Unified Inverse Depth Parametrization for Monocular SLAM

Conference Paper · August 2006

DOI: 10.15607/RSS.2006.II.011 · Source: DBLP

3 authors, including:

J. M. M. Montiel
University of Zaragoza
65 PUBLICATIONS **4,288** CITATIONS
SEE PROFILE

Javier Civera
University of Zaragoza
53 PUBLICATIONS **2,188** CITATIONS
SEE PROFILE

Unified Inverse Depth Parametrization for Monocular SLAM

J.M.M. Montiel
Dpto. Informatica e Ingenieria de Sistemas
Universidad de Zaragoza. Spain
Email: josemari@unizar.es

Javier Civera
Dpto. Informatica e Ingenieria de Sistemas
Universidad de Zaragoza. Spain
Email: jcivera@unizar.es

Andrew J. Davison
Department of Computing
Imperial College London. UK
Email: ajd@doc.ic.ac.uk

***Abstract*— Recent work has shown that the probabilistic SLAM approach of explicit uncertainty propagation can succeed in permitting repeatable 3D real-time localization and mapping even in the 'pure vision' domain of a single agile camera with no extra sensing. An issue which has caused difficulty in monocular SLAM however is the initialization of features, since information from multiple images acquired during motion must be combined to achieve accurate depth estimates. This has led algorithms to deviate from the desirable Gaussian uncertainty representation of the EKF and related probabilistic filters during special initialization steps.**

In this paper we present a new unified parametrization for point features within monocular SLAM which permits efficient and accurate representation of uncertainty during undelayed initialisation and beyond, all within the standard EKF (Extended Kalman Filter). The key concept is direct parametrization of inverse depth, where there is a high degree of linearity. Importantly, our parametrization can cope with features which are so far from the camera that they present little parallax during motion, maintaining sufficient representative uncertainty that these points retain the opportunity to 'come in' from infinity if the camera makes larger movements. We demonstrate the parametrization using real image sequences of large-scale indoor and outdoor scenes.

I. Introduction

A monocular camera is a projective sensor which measures the bearing of image features. To infer the depth of a feature the camera must observe it repeatedly as it translates through the scene, each time capturing a ray of light from the feature to its optic center. The angle between the captured rays is the feature's *parallax* — this is what allows its depth to be estimated.

In computer vision, the well-known concept of a point at infinity is a feature which exhibits no parallax during camera motion due to its extreme depth. A star for instance would be observed at the same image location by a camera which translated through many kilometers pointed up at the sky without rotating. Such a feature cannot be used for estimating camera translation but is a perfect bearing reference for estimating rotation. The homogeneous coordinate systems of visual projective geometry allow explicit representation of points at infinity, and they have proven to play an important role during off-line optimization-based structure and motion estimation from image sequences.

Recent research has shown that the way to improve on off-line sequence estimation and achieve sequential, repeatable motion and structure estimation with a moving camera is to adopt the probabilistic SLAM (Simultaneous Localization and Mapping) approach of explicit uncertainty propagation familiar from mobile robotics. Davison [2] proved that the standard EKF formulation of SLAM can be very successful even when the only source of information is the video from an agile single camera, demonstrating real-time 30Hz motion and structure estimation in 3D.

A significant limitation of Davison's approach, however, was that it could only make use of features within close range of the camera which exhibited significant parallax, and was therefore practically limited to room-scale scenes. The problem was in initialising uncertain depth estimates for distant features. Acknowledging that feature depth uncertainty during initialisation is not well-modelled by a standard Gaussian distribution in Euclidean space, Davison used a particle approach to represent a feature's depth coordinate until conversion to Gaussian representation when the distribution had collapsed sufficiently. Aside from being able to deal only with feature depths within the small pre-defined range along which particles were spread (around 1 to 5 meters), this 'delayed' style of initialisation meant that observations of features were not used to update the camera pose estimate until their conversion into fully initialised features.

It would be relatively simple to deal with points at infinity in SLAM if it were known in advance which features were at infinity and which were not. Those at infinity would be modelled with a special 'direction' parametrization, ignoring their depth, while finite features maintained the standard form. Montiel [8] showed that in the special case where *all features are known to be infinite* — in very large scale outdoor scenes or when the camera rotates on a tripod — SLAM in pure angular coordinates turns the camera into a real-time visual compass.

In the more general case, the difficulty is that we do not know in advance which features are infinite and which are not. We should clarify the discussion by defining the meaning of 'infinity' in the current context. Of course no observable feature is truly infinitely far from the camera (even a star of course has a finite depth). A point at infinity is simply far enough away *relative to the camera motion since it has been observed* that no parallax has been observed.

Let us imagine a camera moving through a 3D scene with

observable features at a range of depths. From the estimation point of view, we can think of all features starting at infinity and 'coming in' as the camera moves far enough to measure sufficient parallax. For nearby indoor features, only a few centimetres of movement will be sufficient. Distant features may require many meters or even kilometers of motion before parallax is observed. It is important that these features are not permanently labelled as infinite — a feature that seems to be at infinity should always have the chance to prove its finite depth given enough motion, or there will be the serious risk of systematic errors in the scene map. Our probabilistic SLAM algorithm must be able to represent that uncertainty in depth of seemingly infinite features. Observing no parallax for a feature after 10 meters of camera translation does tell us something about its depth — it gives a reliable lower bound. We feel that this consideration of uncertainly in locations of points has not been previously required in off-line computer vision algorithms, but that now we have a method for dealing with it in the more difficult on-line case.

Our contribution in this paper is to show that in fact there is a unified and straightforward parametrization for feature locations which can handle both initialisation and standard tracking of both close and very distant features within the standard EKF framework. An explicit parametrization of the *inverse depth* allows a Gaussian distribution to cover uncertainty in depth which spans a depth range from nearby to infinity, and permits seamless crossing over to finite depth estimates of features which have been apparently infinite for long periods of time.

The fact is that the projective nature of a camera means that the image measurement process is nearly linear in this inverse depth coordinate. This is a principle which should perhaps have been noted sooner in SLAM, because inverse depth is a concept used widely in computer vision: it appears in the relation between the image disparity and a point depth in stereo vision; it is interpreted as the parallax with respect to the plane at infinity in [4]; inverse depth is also used to relate the motion field induced by scene points with the camera velocity in optical flow analysis [5], and in Structure from Motion error analysis [9], [1].

The unified representation means that our algorithm requires no special initialisation process for features. They are simply tracked right from the start, immediately contribute to improved camera estimates and have their correlations with all other features in the map correctly modelled. That this can be achieved within the standard EKF means that all the great benefits it offers are maintained in terms of highly efficient representation of correlated uncertainty. We strongly believe that EKF maps, or networks of EKF submaps, will continue to have a central role in SLAM. When parametrizations are chosen carefully, there is often no need to use filtering techniques using particles (e.g. [7]) for instance which can explicitly represent non-Gaussian distributions but have their own disadvantages. Note that our parameterization would be equally compatible with other variants of Gaussian filtering such as sparse information filters.

Sola *et al.* [10] also recently proposed an interesting new approach to monocular feature initialization. In their work, an undelayed initialization of new points was based on maintaining several depth hypotheses as Gaussian volumes for each initialized feature spread in a geometric sum — a development of the particle method of Davison but taking advantage to some extent of the inverse depth concept. As the estimation proceeds, the hypotheses are pruned and an approximation to the Gaussian Sum Filter is proposed keep the computational overhead low. Their results are validated with 2D simulations combining odometry and vision and appear impressive. However, we believe that our approach has significant benefits in terms of uniformity, clarity and simplicity. Further, they make no claims about being able to cope with features at very large 'infinite' depths.

In very recent work, Eade and Drummond have presented an inverse depth initialisation scheme within the context of their FastSLAM-based system for monocular SLAM [3]. Their method which shares many similarities with our approach, and they offer some of the same arguments about advantages in linearity. The position of each new partially initialised feature added to the map is parametrized with three coordinates representing its direction and inverse depth relative to the camera pose at the first observation, and estimates of these coordinates are refined within a set of Kalman Filters for each particle of the map. Once the inverse depth estimation has collapsed, the feature is converted to a fully initialised standard Euclidean representation. While retaining the differentiation between partially and fully-initialised features, they go further and are able to use measurements of partially initialised features with unknown depth to improve estimates of camera orientation via a special epipolar update step.

Their approach certainly appears appropriate within a FastSLAM implementation. However, it lacks the satisfying unified quality of the parametrization we present in this paper, where the transition from partially to fully initialised need not be explicitly tackled and full use is automatically made of all of the information available in measurements. It is this which makes it suitable for direct use in an EKF framework for sparse mapping, with all the advantages that offers in terms of complete and correct representation of uncertainty and correlations. Besides, our system is able to code in the map distant points, in which the inverse depth coding never collapses and cannot be coded with the standard Euclidean representation.

Section II is devoted to the camera motion model, and the parametrization of inverse depth is detailed. The measurement equation is described in section III, and a discussion about measurement equation linearization errors is included. Next, feature initialization from a single feature observation is detailed in Section IV. The paper ends with experimental validation (Section V) over real image sequences captured at 30Hz in large scale environments both indoors and outdoors; links to movies describing the system performance are provided.

II. State Vector Definition

A constant angular and linear velocity model is used to code the hand-held camera motion, so the camera state $\mathbf{x}_v$ is composed of location: $\mathbf{r}^{WC}$ camera optical center, $\mathbf{q}^{WC}$ quaternion defining orientation; velocity $\mathbf{v}^W$ and angular velocity ω^W:

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{r}^{WC} \\ \mathbf{q}^{WC} \\ \mathbf{v}^W \\ \omega^W \end{pmatrix}. \quad (1)$$

At every step it is assumed an unknown linear and angular acceleration zero mean Gaussian processes, $\mathbf{a}^W$ and α^W, producing an impulse of linear and angular velocity:

$$\mathbf{n} = \begin{pmatrix} \mathbf{V}^W \\ \Omega^W \end{pmatrix} = \begin{pmatrix} \mathbf{a}^W \Delta t \\ \alpha^W \Delta t \end{pmatrix}. \quad (2)$$

The state update equation for the camera is:

$$\mathbf{f}_v = \begin{pmatrix} \mathbf{r}^{WC}_{k+1} \\ \mathbf{q}^{WC}_{k+1} \\ \mathbf{v}^W_{k+1} \\ \omega^W_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{r}^{WC}_k + \left(\mathbf{v}^W_k + \mathbf{V}^W_k\right)\Delta t \\ \mathbf{q}^{WC}_k \times \mathbf{q}\left(\left(\omega^W_k + \Omega^W\right)\Delta t\right) \\ \mathbf{v}^W_k + \mathbf{V}^W \\ \omega^W_k + \Omega^W \end{pmatrix} \quad (3)$$

being $\mathbf{q}\left(\left(\omega^W_k + \Omega^W\right)\Delta t\right)$ the quaternion defined by the rotation vector $\left(\omega^W_k + \Omega^W\right)\Delta t$.

A scene 3D point i is defined by the dimension 6 state vector (see Fig 1):

$$\mathbf{y}_i = \begin{pmatrix} x_i & y_i & z_i & \theta_i & \phi_i & \rho_i \end{pmatrix}^\top \quad (4)$$

which models a 3D point located at (see Fig 1):

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right). \quad (5)$$

The state codes the *ray for the first point observation* as: x_i, y_i, z_i, the camera optical center where the 3D point was first observed; and θ_i, ϕ_i azimuth and elevation (coded in the absolute reference) for the ray directional vector $\mathbf{m}\left(\theta_i, \phi_i\right)$. The point depth along the ray d_i is coded by its inverse $\rho_i = 1/d_i$.

The features $\mathbf{y}_i$ are considered as constant along the estimate. It is assumed no unknown input acting on the feature location.

The whole state vector $\mathbf{x}$ is the composed of the camera and all the map features:

$$\mathbf{x} = \left(\mathbf{x}_v^\top, \mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots \mathbf{y}_n^\top\right)^\top. \quad (6)$$

III. Measurement Equation

Each observed feature imposes a constraint between the camera location and the corresponding map feature (see Fig 1). The rotation is coded in the rotation matrix $R^{CW}\left(\mathbf{q}^{WC}\right)$, depending on the camera orientation quaternion. The observation

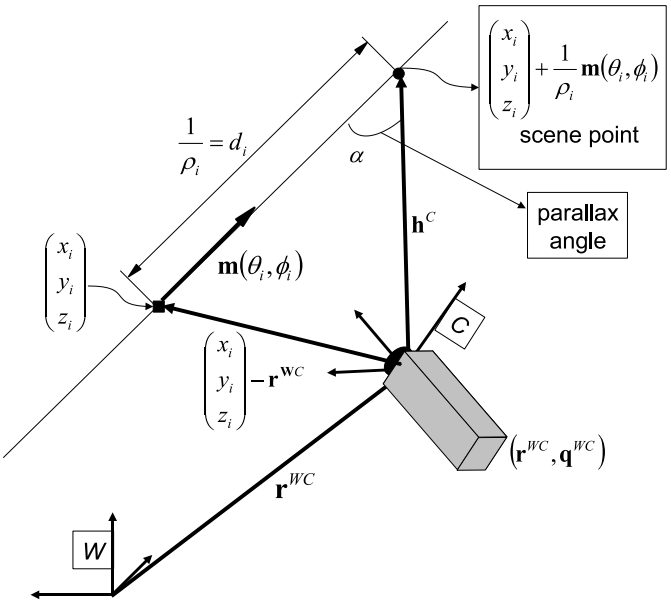

Fig. 1. Feature parametrization and measurement equation.

of a point $\mathbf{y}_i$ from a camera location defines a ray expressed in the camera frame as $\mathbf{h}^C = \begin{pmatrix} h_x & h_y & h_z \end{pmatrix}^\top$:

$$\mathbf{h}^C = \mathbf{R}^{CW}\left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right) - \mathbf{r}^{WC}\right) \quad (7)$$

which is almost equivalent to the next expression if coded with d_i:

$$\mathbf{h}^C = \mathbf{R}^{CW}\left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + d_i\mathbf{m}\left(\theta_i, \phi_i\right) - \mathbf{r}^{W}\right) \quad (8)$$

The difference is that (7) can code a point at infinity using $\rho_i = 0$, even in that case, (7) can be rewritten as:

$$\mathbf{h}^C = \mathbf{R}^{CW}\left(\rho_i\left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \mathbf{r}^{WC}\right) + \mathbf{m}\left(\theta_i, \phi_i\right)\right), \quad (9)$$

analogously, (8) can code a point at zero depth while not (7) nor (9) can.

The camera does not observe directly $\mathbf{h}^C$, but its projection in the the image according to the pinhole model. First, the projection is modeled on the normalized retina:

$$\upsilon = \frac{h_x}{h_z} \quad (10)$$

$$\nu = \frac{h_y}{h_z} \quad (11)$$

and then it is applied the camera calibration to produce the pixel coordinates for the observed point:

$$\mathbf{h} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - \frac{f}{d_x}\upsilon \\ v_0 - \frac{f}{d_y}\nu \end{pmatrix} \quad (12)$$

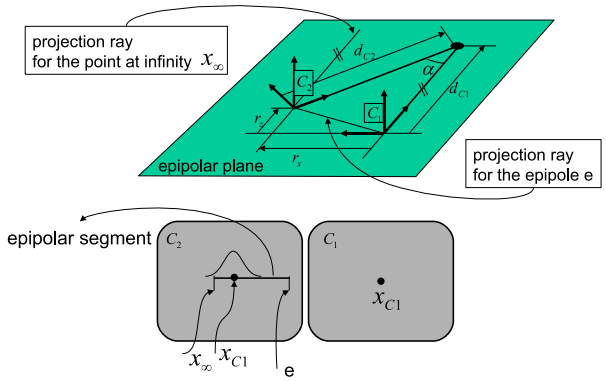

Fig. 2. Observation of a point by two cameras. The geometry has been defined with respect to the epipolar plane. Bottom subfigure shows the same geometry as observed by the cameras

where, u_0,v_0 are the camera center in pixels, f is the focal length and, d_x and d_y the pixel size.

Finally, a radial distortion model has to be applied in order to deal with real camera lenses. In this work we have used the standard photogrammetry two parameters distortion model [6].

It is worth noting, that the measurement equation has a sensitive dependency on the parallax angle α (see Fig. 1). In our calibrated camera context, the parallax is the angle defined by the two rays defined by the same scene point when observed from two different view points. At low parallax, both rays are almost parallel and:

$$\rho_i \left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \mathbf{r}^{WC} \right) + \mathbf{m}\left(\theta_i, \phi_i\right) \approx \mathbf{m}\left(\theta_i, \phi_i\right)$$

what implies that equation (9) can be approximated by:

$$\mathbf{h}^C \approx \mathbf{R}^{CW}\left(\mathbf{m}\left(\theta_i, \phi_i\right)\right)$$

and the measurement equation only provides information about the camera orientation and about the directional vector $\mathbf{m}\left(\theta_i, \phi_i\right)$. This particular case has been exploited in [8] to build a visual compass based on SLAM.

A. Measurement equation linearity

We are using the EKF to estimate the state. The more linear the measurement equation is, the better performance is expected from the Kalman filter. Next, we show how at low parallax angles, equation (7), coded in ρ, improves the linearization when compared with equation (8), coded in d. Because of that we parameterize on the inverse depth.

We focus on the observation of a point from two camera locations (see Fig 2) C_1 (absolute frame) and C_2. The references are aligned with respect to the epipolar plane (defined by the scene point and the two cameras optical centers, see [4] for a detailed explanation) to simplify the measurement equation. The Z axis is aligned with the ray defined by the optical center and the observed point. The Y axis is normal to the epipolar plane. Given a point imaged in C_1 as x_{C_1} its image on C_2, x_{C_2} is constrained to be (if in front of the cameras) on the epipolar segment defined by the epipole (the image of $C1$ on $C2$) and x_∞ (the image on x_{C_2} if the scene point where at infinity). Hence the measurement equation is defined by:

$$\mathbf{y} = \left(0, 0, 0, 0, 0, \frac{1}{d_{c_1}}\right)^T \quad (13)$$

$$\mathbf{R}_{C_2C_1} = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{pmatrix} \quad (14)$$

$$\mathbf{r}^{CW} = \left(r_x, 0, r_z\right). \quad (15)$$

Applying equation (10) to the two different parameterizations, (7) or (8) we obtain corresponding measurement equations for the two parameterizations: $\upsilon\left(\rho\right)$ and $\upsilon\left(d\right)$.

We propose to compare the two parameterizations in terms of their linearity, first we focus on $\upsilon\left(\rho\right)$ then the analysis is extended to $\upsilon\left(d\right)$ and finally a comparison is made.

If $\upsilon\left(\rho\right)$ were perfectly linear in ρ, then $\frac{\partial \upsilon}{\partial \rho}$ should be a constant, modeling ρ as Gaussian, its variation around the linearization point ρ_0 is expected to be in the interval $[\rho_0 - 2\sigma_\rho, \rho_0 + 2\sigma_\rho]$. Next we analyze the first derivative change in that interval.

A first order approximation for the *first derivative* in the interval $[\rho_0 - 2\sigma_\rho, \rho_0 - 2\sigma_\rho]$ is given by the first order Taylor expansion around ρ_0:

$$\frac{\partial \upsilon}{\partial \rho}\left(\rho_0 + \Delta\rho\right) \approx \left.\frac{\partial \upsilon}{\partial \rho}\right|_{\rho_0} + \left.\frac{\partial^2 \upsilon}{\partial \rho^2}\right|_{\rho_0} \Delta\rho. \quad (16)$$

We propose to use the dimensionless ratio between the derivative increment at the interval extreme $\left.\frac{\partial^2 \upsilon}{\partial \rho^2}\right|_{\rho_0} 2\sigma_\rho$ and the derivative in the linearization point $\left.\frac{\partial \upsilon}{\partial \rho}\right|_{\rho_0}$ as a linearity measurement. So:

$$\frac{\frac{\partial^2 \upsilon}{\partial \rho^2} 2\sigma_\rho}{\frac{\partial \upsilon}{\partial \rho}} \approx 0 \quad (17)$$

in order to have an acceptable linearization.

We compute the dimensionless ratio for the ρ parametrization:

$$\frac{2\sigma_\rho}{\rho_0} 2\left(1 - \frac{d_{C_1}}{d_{C_2}}\cos\alpha\right) \approx 0 \quad (18)$$

Which says that, at low parallax, and when $\frac{d_{C_1}}{d_{C_2}} \approx 1$, the term $\left(1 - \frac{d_{C_1}}{d_{C_2}}\cos\alpha\right) \approx 0$ and low linearization error can be achieved even if $\frac{2\sigma_\rho}{\rho_0} \gg 0$. So huge initial uncertainty regions can be coded Gaussianly. For example, considering $\alpha = 5^\circ$ $\sigma_\rho = 0.5, \rho_0 = 0.5$ the coded acceptance region extends from $[0.67, \infty]$, and the ratio is only 0.8%.

When the parallax angle increases, $\left(1 - \frac{d_{C_1}}{d_{C_2}}\cos\alpha\right)$ also increases, but the uncertainty in ρ reduces and hence $\frac{2\sigma_\rho}{\rho}$ is reduced and condition (18) is fulfilled even with moderate or high parallax angles.

When we compute (17) for the d parametrization:

$$\frac{2\sigma_d}{d_{C_2}}\left(2\cos\alpha\right) \approx 0 \quad (19)$$

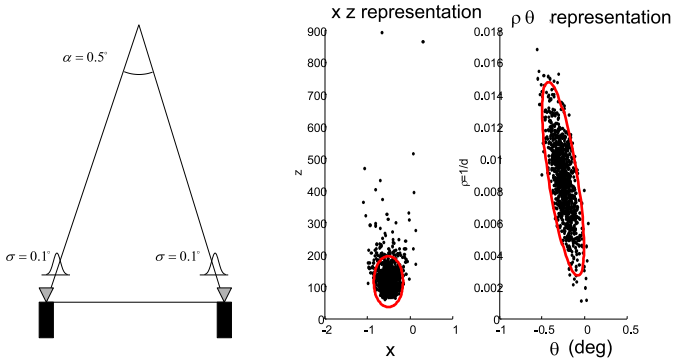

Fig. 3. Simulation of a point reconstruction from two low parallax observations. It is show how the reconstruction error coded in ρ, θ is Gaussian while coded as cartesian XZ is not Gaussian. Red ellipses represent linear uncertainty propagation from the rays Gaussian error

so, at low parallax, $\cos\alpha \approx 1$, and hence a good linearization can be achieved only if:

$$\frac{2\sigma_d}{d_{C_2}} \approx 0 \Rightarrow \sigma_d \ll d_{C_2} \tag{20}$$

which makes difficult coding huge initial uncertainty regions. For example, $\alpha = 5°, d_{C_1} = 20, \sigma_d = 10$ code an acceptance interval $[0, 40]$ and the ratio is 200%.

As an example of the improvement in the measurement equation linearization, figure 3 shows a simulation of a low parallax $(0.5°)$ point reconstruction when observed by two cameras at known locations. The cameras observe the rays with a Gaussian error, $\sigma = 0.1°$. It is shown the 3D point reconstruction modeled with XZ cartesian coordinates or with ρ, θ coordinates. The 95% uncertainty region propagated from the image error is plotted as well. It is shown the Gaussianity in ρ, θ but not in XZ.

IV. Feature Initialization

It is a remarkable quality of our proposal that new features are initialized using only one image, the image where the feature is first observed; the initialization includes both the feature state initial values and the covariance assignment. Despite the initial uncertainty region covers a huge range depth ($[1, \infty]$ in our experiments) because of the low linearization errors (18) the uncertainty is successfully coded as Gaussian; once initialized, the feature is processed with the standard EKF prediction-update loop.

It is worth noting, that thanks to the proposed parametrization, while the feature is observed at low parallax, the feature will be used mainly to determine the camera orientation but the feature depth will be kept quite uncertain, including in its uncertainty region the even infinity; if the camera translation is able to produce a parallax big enough then the feature depth estimation will be improved.

The initial location for the observed feature is defined as:

$$\hat{\mathbf{y}}\left(\hat{\mathbf{r}}^{WC}, \hat{\mathbf{q}}^{WC}, \mathbf{h}, \rho_0\right) = \begin{pmatrix} \hat{x}_i & \hat{y}_i & \hat{z}_i & \hat{\theta}_i & \hat{\phi}_i & \hat{\rho}_i \end{pmatrix}^\top \tag{21}$$

from the camera location estimate at step k (the k indexes have been dropped for simplicity), and the observation of a new feature: $\mathbf{h} = \begin{pmatrix} u & v \end{pmatrix}^\top$ and, the initial ρ_0.

The projection ray initial point (see Fig 1) is directly taken from the current camera location estimate:

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{pmatrix} = \hat{\mathbf{r}}^{WC}_{k|k} \tag{22}$$

The projection ray directional vector is computed from the observed point, expressed in the absolute frame:

$$\mathbf{h}^W = \mathbf{R}_{WC}\left(\mathbf{q}^{WC}_{k|k}\right)\mathbf{h}^C \begin{pmatrix} \upsilon \\ \nu \\ 1 \end{pmatrix} \tag{23}$$

being υ and ν the image in the normalized retina. Despite being $\mathbf{h}^W$ a non-unitary directional vector, the angles can be derived as:

$$\begin{pmatrix} \theta_i \\ \phi_i \end{pmatrix} = \begin{pmatrix} \arctan\left(-\mathbf{h}^W_y, \sqrt{{\mathbf{h}^W_x}^2 + {\mathbf{h}^W_z}^2}\right) \\ \arctan\left(\mathbf{h}^W_x, \mathbf{h}^W_z\right) \end{pmatrix} \tag{24}$$

The covariance for $\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{\theta}_i$, and $\hat{\phi}_i$ is derived from the image measurement error covariance $\mathbf{R}_j$ and the state covariance estimate $\hat{\mathbf{P}}_{k|k}$.

The initial value for ρ_0 is derived heuristically to cover in its 95% acceptance region a working space from infinity to a predefined close distance, $d_{\min}$ expressed as inverse depth: $\left[\frac{1}{d_{\min}}, 0\right]$, so:

$$\hat{\rho}_0 = \frac{\rho_{\min}}{2} \quad \sigma_\rho = \frac{\rho_{\min}}{4} \quad \rho_{\min} = \frac{1}{d_{\min}}. \tag{25}$$

In our experiments $d_{\min} = 1, \hat{\rho}_0 = 0.5, \sigma_\rho = 0.25$.

The state covariance after feature initialization is:

$$\hat{\mathbf{P}}^{\text{new}}_{k|k} = \mathbf{J}\begin{pmatrix} \hat{\mathbf{P}}_{k|k} & 0 & 0 \\ 0 & \mathbf{R}_j & 0 \\ 0 & 0 & \sigma^2_\rho \end{pmatrix}\mathbf{J}^\top$$

$$\mathbf{J} = \left(\begin{array}{c|c} I & 0 \\ \hline \frac{\partial \mathbf{y}}{\partial \mathbf{r}^{WC}}, \frac{\partial \mathbf{y}}{\partial \mathbf{q}^{WC}}, 0, \ldots, 0, & \frac{\partial \mathbf{y}}{\partial \mathbf{h}}, \frac{\partial \mathbf{y}}{\partial \rho} \end{array}\right)$$

V. Experimental Results

The performance has been tested on real image sequences acquired with hand-held low cost Unibrain IEEE1394 camera, with a $90°$ field of view and 320×240 resolution monochrome at 30 fps.

Our current experiments are run in Matlab; however we believe that 30Hz performance could be achieved in real time. Current C++ implementations for monocular SLAM with dimension 3 for every point feature can run at 30 Hz. for maps up to 100 features. Our feature is dimension six. However our system offers computational load advantages: i) the simple feature intialization is cheaper than the current approaches. ii) Several features can initialized from a frame and rotation information is obtained from the second time a feature is

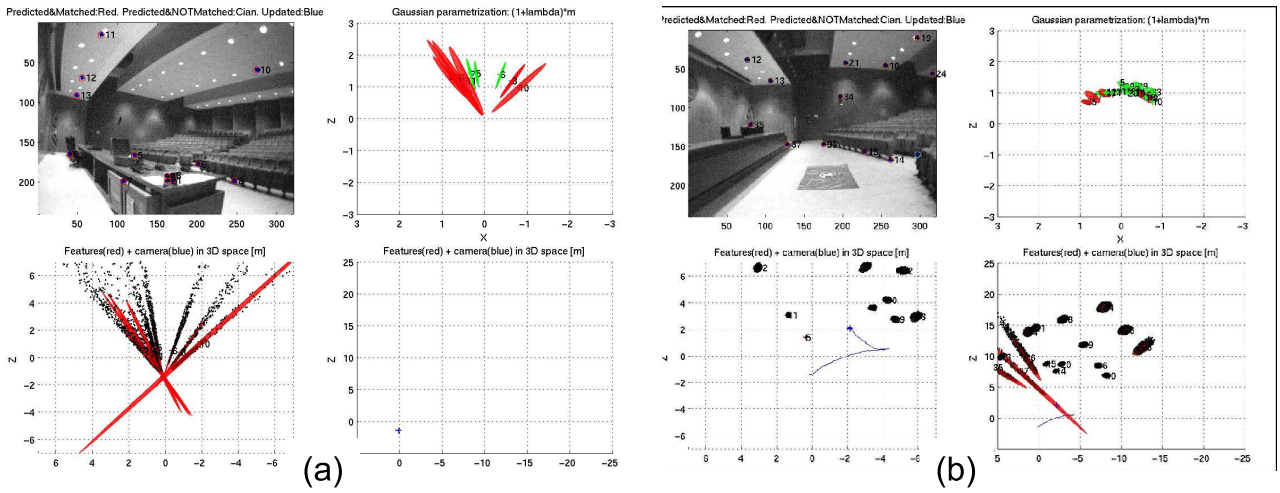

Fig. 4. First (a) and last((b) images of the sequence. To display a map that contains features at very different depths, two top views at different scales are plotted. The top view plotted at bottom left subfigure displays the close features; the top view plotted at the bottom right subfigure displays the distant features. Both top views compare our inverse depth Gaussian parametrization with the standard XYX Gaussian parametrization by the comparison of their uncertainty regions. The Gaussian inverse depth acceptance regions are plotted in XYZ as a cloud of black dots numerically propagated from the Gaussian 6 dimensional superellipsoidal acceptance region coded in inverse depth. The standard Gaussian XYZ acceptance ellipsoids are linearly propagated from the 6 dimensional Gaussian coded in inverse depth by means of the Jacobian. The camera trajectory and its uncertainty is shown in blue. At the initial step (a), most the features are at low parallax. At the final step(b), parallax enough has been gathered for the majority of the features and the feature uncertainty is low.

observed, because of that the search regions for matches are reduced and hence the processing time is reduced. iii) when the features are observed with a moderate parallax, the features can be coded with a dimension 3 XYZ state. So we expect to achieve real time performance at 30 Hz. for reasonable map sizes.

The first experiment, is a 500 frames movie of a lecture theater. The second experiment is 870 frames movie of an outdoors scene where close objects temporarily occlude distant features.

A. *Indoor sequence*

The movie showing the input sequence and the estimation history can be reached at `http://webdiis.unizar.es/%7Ejosemari/in.avi`

The purpose of the experiment was to analyze the performance in an environment with features at different depths. We particularly analyze initialization for three features initialized in the same frame but located at different depths.

Figure 4 shows the image where the analyzed features are initialized (frame 18 in the sequence) and the last image in the sequence; the top view of the map with the feature covariance is plotted as well. To display a map that contains features at very different depths, two top views at different scales are plotted. The top view plotted at bottom left subfigure displays the close features; the top view plotted at the bottom right subfigure displays the distant features. Both top views compare our inverse depth Gaussian parametrization with the standard XYX Gaussian parametrization by the comparison of their uncertainty regions. The Gaussian inverse depth acceptance regions are plotted in XYZ as a cloud of black dots numerically propagated from the Gaussian 6 dimensional superellipsoidal acceptance region coded in inverse depth. The standard Gaussian XYZ acceptance ellipsoids are linearly propagated from the 6 dimensional Gaussian coded in inverse depth by means of the Jacobian.

At the beginning of the sequence, the depth uncertainty is huge, even including the infinity, due to the small translation, no parallax is observed in the features. It is worth noting that Gaussianity in inverse depth is not mapped to a Gaussian in XYZ, so the red ellipsoids are far from representing the XYZ distribution error, especially in depth. As stated by equation (18), is at low parallax when the inverse depth parametrization plays a key role.

As the camera moves, the translation produces parallax, the features depth estimate improves, so in the last image, most of the map features have reduced their uncertainty. As a result the both the uncertainty in XYZ and in inverse depth are Gaussian and the black and the red uncertainty regions become coincident.

Figure 5 focus on the evolution of the estimate corresponding to features 11, 12 and 13 at frames 1, 10, 25, 50, 100 and 200 counted since feature initialization. In top view it is plotted both the XYZ Gaussian uncertainty (red ellipsoid) and the region in inverse depth (black dots); the parallax for each feature at every step is also displayed. When initialized, the ρ Gaussian 95% acceptance region includes $\rho = 0$ so the infinite is considered. The corresponding acceptance region in depth is quite asymmetric, excluding low depths but that extends at high depth down to infinity, and even negative depths corresponding to negative ρ (negative depths are not represented). As rays producing bigger parallax are gathered, the uncertainty in ρ becomes narrower but still maps to a non Gaussian distribution in XYZ. Eventually, both ρ and XYZ

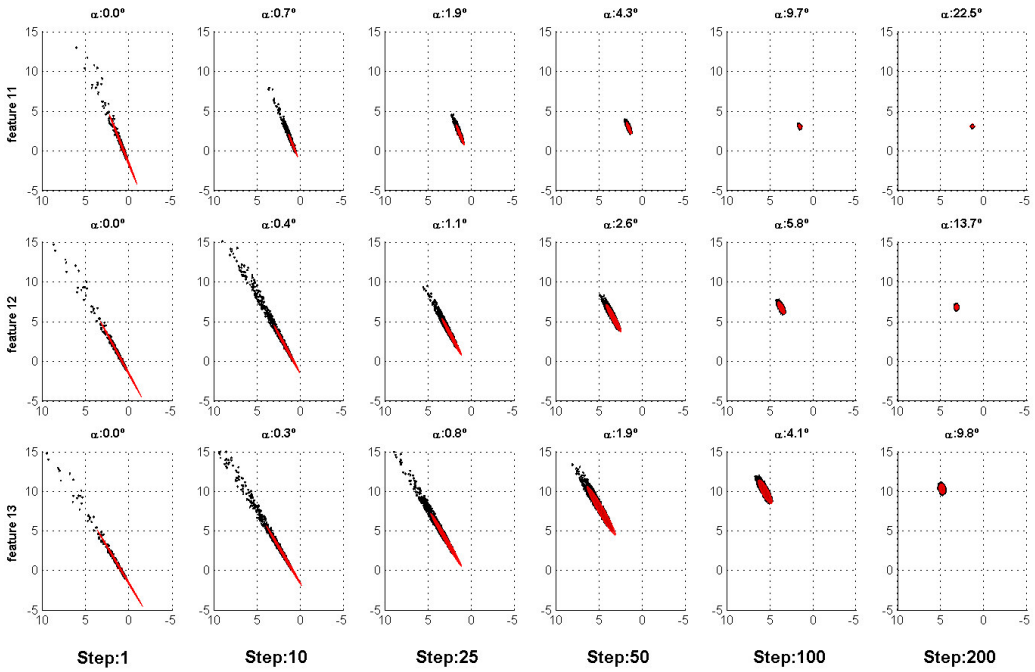

Fig. 5. Feature initialization. Every row shows the evolution of a feature estimation in top view. Per each feature, the estimation after 1, 10, 25, 50, 100 and 200 frames since initialization are plotted; the parallax between the initial observation and the current frame is detailed on top of every subplot. Black dots are a numerical representation for the 95% uncertainty region gaussian in the inverse depth. The red ellipsoid is the uncertainty region coded as Gaussian in XYZ.

regions became both narrow and Gaussian because enough parallax is available.

Let us focus on the distant features. The camera translates after initialization but this translation does not produce parallax because the feature is distant. This information is coded in ρ shifting its value towards zero and narrowing its uncertainty; in the XYZ space this implies having still an asymmetrical acceptance region but that now excludes the low depths. Intuitively, if the camera has translated and no parallax has been detected, then the observed feature cannot be close, so even if the depth cannot be estimated because the feature is distant, some information about its depth has been coded in the estimate.

As the estimation proceeds, when enough parallax is eventually available, the estimation evolves to a narrow Gaussian in ρ that when transformed to XYZ cuts down the probability corresponding to high depths collapsing finally to a Gaussian estimate both in inverse depth and in XYZ.

B. Outdoor sequence

Given the system ability to deal with both close and distant features, it has a nice performance outdoors. The whole experiment sequence along with the estimated map can be reached at `http://webdiis.unizar.es/%7Ejosemari/out.avi`. Figure 6 shows three frames of the movie illustrating the performance. It displays as well the map after processing the whole movie. As in Section V-A, the map represented by two top views at different scales.

Two of the problems that have to be tackled outdoors are distant features and partial occlusion due to the fact that there are objects at quite different depths displaying rather different parallax as the camera moves.

For most of the features, the camera ends up gathering enough parallax to estimate their depth. However, being outdoors, there are rather distant features producing no parallax. It shown how distant features, e.g 24 or 39, in the buildings at the background are persistently tracked along the sequence; however the depth cannot be estimated. The estimation error coded as gaussian in inverse depth is successfully managed by the EKF, and the features behaves as points at infinity. It can be noticed as well the poor error representation if coded as Gaussian in XYZ.

Regarding partial occlusion, The signaled feature in Fig 6, labeled as 36, shows the system ability to reobserve features, from a different point of view after long partial occlusion.

VI. Conclusion

We have presented a parametrization for monocular SLAM which permits operation based uniquely on the standard EKF prediction-update procedure at every step, unifying initialization with the tracking of known features. Our inverse depth parametrization for 3D points allows unified modelling and processing on for any point in the scene, close or distant, or even at 'infinity'. In fact, close, distant or just-initialized features are processed with the routine EKF prediction-update loop without making any binary decisions.

The key factor is that due to the inverse depth parametrization our measurement equation has low linearization error

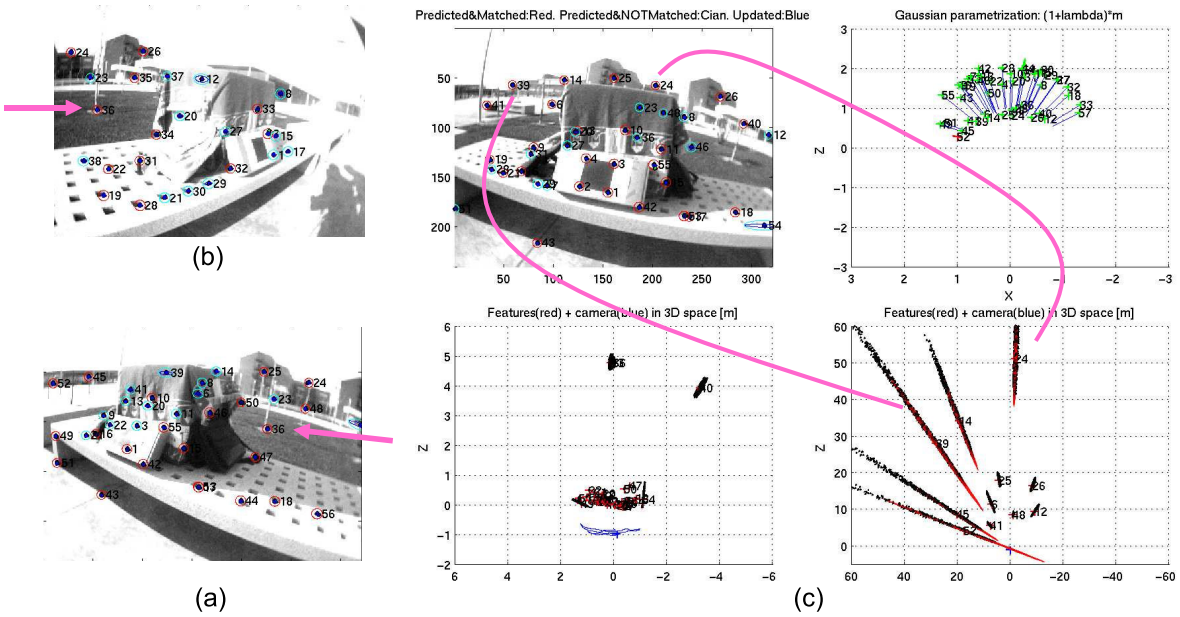

Fig. 6. Subfigures (a) and (b) display frames 197 and 454, showing how scenes with objects at quite different distances are likey to produce partial occlusion. The system can nicely reobserve them after the occlusion as shown in the signaled feature (labeled as 36) on the tree basis. Subfigure (c) Shows the system ability to track successfully distant features along hundreds of frames, being Gaussian in lambda but not Gaussian in XYZ. The lines pairs the image of the features with the top view reconstruction.

at low parallax, and hence the estimation uncertainty is accurately modeled as Gaussian in inverse depth. In Section III-A we presented a simplified model which approximately quantifies the linearization error. It provides a theoretical understanding of the impressive performance of the EKF with the proposed parametrization.

The inverse depth parametrization implies a dimension 6 state vector per feature compared to dimension 3 for Euclidean XYZ coding. This doubles the the size of the map state vector, and hence produces a 4-fold increase in computational cost if all features retain the new parametrization. However, our experiments show that the uncertainties in close feature locations collapse after several frames to accurate Gaussian distributions in Euclidean 3D space, indicating the opportunity to safely convert these features back to an XYZ parametrization and return to dimension 3, meaning that the long-term computational cost would not significantly increase. Further, however, the value of immediate initialization that the new parametrization provides means that right through tracking the amount of uncertainty in the system will be lower (removing jitter from camera pose estimation) and this will lead to computational benefits in terms of smaller search regions and improved image processing speed.

The experiments presented have validated the method with real imagery, using a hand-held camera as the unique sensor both indoors and outdoors. Our current experiments have been run off-line programmed in Matlab, but we are confident in achieving real-time performance in C++ in the near future for numbers of features up to perhaps 100 using current PC hardware — enough to map large rooms or parts of outdoor scenes in practical scenarios.

ACKNOWLEDGMENT

This research was supported by Spanish CICYT DPI2003-07986, EPSRC GR/T24685, EPSRC Advanced Research Fellowship to AJD, and Royal Society International Joint Project grant between U. of Oxford, U. of Zaragoza and Imperial College.

We are very grateful to David Murray, Ian Reid and other members of Oxford's Active Vision Laboratory for discussions and software collaboration.

REFERENCES

[1] A. Chowdhury and R. Chellappa. Stochastic approximation and rate-distortion analysis for robust structure and motion estimation. *IJCV*, 55(1):27–53, 2003.

[2] A. Davison. Real-time simultaneous localization and mapping with a single camera. In *Proc. International Conference on Computer Vision*, 2003.

[3] E. Eade and T. Drummond. Scalable monocular SLAM. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[5] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion i: Algorithm and implementation. *IJCV*, pages 95–117, 1992.

[6] E. Mikhail, J. Bethel, and M. J.C. *Introduction to Modern Photogrammetry*. John Wiley & Sons, 2001.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[8] J. Montiel and A. J. Davison. A visual compass based on SLAM. In *Proc. Intl. Conf. on Robotics and Automation*, 2006(accepted).

[9] J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *IJCV*, 34(2):163–192, 1999.

[10] J. Sola, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only SLAM. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.