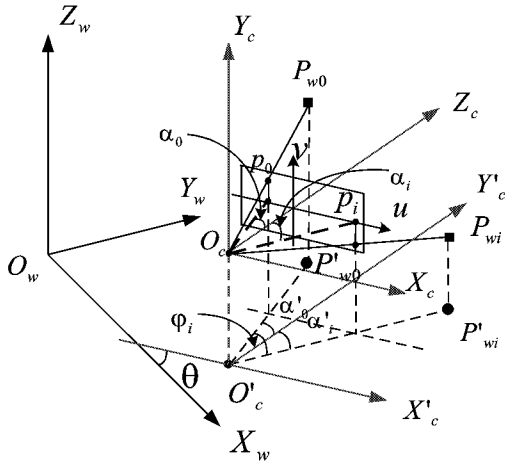


Fig. 1. Relationship of world coordinate and camera coordinate.

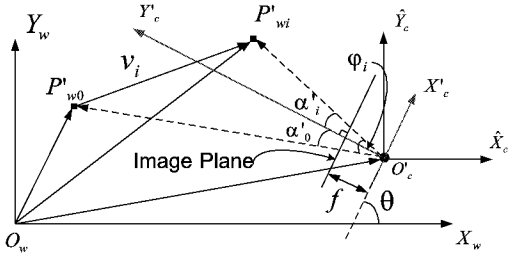


Fig. 2. Top view of Fig.1.

A map of the robot's environment means that the vectors P_{w0}, ..., P_{wi}, ..., in the world coordinate system are given. We then project these landmarks on the ground plane $X_wO_wY_w$ and represent their 2D projections as P'_{w0}, ..., P'_{wi},..., . φ_i is the angular separation between landmarks on the floor. Angular separation is defined to be the angle between each projected landmark P'_{wi} relative to P'_{w0}, which is assumed to be the reference projected landmark. The reference landmark may be chosen as the most "reliable" landmark if such reliability information exists. Note that in Fig. 1, angular separation φ_i between landmarks is computed along the plane π bounded by the optical center and the u-axis of the image plane. This horizontal plane π is assumed to be parallel to the ground

plane. The angular separation φ_i on the floor is the same as the angular separation on the plane π. Thus a calibrated camera can be used as a 2D protractor for obtaining the separation angles φ_i between landmarks. These angles are computed by projecting the image points P_0, ..., P_i, ..., of the landmarks onto the plane π. Given the focal length f of a calibrated camera and the u-coordinates u_0, ..., u_i, ..., of the projected image points of the landmarks, the angle between the optical axis and the vector O_cP_i subtended from the optical centre to each u_i can be computed by the formula $\alpha_i = arctan(u_i/f)$. We then calculate the angular separation as $\varphi_i = \alpha_i - \alpha_0$. Obviously, $\alpha'_0 = \alpha_0$ and $\alpha'_i = \alpha_i$.

Fig.2 shows a top view of Fig.1 under the world coordinate system. Both X'_c and Y'_c refers to the projection of the camera centered coordinate frame and X_w and Y_w refers to the world coordinate frame. v_i is the difference vector between the projected points of reference landmark 0 and landmarks i. Now the problem of finding the robot's pose (position and orientation) can be summarized as follows: Given the external 2D position P'_{w0}, ..., P'_{wn} of n+1 landmarks and their corresponding separation angles φ_1, ... , φ_n, estimate the position O'_c and the orientation θ of the robot.

B. Algorithm Synopsis

The projected point of each landmark P'_{wi} on the floor can be written as a complex number. For each landmark we have an expression $P'_{wi} = l_i e^{j\alpha_i} \quad for \;\; i = 1, ..., n$. where l_i is the unknown distance of the robot to the landmark projection P'_{wi}. The letter j is an imaginary unit where $j = \sqrt{-1}$. We denote the reference projection as P'_{w0} and compute the angular separation of each projection with respect to P'_{w0} as $\varphi_i = \alpha_i - \alpha_0$ for $i = 1, ..., n$. Dividing the complex number representation of each landmark projection P'_{w1}, ..., P'_{wn} by the projection of the reference landmark P'_{w0} yields a set of equations that includes the angular separation $\varphi_i = \alpha_i - \alpha_0$ for $i = 1, ..., n$.

$$\frac{P'_{wi}}{P'_{w0}} = \frac{P'_{w0} + v_i}{P'_{w0}} = \frac{l_i}{l_0} e^{j(\alpha_i - \alpha_0)} = \frac{l_i}{l_0} e^{j\varphi_i} \tag{1}$$

where $v_i = P'_{wi} - P'_{w0}$. After some algebraic operation, we obtain a set of equations whose unknowns are vectors P'_{w0}, v_i and length ratios l_i/l_0,

$$\frac{1}{P'_{w0}} = \frac{l_i}{l_0}\frac{1}{v_i} e^{j\varphi_i} - \frac{1}{v_i} \tag{2}$$

To remove the dependence on P'_{w0}, we substitute the left-hand side of Eq.(2) with the expression on the right-hand side for a different index k. The only unknowns in Eq.(3) are vector v_i and length ratios l_i/l_0.

$$\frac{l_k}{l_0}\frac{1}{v_k} e^{j\varphi_k} - \frac{1}{v_k} = \frac{l_i}{l_0}\frac{1}{v_i} e^{j\varphi_i} - \frac{1}{v_i} \tag{3}$$
$$for \;\; i, k = 1, ..., n \;\; and \;\; k \neq i$$

Since the separation angles φ_1, ..., φ_n are independent of the robot's orientation, we can rewrite Eq.(3) using a different coordinate system. This new camera coordinate system is parallel to the world coordinate system. Its axes $\hat{X}_c$ and $\hat{Y}_c$ are parallel to axes X_w and Y_w of the world coordinate frame respectively. Therefore landmarks, P'_{wi}, are described as vector $\hat{P}'_{wi}$ under this new coordinate system and Eq.(3) now becomes

$$\frac{l_k}{l_0}\frac{1}{\hat{v}_k} e^{j\varphi_k} - \frac{1}{\hat{v}_k} = \frac{l_i}{l_0}\frac{1}{\hat{v}_i} e^{j\varphi_i} - \frac{1}{\hat{v}_i} \tag{4}$$
$$for \;\; i, k = 1, ..., n \;\; and \;\; k \neq i$$

If we let $r_i = l_i/l_0$, $b_i = (b_{xi}, b_{yi}) = c_i e^{j\varphi_i}$, $c_i = (c_{xi}, c_{yi}) = \mathbf{1}/\hat{v}_i$, where b_{xi} and c_{xi} are the real coordinate, b_{yi} and c_{yi} are the imaginary coordinate. Thus Eq.(4) can be transformed into a matrix equation of the form

$$\mathbf{Ar} = \mathbf{c} \tag{5}$$

Where $\mathbf{A}$ is a $n(n-1) \times n$ matrix. The Least Squares solution of Eq.(5) yields $\mathbf{r} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{c}$ where $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \mathbf{A}^+$ is called the pseudo inverse of matrix $\mathbf{A}$. Since it is difficult to get a real useful value for $\mathbf{r}$, we will describe an effective method to obtain $\mathbf{r}$ and as a result obtaining the position estimate p of the robot in the next section.

C. Position Estimation

After initialising the values of vectors $\mathbf{v}$, $\mathbf{c}$ and $\mathbf{b}$ as described above, the procedure computes each component vector $\mathbf{s} = \frac{I}{2}\mathbf{A}^T\mathbf{c}$ as,

$$\mathbf{s}_i = nb_i^T c_i - b_i^T \sum_{j, j\neq i} c_j = n(b_{xi}c_{xi} + b_{yi}c_{yi}) - (b_{xi}\textstyle\sum_{j=1}^{n} c_{xj} + b_{yi}\sum_{j=1}^{n} c_{yj}) \tag{6}$$

after obtaining the value of $\mathbf{s}$ and given the vector $\mathbf{b}$, the procedure calculates the vector $\mathbf{r} = 2(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{s}$ which exploits the special form of matrix:

$$\mathbf{A}^T\mathbf{A} = 2(n\mathbf{D} - b_x b_x^T - b_y b_y^T) \tag{7}$$

where $\mathbf{D}$ is a diagonal matrix whose ith entry is $b_i^T b_i = b_{xi}^2 + b_{yi}^2$. Matrices $b_x b_x^T$ and $b_y b_y^T$ are outer products. Thus

$$\begin{aligned}\mathbf{r} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{c} = (\mathbf{A}^T\mathbf{A})^{-1} \cdot 2\mathbf{s} \\ &= (n\mathbf{D} - b_x b_x^T - b_y b_y^T)^{-1} \cdot \mathbf{s} \\ &= \mathbf{K}^{-1}\mathbf{s} + \tfrac{(\mathbf{K}^{-1}b_y)((\mathbf{K}^{-1}b_y)^T\mathbf{s})}{1-(\mathbf{K}^{-1}b_y)^T b_y}\end{aligned} \tag{8}$$

Where $\mathbf{K} = (n\mathbf{D} - b_x b_x^T)$.

$$\begin{aligned}\mathbf{K}^{-1}\mathbf{s} &= (n\mathbf{D} - b_x b_x^T)^{-1}\mathbf{s} \\ &= (n\mathbf{D})^{-1}\mathbf{s} + \tfrac{(n\mathbf{D})^{-1}b_x((n\mathbf{D})^{-1}b_x)^T\mathbf{s}}{1-((n\mathbf{D})^{-1}b_x)^T b_x}\end{aligned} \tag{9}$$

$$\mathbf{K}^{-1}b_y = (n\mathbf{D})^{-1}b_y + \frac{(n\mathbf{D})^{-1}b_x((n\mathbf{D})^{-1}b_x)^T b_y}{1-((n\mathbf{D})^{-1}b_x)^T b_x} \tag{10}$$

The ith diagonal entry of $(n\mathbf{D})^{-1}$ is $1/(n(b_{xi}^2 + b_{yi}^2))$. A solution to $\mathbf{r} = (r_1, r_2, ..., r_n)^T$ can then be used to solve the projection of the robot's position p_i by using Eq.(11)

$$\hat{P}_{0,i} = \frac{\mathbf{1}}{r_i b_i - c_i} \qquad p_i = (P'_{w0} - \hat{P}_{0,i}) \tag{11}$$

Ideally the set of estimates for p_i should have the same value for all $\hat{P}_{0,i}$. In reality noise will be present, so we take the centroid of the set as an estimate of the projection of the robot's position. $p = \frac{1}{n}\sum_{i=1}^{n} p_i$. Once we knew the position of the robot, its orientation θ can then be obtained by $\theta = \angle(\hat{P}'_{w0}, \hat{X}_c) - \angle(P'_{w0}, X_c)$. If the ith landmark is very close to the robot's position, the method described above may yield a negative value r_i. But r_i should be always positive. Thus, if $r_i < 0$, then let $r_i = 0$.

III. MBL Initialisation using LBL

Now we describe how LBL is used to initialise MBL. Since our focus is to determine whether LBL and MBL can work together (i.e. can LBL give an initialisation within the convergence region of MBL pose estimation in typical viewed scenes and can MBL refine this pose estimate - if not, then we might as well use LBL alone), we assume here that the correspondence problem between viewed features and mapped features is solved, accepting that it is necessary but non-trivial for any LBL system. We now only require a simple coordinate transformation to match up the different reference frames used in the two systems.

In order to use the results from landmark based method as an initial value to Lowe's method, we need to transform the right handed coordinate system output from this method to a left handed system as required by the input to fully projective formulation method. The procedure for obtaining this transformation matrix is described in four steps (Fig. 3).

(1). Transform the origin of the world coordinate to viewpoint $E(E_1, E_2, E_3)$, denoted by $\mathbf{T}_1$.

$$\mathbf{T}_1 = \begin{pmatrix} 1 & 0 & 0 & -E_1 \\ 0 & 1 & 0 & -E_2 \\ 0 & 0 & 1 & -E_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2). Rotate the new X' axis by 90^o , denoted by $\mathbf{T}_2$.

$$\mathbf{T}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(3). Rotate the new Y' axis by $\pi - \theta$, denoted by $\mathbf{T}_3$.

$$\mathbf{T}_3 = \begin{pmatrix} cos(\pi-\theta) & 0 & sin(\pi-\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\pi-\theta) & 0 & cos(\pi-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(4). Finally, reverse the X' axis, denoted by $\mathbf{T}_4$.

$$\mathbf{T}_4 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The transform matrix can be stated as follows:

$$\mathbf{T} = \mathbf{T}_4\mathbf{T}_3\mathbf{T}_2\mathbf{T}_1 = \begin{pmatrix} -cos\theta' & sin\theta' & 0 & E_1 cos\theta' - E_2 sin\theta' \\ 0 & 0 & 1 & -E_3 \\ -sin\theta' & -cos\theta' & 0 & E_1 sin\theta' + E_2 cos\theta' \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $\theta' = \pi - \theta$.

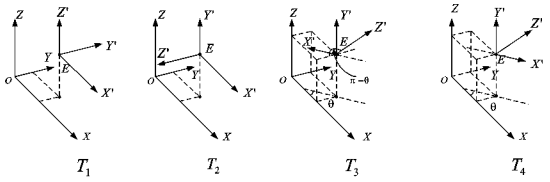


Fig. 3. Transformation from world coordinate system to camera coordinate system.

IV. Experimental Results

In this section, we first look at the results of a simulation with various levels of noise injected into various parameters in order to examine the performance and sensitivity of our algorithms. Specifically, we inject noise into both the image position of viewed features and in the camera parameters, both intrinsic and extrinsic.

The first simulation examines the sensitivity of our LBL method to all of these different noises. We then compare the performance of LBL alone with that of LBL+MBL.

A. Simulated Experiment for the LBL method

In order to analyse the effects of intrinsic and extrinsic parameters of the camera on position and orientation of the robot as well as testing the robustness of our combined LBL+MBL method, we carried out extensive experiments using synthetic data.

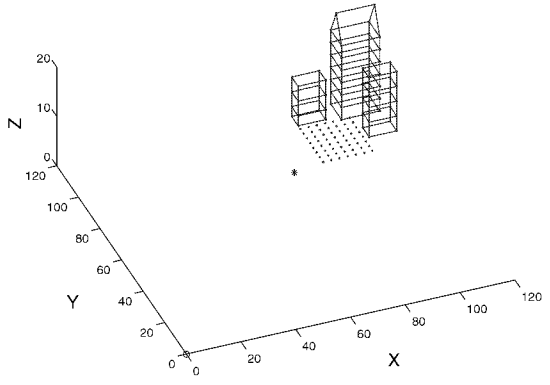
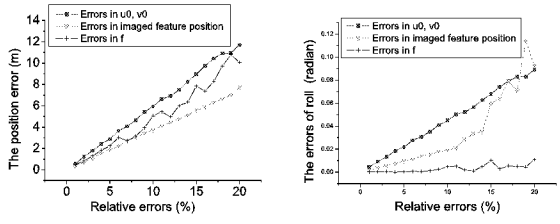


Fig. 4. 3D Euclidean scene. Symbol "*" shows position of the camera and symbol "o" shows the origin of the world coordinate system.

The simulated experiment was carried out on a 3D Euclidean model (Fig.4). The scene consists of 122 points representing the landmarks, in which case the global $(X, Y)_i$ coordinates of the points represent their projections to the ground plane. The height of the tallest 3D landmark is $15m$. The farthest distance between the camera and 3D landmark is $75.3m$. The intrinsic camera parameters were chosen as follows: the synthetic camera had an aspect ratio of one with no skew, a focal length value $f = 6.00mm$ which is invariant, $dx = dy = 0.01mm$, a principal point value of $(319, 239)$ pixels, and an image size is 640×480. The position of the camera at $O_c = (65, 65, 8)$ and the origin

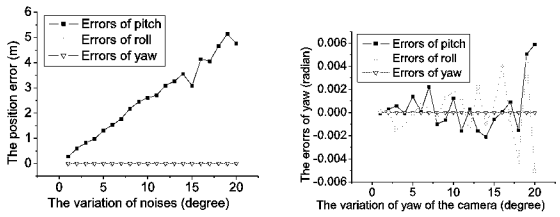
of the world coordinate system at $O_w = (0,0,0)$. Let $\mathbf{p}_{true} = (X,Y)_{true}$ and yaw_{true} signify the true value of the camera pose in the world coordinate system. In a similar manner, we denote $\mathbf{p}_{estimate}$ and $yaw_{estimate}$ as their estimated value. For all the tests that were performed, we compute the estimated values as average values executed 10,000 times under different Gaussian noise condition. The variation of relative error for each parameter ranges from 1% to 20%. The relative errors were created by adding Gaussian noises with mean equal to zero, variance equal to one and standard deviation equal to one. Figure 5 shows the influence of intrinsic parameters have on the location and orientation of the robot. Fig.5(a) shows the variation of the Euclidean error in position $d = ||p_{true} - p_{estimate}||$ with different noises added, and the yaw error is equal to $|yaw_{true} - yaw_{estimate}|$. From the experimental results of Fig.5(a) it can be observed that the position of the camera is very sensitive to the principal point (u_0, v_0) errors. When we added 20% noise to the principal point, the resulting position error is 11.71 meters. However, the position of the camera is not sensitive to feature point noise (noise on the image location of a point/corner feature). In another scenario, we added the same noise percentage to feature points, the resulting position error is equal to 7.76 meters. Fig.5(b) indicates that yaw angle is very sensitive to errors in principal point and feature points, but not to errors in focal length. From the experimental results we also know that pitch angle is highly sensitive to noises in feature points but not to noises in principal point and focal length while yaw angle is very sensitive to errors in principal point and feature points, but not to errors in focal length.



(a) (b)

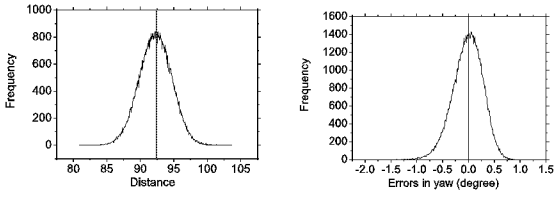
Fig. 5. The effect of intrinsic parameters on location and orientation of the robot.

Figure 6 shows the effect that extrinsic parameters have on location and orientation of the robot. Fig.6(a) shows that the position of the camera is highly sensitive to pitch but not to roll and yaw. Fig 6(b) illustrate that the variation of the extrinsic parameter yaw does not affect the orientation of the robot at all but is highly sensitive to pitch and roll angles. So in practice, we should maintain the X_c axis of the camera parallel to the ground plane and the Y_c axis perpendicular to the ground plane. (Alternatively we may use ground plane homography techniques to determine the orientation of the ground plane with respect to the camera.)



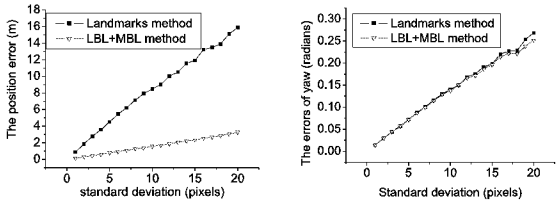
(a) (b)

Fig. 6. The effect of extrinsic parameters on location and orientation of the robot.



(a) (b)

Fig. 7. Distributions of position and orientation errors with 5% Gaussian noise added to the focal length f with mean equal to zero, variance equal to one and standard deviation equal to one.



(a) (b)

Fig. 8. Comparison between landmarks and our method.

In order to test the robustness of our method, we set $\theta = 0^o$ and added 5% Gaussian noise to the focal length f with mean equal to zero, variance equal to one and standard deviation equal to one. We ran the program 100,000 times under different noise conditions. Figure 7 shows how the distance of the camera to the origin of the world coordinate and $|yaw_{true} - yaw_{estimate}|$ are distributed for different noise variation. Fig.7(a) shows distance with mean equal to $92.25m$ and standard deviation equal to $2.41m$. The ground truth distance is $92.27m$. Fig.7(b) shows errors in yaw with mean equal to 0.01503^o and standard deviation equal to 0.28787^o. The ground truth is 0^o. Note that the distributions are almost normal, and the mean values are very near to ground truth values. For different noise levels, the computed results are similar which proves that our method is very robust under different noise conditions.

B. Simulated Experiment for the LBL+MBL method

The 3D points $(X,Y,Z)_i$ in the model shown in fig. 4 form the known 3D model used in the MBL phase of our

method. Figure 8 shows the results of the LBL alone and our method which combines LBL and MBL. The estimated values we got are all the average of 10, 000 runs by adding different Gaussian noises to the image coordinates. The mean is zero and the standard deviation from 1 to 20 pixels. Fig.8(a) Shows the variation of distance from T_{true} to $T_{estimate}$ with different noises. T_{true} and $T_{estimate}$ are the ground truth and estimated values of the position of the robot. Fig.8(b) shows the variation of the errors of yaw (the robot's orientation in the direction of the axis x) relative to the real value. The results show that using our method, we can get more accurate and stable camera position than the landmarks method. This is because in Lowe's method, the rotation parameters $\mathbf{R}$ and focal length f remain the same as in the previous transform, but the position vector $\mathbf{t}$ has been replaced by the new parameters D_x, D_y and D_z, and then Newton's method is carried out by calculating the optimum correction rotations $\Delta\phi_x$, $\Delta\phi_y$ and $\Delta\phi_z$ to be made about the camera-centred axes. New parameters D_x, D_y simply specify the location of the object on the image plane and D_z specifies the depth of the object from the camera. So when Lowe's method was used after we get initial value from landmarks method, we can get more accurate position of the camera, but the value for rotation is about the same, as shown in figure 8(b).

C. Real Image Experiment for LBL+MBL

Figure 9 shows the three images of a scene, viewed from different robot/camera poses. The scene contains artificial landmarks for which we have manually measured the 3D map $(X, Y, Z)_i$ of features. The map of features is defined by the centres of the chessboard targets, which can be detected by a standard corner detector, such as Plessey-Harris detector [14]. Landmarks are obtained by taking the ground plane projections of these points, namely $(X, Y)_i$. (Note that our method also works with natural features, but we have not focussed on the extraction and identification of natural mapped features in this work.) The first image was taken with the position of the camera located at (64.7, 136.7)cm and an orientation angle $\theta = 0^o$. The second image was taken with the position of the camera shifted to (-10, 145)cm and orientation angle $\theta = -20^o$. The third image was taken with the position of the camera located at (126, 125)cm and orientation angle $\theta = 10^o$. The height of the camera is measured to be 110 cm. For the camera coordinate system, the $X_cO_cZ_c$ plane is parallel to the ground plane. The experimental results are shown in table I. In this experiment, the maximum distances from robot to the object are 332.3cm at the first position, 318.6cm at the second position, and 327.7cm at the third position. From the data obtained, we have computed the relative errors of the distances between real and estimated position to be 0.84%, 1.29% and 0.73%, respectively. The experimental results show that our proposed method in this paper is viable for practical purposes.

V. Conclusions

Based on both LBL and MBL methods, we have proposed an algorithm for visually localizing a mobile robot using a single, standard monocular camera. Firstly, in order to get accurate and robust initial pose estimates, we have extended Betke and Gurvit's LBL method to standard perspective cameras. We have analyzed the robustness of this LBL method and tested its accuracy by carrying out intensive experiments using synthetic data. Different parameters have different effects on the LBL position and orientation estimates of the robot. Some parameters are more important than others. For example, the position of the camera is highly sensitive to errors in principal point u_0, v_0, but not to feature position errors. The orientation is very sensitive to feature position errors, but not to errors in focal length. This gives us insight into how to get good initial pose estimates, which are then more likely to be within the convergence region of any iterative MBL refinement scheme. We find that MBL can improve the position estimate of the robot but not the orientation error, when the system is subject to various levels of feature position noise. In an experiment with real images in the combined LBL+MBL method, the average relative errors for X, Y, Z are 0.95%. The average absolute errors for pitch, roll and yaw are 1.7^o, 1.5^o and 0.5^o, respectively. Our thorough sensitivity analysis through simulated experiment and our initial experiments with real images indicate that the proposed method is suitable for practical visual localisation applications.

TABLE I
EXPERIMENTAL RESULTS USING REAL IMAGE.

Different values	Position(cm) (X,Y,Z)	Angle(^{o}) (P, R, Y)
Ground truth value	(64.7,136.7,110)	(0,0,0)
	(-11,148.1,110)	(0,-20,0)
	(126.2,125,110)	(0,10,0)
Computed results	(67.1,135.3,110.1)	(1.6,-3.1,-0.4)
	(-10,146.8,113.8)	(2.1,-20.03,-0.2)
	(127,125.5,107.8)	(1.5,11.3,-1.0)
Absolute errors	2.8	(1.6,3.1,0.4)
	4.1	(2.1,0.03,0.2)
	2.4	(1.5,1.3,1.0)

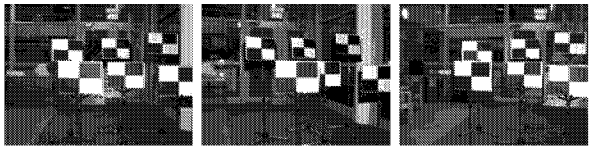


Fig. 9. Known artifical landmarks and scene model viewed from three poses.

REFERENCES

[1] D. Lowe, "Object recognition from local scale-invariant features," In proceedings of the Seventh International Conference on Computer Vision (ICCV'99), Kerkyra, Greece, September, 1999, pp.1150-1157.

[2] S. Se, D. Lowe, and J. Little, "Local and global localization for mobile robots using visual landmarks," In Proceedings of the IEEE/RSJ international Conference on Intelligent Robots and Systems (IROS), Maui, Hawaii, October, 2001, pp.414-420.

[3] S. Se, D. Lowe and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, May, 2001, pp.2051-2058.

[4] S. Se, D. Lowe and J. Little, "Global localization using distinctive visual features," In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, October, 2002, pp.226-231.

[5] S. Thrun, A. Buecken, W. Burgard, D. Fox, T. Froehlinghaus, D. Henning, T. Hofmann, M. Krell, and T. Schmidt, "Map learning and high-speed navigation in RHINO", In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems,AAAI Press, 1998, pp.21–52.

[6] W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The interactive museum tour-guide robot", In Proceedings of the ifteenth National Conference on Artificial Intelligence (AAAI-98), Madison, Wisconsin, July 1998, pp.11-18

[7] D. Fox, W. Burgard, F. Dellaert and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots", In Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida, July, 1999, pp.343-349.

[8] F. Dellaert, W. Burgard, D. Fox and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99), Fort Collins, Colorado, June, 1999, pp.588-594.

[9] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," International Journal of Computer Vision, 1998, 29(1), pp.5-28.

[10] M. Betke and L. Gurvits, "Mobile Robot Localization Using Landmarks," IEEE Trans. on Robotics and Automation, 1997, 13(2), pp.251-263.

[11] D. Lowe, "Fitting parameterised three-dimensional models to images," IEEE trans. On Pattern Analysis and Machine Intelligence, May, 1991, 13(5), pp.441-450.

[12] D. Gennery, "Visual tracking of known three-dimensional objects," International Journal of Computer Vision, April, 1992, 7(3), pp.243-270.

[13] H. Araújo, R.L. Carceroni and C.M. Brown, "A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose-Estimation Algorithm," Computer Vision and Image Understanding, May, 1998, 70(2), pp.227-238.

[14] C. J. Harris and M. Stephens. A combined corner and edge detector. In 4th Alvey Vision Conference Manchester, 1988, pp. 147-151.