Inverse Depth Parametrization for Monocular SLAM

Javier Civera, Andrew J. Davison, and J. M. Martínez Montiel

Abstract—**We present a new parametrization for point features within monocular simultaneous localization and mapping (SLAM) that permits efficient and accurate representation of uncertainty during undelayed initialization and beyond, all within the standard extended Kalman filter (EKF). The key concept is direct parametrization of the inverse depth of features relative to the camera locations from which they were first viewed, which produces measurement equations with a high degree of linearity. Importantly, our parametrization can cope with features over a huge range of depths, even those that 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" smoothly from infinity if the camera makes larger movements. Feature initialization is undelayed in the sense that even distant features are immediately used to improve camera motion estimates, acting initially as bearing references but not permanently labeled as such. The inverse depth parametrization remains well behaved for features at all stages of SLAM processing, but has the drawback in computational terms that each point is represented by a 6-D state vector as opposed to the standard three of a Euclidean *XYZ* representation. We show that once the depth estimate of a feature is sufficiently accurate, its representation can safely be converted to the Euclidean *XYZ* form, and propose a linearity index that allows automatic detection and conversion to maintain maximum efficiency—only low parallax features need be maintained in inverse depth form for long periods. We present a real-time implementation at 30 Hz, where the parametrization is validated in a fully automatic 3-D SLAM system featuring a hand-held single camera with no additional sensing. Experiments show robust operation in challenging indoor and outdoor environments with a very large ranges of scene depth, varied motion, and also real time 360° loop closing.**

Index Terms—**Monocular simultaneous localization and mapping (SLAM), real-time vision.**

I. Introduction

A MONOCULAR camera is a projective sensor that measures the bearing of image features. Given an image sequence of a rigid 3-D scene taken from a moving camera, it is now well known that it is possible to compute both a scene structure and a camera motion up to a scale factor. To infer the 3-D position of each feature, the moving camera must observe it repeatedly each time, capturing a ray of light from the feature to its optic center. The measured angle between the captured rays from different viewpoints is the feature's *parallax*—this is what allows its depth to be estimated.

In offline "structure from motion (SFM)" solutions from the computer vision literature (e.g., [11] and [23]), motion and structure are estimated from an image sequence by first applying a robust feature matching between pairs or other short overlapping sets of images to estimate relative motion. An optimization procedure then iteratively refines global camera location and scene feature position estimates such that features project as closely as possible to their measured image positions (bundle adjustment). Recently, work in the spirit of these methods, but with "sliding window" processing and refinement rather than global optimization, has produced impressive real-time "visual odometry" results when applied to stereo sequences in [21] and for monocular sequences in [20].

An alternative approach to achieving real-time motion and structure estimation are online visual simultaneous localization and mapping (SLAM) approaches that use a probabilistic

Manuscript received February 27, 2007; revised September 28, 2007. First published October 3, 2008; current version published October 31, 2008. This work was supported in part by the Spanish Grant PR2007-0427, Grant DPI2006-13578, and Grant DGA(CONSI+D)-CAI IT12-06, in part by the Engineering and Physical Sciences Research Council (EPSRC) Grant GR/T24685, in part by the Royal Society International Joint Project grant between the University of Oxford, University of Zaragoza, and Imperial College London, and in part by the Robotics Advancement through Web-Publishing of Sensorial and Elaborated Extensive Datasets (RAWSEEDS) under Grant FP6-IST-045144. The work of A. J. Davison was supported the EPSRC Advanced Research Fellowship Grant. This paper was recommended for publication by Associate Editor P. Rives and Editor L. Parker upon evaluation of the reviewers' comments.

J. Civera and J. M. Martínez Montiel are with the Departamento de Informática, University of Zaragoza, 50018 Zaragoza, Spain (e-mail: josemari@unizar.es; jcivera@unizar.es).

A. J. Davison is with the Department of Computing, Imperial College London, SW7 2AZ London, U.K. (e-mail: ajd@doc.ic.ac.uk).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the author. This material includes the following video files. *inverseDepth_indoor.avi* (11.7 MB) shows simultaneous localization and mapping, from a hand-held camera observing an indoor scene. All the processing is automatic, the image sequence being the only sensorial information used as input. It is shown as a top view of the computed camera trajectory and 3-D scene map. Image sequence is acquired with a hand-held camera 320 £ 240 at 30 frames/second. *Player information* XviD MPEG-4. *inverseDepth_outdoor.avi* (12.4 MB) shows real-time simultaneous localization and mapping, from a hand-held camera observing an outdoor scene, including rather distant features. All the processing is automatic, the image sequence being the only sensorial information used as input. It is shown as a top view of the computed camera trajectory and 3-D scene map. Image sequence is acquired with a hand-held camera 320 £ 240 at 30 frames/second. The processing is done with a standard laptop. *Player information* XviD MPEG-4. *inverseDepth_loopClosing.avi* (10.2MB) shows simultaneous localization and mapping, from a hand-held camera observing a loop-closing indoor scene. All the processing is automatic, the image sequence being the only sensorial information used as input. It is shown as a top view of the computed camera trajectory and 3-D scene map. Image sequence is acquired with a hand-held camera 320 £ 240 at 30 frames/second. *Player information* XviD MPEG-4. *inverseDepth_loopClosing_ID_to_XYZ_conversion.avi* (10.1 MB) shows simultaneous localization and mapping, from a hand-held camera observing the same loop-closing indoor sequence as in inverseDepth loopClosing.avi, but switching from inverse depth to *XYZ* parameterization when necessary. All the processing is automatic, the image sequence being the only sensorial information used as input. It is shown as a top view of the computed camera trajectory and 3-D scene map. Image sequence is acquired with a hand-held camera 320 £ 240 at 30 frames/second. *Player information* XviD MPEG-4. *inverseDepth_indoorRawImages.tar.gz* (44 MB) shows indoor sequence raw images in .pgm format. Camera calibration in an ASCII file. *inverseDepth_outdoorRawImages.tar.gz* (29 MB) shows outdoor sequence raw images in .pgm format. Camera calibration in an ASCII file. *inverseDepth_loopClosingRawImages.tar.gz* (33 MB) shows loop-closing sequence raw images in .pgm format. Camera calibration in an ASCII file. *Contact information* jcivera@unizar.es; josemari@unizar.es.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2008.2003276

filtering approach to sequentially update estimates of the positions of features (the map) and the current location of the camera. These SLAM methods have different strengths and weaknesses to visual odometry, being able to build consistent and drift-free global maps, but with a bounded number of mapped features. The core single extended Kalman filter (EKF) SLAM technique, previously proven in multisensor robotic applications, was first applied successfully to real-time monocular camera tracking by Davison *et al.* [8], [9] in a system that built sparse room-sized maps at 30 Hz.

A significant limitation of Davison's and similar approaches, however, was that they could only make use of features that were close to the camera relative to its distance of translation, and therefore exhibited significant parallax during motion. The problem was in initializing uncertain depth estimates for distant features: in the straightforward Euclidean *XYZ* feature parametrization adopted, position uncertainties for low parallax features are not well represented by the Gaussian distributions implicit in the EKF. The depth coordinate of such features has a probability density that rises sharply at a well-defined minimum depth to a peak, but then, tails off very slowly toward infinity—from low parallax measurements, it is very difficult to tell whether a feature has a depth of 10 units rather than 100, 1000, or more. For the rest of the paper, we refer to Euclidean *XYZ* parametrization simply as *XYZ*.

There have been several recent methods proposed for coping with this problem, relying on generally undesirable special treatment of newly initialized features. In this paper, we describe a new feature parametrization that is able to smoothly cope with initialization of features at all depths—even up to "infinity"—within the standard EKF framework. The key concept is direct parametrization of inverse depth relative to the camera position from which a feature was first observed.

A. Delayed and Undelayed Initialization

The most obvious approach to coping with feature initialization within a monocular SLAM system is to treat newly detected features separately from the main map, accumulating information in a special processing over several frames to reduce depth uncertainty before insertion into the full filter with a standard *XYZ* representation. Such *delayed initialization* schemes (e.g., [3], [8], and [14]) have the drawback that new features, held outside the main probabilistic state, are not able to contribute to the estimation of the camera position until finally included in the map. Further, features that retain low parallax over many frames (those very far from the camera or close to the motion epipole) are usually rejected completely because they never pass the test for inclusion.

In the delayed approach of Bailey [2], initialization is delayed until the measurement equation is approximately Gaussian and the point can be safely triangulated; here, the problem was posed in 2-D and validated in simulation. A similar approach for a 3-D monocular vision with inertial sensing was proposed in [3]. Davison [8] reacted to the detection of a new feature by inserting a 3-D semiinfinite ray into the main map representing everything about the feature except its depth, and then, used an auxiliary particle filter to explicitly refine the depth estimate over several frames, taking advantage of all the measurements in a high frame rate sequence, but again with new features held outside the main state vector until inclusion.

More recently, several *undelayed initialization* schemes have been proposed, which still treat new features in a special way but are able to benefit immediately from them to improve camera motion estimates—the key insight being that while features with highly uncertain depths provide little information on camera translation, they are extremely useful as bearing references for orientation estimation. The undelayed method proposed by Kwok and Dissanayake [15] was a multiple hypothesis scheme, initializing features at various depths and pruning those not reobserved in subsequent images.

Sola *et al.* [24], [25] described a more rigorous undelayed approach using a Gaussian sum filter approximated by a federated information sharing method to keep the computational overhead low. An important insight was to spread the Gaussian depth hypotheses along the ray according to inverse depth, achieving much better representational efficiency in this way. This method can perhaps be seen as the direct stepping stone between Davison's particle method and our new inverse depth scheme; a Gaussian sum is a more efficient representation than particles (efficient enough that the separate Gaussians can all be put into the main state vector), but not as efficient as the single Gaussian representation that the inverse depth parametrization allows. Note that neither [15] nor [25] considers features at very large "infinite" depths.

B. Points at Infinity

A major motivation of the approach in this paper is not only the efficient undelayed initialization, but also the desire to cope with features at *all* depths, particularly in outdoor scenes. In SFM, the well-known concept of a point at infinity is a feature that 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 that 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 used normally in SFM allow explicit representation of points at infinity, and they have proven to play an important role during offline structure and motion estimation.

In a sequential SLAM system, the difficulty is that we do not know in advance which features are infinite and which are not. Montiel and Davison [19] 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, let us imagine a camera moving through a 3-D 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 centimeters 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 labeled 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 the uncertainty in depth of seemingly infinite features. Observing no parallax for a feature after 10 units of camera translation does tell us something about its depth—it gives a reliable lower bound, which depends on the amount of motion made by the camera (if the feature had been closer than this, we *would* have observed parallax). This explicit consideration of uncertainty in the locations of points has not been previously required in offline computer vision algorithms, but is very important in a more difficult online case.

C. Inverse Depth Representation

Our contribution is to show that, in fact, there is a unified and straightforward parametrization for feature locations that can handle both initialization and standard tracking of both close and very distant features within the standard EKF framework. An explicit parametrization of the *inverse depth* of a feature along a semiinfinite ray from the position from which it was first viewed allows a Gaussian distribution to cover uncertainty in depth that spans a depth range from nearby to infinity, and permits seamless crossing over to finite depth estimates of features that have been apparently infinite for long periods of time. The unified representation means that our algorithm requires no special initialization 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 modeled. Note that our parameterization would be equally compatible with other variants of Gaussian filtering such as sparse information filters.

We introduce a linearity index and use it to analyze and prove the representational capability of the inverse depth parametrization for both low and high parallax features. The only drawback of the inverse depth scheme is the computational issue of increased state vector size since an inverse depth point needs six parameters rather than the three of *XYZ* coding. As a solution to this, we show that our linearity index can also be applied to the *XYZ* parametrization to signal when a feature can be safely switched from inverse depth to *XYZ*; the usage of the inverse depth representation can, in this way, be restricted to low parallax feature cases where the *XYZ* encoding departs from Gaussianity. Note that this "switching," unlike in delayed initialization methods, is purely to reduce computational load; SLAM accuracy with or without switching is almost the same.

The fact is that the projective nature of a camera means that the image measurement process is nearly linear in this inverse depth coordinate. Inverse depth is a concept used widely in computer vision: it appears in the relation between image disparity and point depth in a stereo vision; it is interpreted as the parallax with respect to the plane at infinity in [12]. Inverse depth is also used to relate the motion field induced by scene points with the camera velocity in optical flow analysis [13]. In the tracking community, "modified polar coordinates" [1] also exploit the linearity properties of the inverse depth representation in a slightly different, but closely related, problem of a target motion analysis (TMA) from measurements gathered by a bearing-only sensor with known motion.

However, the inverse depth idea has not previously been properly integrated in sequential, probabilistic estimation of motion, and structure. It has been used in EKF-based sequential depth estimation from camera-known motion [16], and in a multibaseline stereo, Okutomi and Kanade [22] used the inverse depth to increase matching robustness for scene symmetries; matching scores coming from multiple stereo pairs with different baselines were accumulated in a common reference coded in inverse depth, this paper focusing on matching robustness and not on probabilistic uncertainty propagation. Chowdhury and Chellappa [5] proposed a sequential EKF process using inverse depth, but this was in some way short of full SLAM in its details. Images are first processed pairwise to obtain a sequence of 3-D motions that are then fused with an individual EKF per feature.

It is our parametrization of inverse depth *relative to the positions from which features were first observed*, which means that a Gaussian representation is uniquely well behaved, this is the reason why a straightforward parametrization of monocular SLAM in the homogeneous coordinates of SFM will not give a good result—that representation only meaningfully represents points that appear to be infinite relative to the coordinate origin. It could be said in projective terms that our method defines separate but correlated projective frames for each feature. Another interesting comparison is between our method, where the representation for each feature includes the camera position from which it was first observed, and smoothing/full SLAM schemes, where all historical sensor pose estimates are maintained in a filter.

Two recently published papers from other authors have developed methods that are quite similar to ours. Trawny and Roumeliotis [26] proposed an undelayed initialization for 2-D monocular SLAM that encodes a map point as the intersection of two projection rays. This representation is overparametrized but allows undelayed initialization and encoding of both close and distant features, the approach validated with simulation results.

Eade and Drummond presented an inverse depth initialization scheme within the context of their FastSLAM-based system for monocular SLAM [10], offering some of the same arguments about advantages in linearity as in our paper. The position of each new partially initialized 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 initialized standard *XYZ* representation. While retaining the differentiation between partially and fully initialized features, they go further and are able to use measurements of partially initialized features with unknown depth to improve estimates of camera orientation and translation 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 initialized need not be explicitly tackled and full use is automatically made of all of the information available in measurements.

This paper offers a comprehensive and extended version of our work previously published as two conference papers [7], [18]. We now present a full real-time implementation of the inverse depth parameterization that can map up to 50–70 features in real time on a standard laptop computer. Experimental validation has shown the important role of an accurate camera calibration to improve the system performance, especially with wide-angle cameras. Our results section includes new real-time experiments, including the key result of vision-only loop closing. Input test image sequences and movies showing the computed solution are included in the paper as multimedia material.

Section II is devoted to defining the state vector, including the camera motion model, *XYZ* point coding, and inverse depth point parametrization. The measurement equation is described in Section III. Section IV presents a discussion about measurement equation linearization errors. Next, feature initialization from a single-feature observation is detailed in Section V. In Section VI, the switch from inverse depth to *XYZ* coding is presented, and in Section VII, we present experimental validations over real-image sequences captured at 30 Hz in large-scale environments, indoors and outdoors, including real-time performance, and a loop closing experiment; links to movies showing the system performance are provided. Finally, Section VIII is devoted to conclusions.

II. State Vector Definition

A. Camera Motion

A constant angular and linear velocity model is used to model handheld camera motion. The camera state $\mathbf{x}_v$ is composed of pose terms: $\mathbf{r}^{WC}$ camera optical center position and $\mathbf{q}^{WC}$ quaternion defining orientation, and linear and angular velocity $\mathbf{v}^W$ and ω^C relative to world frame W and camera frame C, respectively.

We assume that linear and angular accelerations $\mathbf{a}^W$ and α^C affect the camera, producing at each step, an impulse of linear velocity $\mathbf{V}^W = \mathbf{a}^W \Delta t$ and angular velocity $\Omega^C = \alpha^C \Delta t$, with zero mean and known Gaussian distribution. We currently assume a diagonal covariance matrix for the unknown input linear and angular accelerations.

The state update equation for the camera is

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

where $\mathbf{q}((\omega^C_k + \Omega^C)\Delta t)$ is the quaternion defined by the rotation vector $(\omega^C_k + \Omega^C)\Delta t$.

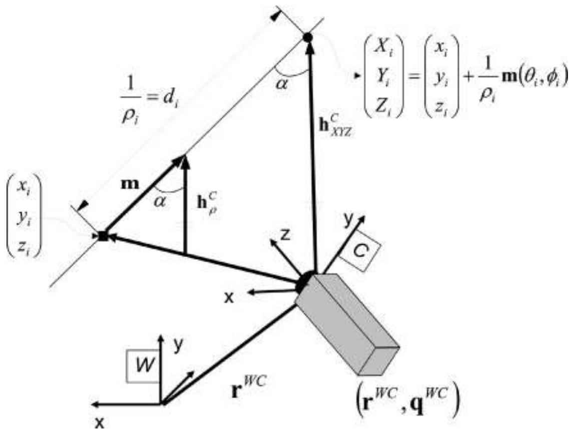
$\frac{1}{\rho_i} = d_i$
α
$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}(\theta_i, \phi_i)$
$\mathbf{h}^C_{XYZ}$
$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$
$\mathbf{m}$
α
$\mathbf{h}^C_\rho$
z
y
C
x
W
y
x
$\mathbf{r}^{WC}$
$(\mathbf{r}^{WC}, \mathbf{q}^{WC})$

Fig. 1. Feature parametrization and measurement equation.

B. Euclidean XYZ Point Parametrization

The standard representation for scene points i in terms of Euclidean *XYZ* coordinates (see Fig. 1) is

$$\mathbf{x}_i = \begin{pmatrix} X_i & Y_i & Z_i \end{pmatrix}^\top. \tag{2}$$

In this paper, we refer to the Euclidean *XYZ* coding simply as *XYZ* coding.

C. Inverse Depth Point Parametrization

In our new scheme, a scene 3-D point i can be defined by the 6-D state vector:

$$\mathbf{y}_i = \begin{pmatrix} x_i & y_i & z_i & \theta_i & \phi_i & \rho_i \end{pmatrix}^\top \tag{3}$$

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

$$\mathbf{x}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right) \tag{4}$$

$$\mathbf{m} = \left(\cos\,\phi_i \sin\,\theta_i, -\sin\,\phi_i, \cos\,\phi_i \cos\,\theta_i\right)^\top. \tag{5}$$

The $\mathbf{y}_i$ vector encodes the *ray from the first camera position from which the feature was observed* by x_i, y_i, z_i, the camera optical center, and θ_i, ϕ_i azimuth and elevation (coded in the world frame) defining unit directional vector $\mathbf{m}\left(\theta_i, \phi_i\right)$. The point's depth along the ray d_i is encoded by its inverse $\rho_i = 1/d_i$.

D. Full State Vector

As in standard EKF SLAM, we use a single-joint state vector containing camera pose and feature estimates, with the assumption that the camera moves with respect to a static scene. The whole state vector $\mathbf{x}$ is composed of the camera and all the map features

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

III. Measurement Equation

Each observed feature imposes a constraint between the camera location and the corresponding map feature (see Fig. 1).

Observation of a point $\mathbf{y}_i(\mathbf{x}_i)$ defines a ray coded by a directional vector in the camera frame $\mathbf{h}^C = \begin{pmatrix} h_x & h_y & h_z \end{pmatrix}^\top$. For points in *XYZ*

$$\mathbf{h}^C = \mathbf{h}^C_{XYZ} = \mathbf{R}^{CW}\left(\begin{matrix} X_i \\ Y_i \\ Z_i \end{matrix} - \mathbf{r}^{WC}\right). \tag{7}$$

For points in inverse depth

$$\mathbf{h}^C = \mathbf{h}^C_{\rho} = \mathbf{R}^{CW}\left(\rho_i\left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \mathbf{r}^{WC}\right) + \mathbf{m}\left(\theta_i, \phi_i\right)\right) \tag{8}$$

where the directional vector has been normalized using the inverse depth. It is worth noting that (8) can be safely used even for points at infinity, i.e., $\rho_i = 0$.

The camera does not directly observe $\mathbf{h}^C$ but its projection in the image according to the pinhole model. Projection to a normalized retina, and then, camera calibration is applied:

$$\mathbf{h} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - \dfrac{f}{d_x}\dfrac{h_x}{h_z} \\ v_0 - \dfrac{f}{d_y}\dfrac{h_y}{h_z} \end{pmatrix} \tag{9}$$

where u_0, v_0 is the camera's principal point, f is the focal length, and d_x, d_y is the pixel size. Finally, a distortion model has to be applied to deal with real camera lenses. In this paper, we have used the standard two parameters distortion model from photogrammetry [17] (see the Appendix for details).

It is worth noting that the measurement equation in inverse depth has a sensitive dependency on the parallax angle α (see Fig. 1). At low parallax, (8) can be approximated by $\mathbf{h}^C \approx \mathbf{R}^{CW}\left(\mathbf{m}\left(\theta_i, \phi_i\right)\right)$, and hence, the measurement equation only provides information about the camera orientation and the directional vector $\mathbf{m}\left(\theta_i, \phi_i\right)$.

IV. Measurement Equation Linearity

The more linear the measurement equation is, the better a Kalman filter performs. This section is devoted to presenting an analysis of measurement equation linearity for both *XYZ* and inverse depth codings. These linearity analyses theoretically support the superiority of the inverse depth coding.

A. Linearized Propagation of a Gaussian

Let x be an uncertain variable with Gaussian distribution $x \sim N\left(\mu_x, \sigma_x^2\right)$. The transformation of x through the function f is a variable y that can be approximated with Gaussian distribution:

$$y \sim N\left(\mu_y, \sigma_y^2\right),\ \mu_y = f\left(\mu_x\right),\ \sigma_y^2 = \left.\frac{\partial f}{\partial x}\right|_{\mu_x} \sigma_x^2 \left.\frac{\partial f}{\partial x}\right|_{\mu_x}^\top \tag{10}$$

if the function f is linear in an interval around μ_x (Fig. 2). The interval size in which the function has to be linear depends on σ_x; the bigger σ_x the wider the interval has to be to cover a significant fraction of the random variable x values. In this paper, we fix the linearity interval to the 95% confidence region defined by $[\mu_x - 2\sigma_x, \mu_x + 2\sigma_x]$.

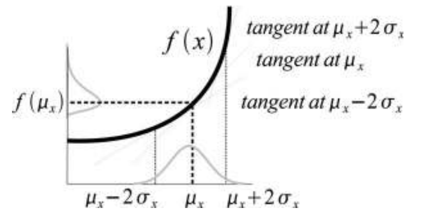

Fig. 2. First derivative variation in $[\mu_x - 2\sigma_x, \mu_x + 2\sigma_x]$ codes the departure from Gaussianity in the propagation of the uncertain variable through a function.

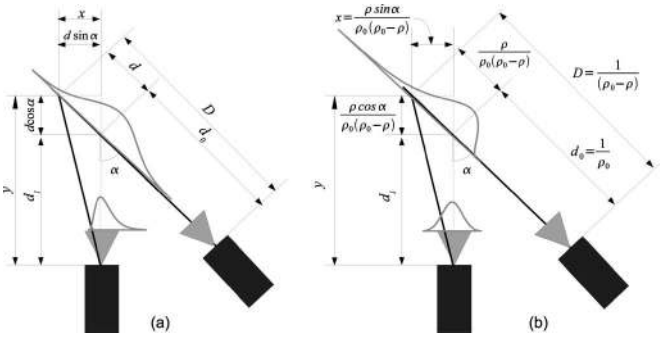
x
$d\sin\alpha$
d
D
d_0
$d\cos\alpha$
y
d_1
α
(a)
$x=\dfrac{\rho\sin\alpha}{\rho_0(\rho_0-\rho)}$
$\dfrac{\rho}{\rho_0(\rho_0-\rho)}$
$D=\dfrac{1}{(\rho_0-\rho)}$
$\dfrac{\rho\cos\alpha}{\rho_0(\rho_0-\rho)}$
$d_0=\dfrac{1}{\rho_0}$
y
d_1
α
(b)

Fig. 3. Uncertainty propagation from the scene point to the image. (a) *XYZ* coding. (b) Inverse depth coding.

If a function is linear in an interval, the first derivative is constant in that interval. To analyze the first derivative variation around the interval $[\mu_x - 2\sigma_x, \mu_x + 2\sigma_x]$, consider the Taylor expansion for the *first derivative*:

$$\frac{\partial f}{\partial x}\left(\mu_x + \Delta x\right) \approx \left.\frac{\partial f}{\partial x}\right|_{\mu_x} + \left.\frac{\partial^2 f}{\partial x^2}\right|_{\mu_x} \Delta x. \tag{11}$$

We propose to compare the value of the derivative at the interval center μ_x with the value at the extremes $\mu_x \pm 2\sigma_x$, where the deviation from linearity will be maximal, using the following dimensionless linearity index:

$$L = \left|\frac{\left.\dfrac{\partial^2 f}{\partial x^2}\right|_{\mu_x} 2\sigma_x}{\left.\dfrac{\partial f}{\partial x}\right|_{\mu_x}}\right|. \tag{12}$$

When $L \approx 0$, the function can be considered linear in the interval, and hence, Gaussianity is preserved during transformation.

B. Linearity of XYZ Parametrization

The linearity of the *XYZ* representation is analyzed by means of a simplified model that only estimates the depth of a point with respect to the camera. In our analysis, a scene point is observed by two cameras [Fig. 3(a)], both of which are oriented toward the point. The first camera detects the ray on which the point lies. The second camera observes the same point from a distance d_1; the parallax angle α is approximated by the angle between the cameras' optic axes.

The point's location error d is encoded as Gaussian in depth

$$D = d_0 + d, \qquad d \sim N\left(0, \sigma_d^2\right). \tag{13}$$

This error d is propagated to the image of the point in the second camera u as

$$u = \frac{x}{y} = \frac{d \sin \alpha}{d_1 + d \cos \alpha}. \tag{14}$$

The Gaussianity of u is analyzed by means of (12), giving the following linearity index:

$$L_d = \left| \frac{(\partial^2 u / \partial d^2) 2\sigma_d}{\partial u / \partial d} \right| = \frac{4\sigma_d}{d_1} \left| \cos \alpha \right|. \tag{15}$$

C. Linearity of Inverse Depth Parametrization

The inverse depth parametrization is based on the same scene geometry as the direct depth coding, but the depth error is encoded as Gaussian in inverse depth [Fig. 3(b)]:

$$D = \frac{1}{\rho_0 - \rho}, \qquad \rho \sim N\left(0, \sigma_\rho^2\right) \tag{16}$$

$$d = D - d_0 = \frac{\rho}{\rho_0 \left(\rho_0 - \rho\right)} \qquad d_0 = \frac{1}{\rho_0}. \tag{17}$$

So, the image of the scene point is computed as

$$u = \frac{x}{y} = \frac{d \sin \alpha}{d_1 + d \cos \alpha} = \frac{\rho \sin \alpha}{\rho_0 d_1 \left(\rho_0 - \rho\right) + \rho \cos \alpha} \tag{18}$$

and the linearity index L_ρ is now

$$L_\rho = \left| \frac{(\partial^2 u / \partial \rho^2) 2\sigma_\rho}{\partial u / \partial \rho} \right| = \frac{4\sigma_\rho}{\rho_0} \left| 1 - \frac{d_0}{d_1} \cos \alpha \right|. \tag{19}$$

D. Depth Versus Inverse Depth Comparison

When a feature is initialized, the depth prior has to cover a vast region in front of the camera. With the inverse depth representation, the 95% confidence region with parameters ρ_0, σ_ρ is

$$\left[\frac{1}{\rho_0 + 2\sigma_\rho}, \frac{1}{\rho_0 - 2\sigma_\rho} \right]. \tag{20}$$

This region cannot include zero depth but can easily extend to infinity.

Conversely, with the depth representation, the 95% region with parameters d_0, σ_d is $[d_0 - 2\sigma_d, d_0 + 2\sigma_d]$. This region can include zero depth but cannot extend to infinity.

In the first few frames, after a new feature has been initialized, little parallax is likely to have been observed. Therefore, $d_0/d_1 \approx 1$ and $\alpha \approx 0 \Longrightarrow \cos \alpha \approx 1$. In this case, the L_d linearity index for depth is high (bad), while the L_ρ linearity index for inverse depth is low (good): during initialization, the inverse depth measurement equation linearity is superior to the *XYZ* coding.

As estimation proceeds and α increases, leading to more accurate depth estimates, the inverse depth representation continues to have a high degree of linearity. This is because in the expression for L_ρ, the increase in the term $|1 - (d_0/d_1) \cos \alpha|$ is compensated by the decrease in $4\sigma_\rho/\rho_0$. For inverse depth features, a good linearity index is achieved along the whole estimation history. So, the inverse depth coding is suitable for both low and high parallax cases if the feature is continuously observed.

The *XYZ* encoding has low computational cost, but achieves linearity only at low depth uncertainty and high parallax. In Section VI, we explain how the representation of a feature can be switched over such that the inverse depth parametrization is only used when needed—for features that are either just initialized or at extreme depths.

V. Feature Initialization

From just a single observation, no feature depth can be estimated (although it would be possible in principle to impose a very weak depth prior by knowledge of the type of scene observed). What we do is to assign a general Gaussian prior in inverse depth that encodes probabilistically the fact that the point has to be in front of the camera. Hence, due to the linearity of inverse depth at low parallax, the filter can be initialized from just one observation. Experimental tuning has shown that infinity should be included with reasonable probability within the initialization prior, despite the fact that this means that depth estimates can become negative. Once initialized, features are processed with the standard EKF prediction-update loop—even in the case of negative inverse depth estimates—and immediately contribute to camera location estimation within SLAM.

It is worth noting that while a feature retains low parallax, it will automatically be used mainly to determine the camera orientation. The feature's depth will remain uncertain with the hypothesis of infinity still under consideration (represented by the probability mass corresponding to negative inverse depths). If the camera translates to produce enough parallax, then the feature's depth estimation will be improved and it will begin to contribute more to the camera location estimation.

The initial location for a newly observed feature inserted into the state vector is

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

a function of the current camera pose estimate $\hat{\mathbf{r}}^{WC}, \hat{\mathbf{q}}^{WC}$, the image observation $\mathbf{h} = \left(u \quad v \right)^\top$, and the parameters determining the depth prior ρ_0, σ_ρ.

The endpoint of the initialization ray (see Fig. 1) is taken from the current camera location estimate

$$\left(\hat{x}_i \quad \hat{y}_i \quad \hat{z}_i\right)^\top = \hat{\mathbf{r}}^{WC} \tag{22}$$

and the direction of the ray is computed from the observed point, expressed in the world coordinate frame

$$\mathbf{h}^W = \mathbf{R}_{WC}\left(\mathbf{q}^{\hat{W}C}\right)\left(\upsilon \quad \nu \quad 1\right)^\top \tag{23}$$

where υ and ν are normalized retina image coordinates. Despite $\mathbf{h}^W$ being a nonunit directional vector, the angles by which we parametrize its direction can be calculated as

$$\begin{pmatrix} \theta_i \\ \phi_i \end{pmatrix} = \begin{pmatrix} \arctan\left(\mathbf{h}_x^W, \mathbf{h}_z^W\right) \\ \arctan\left(-\mathbf{h}_y^W, \sqrt{\mathbf{h}_x^{W\,2} + \mathbf{h}_z^{W\,2}}\right) \end{pmatrix}. \tag{24}$$

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

The initial value for ρ_0 and its standard deviation are set empirically such that the 95% confidence region spans a range of depths from close to the camera up to infinity. In our experiments, we set $\hat{\rho}_0 = 0.1, \sigma_\rho = 0.5$, which gives an inverse depth confidence region $[1.1, -0.9]$. Notice that infinity is included in this range. Experimental validation has shown that the precise values of these parameters are relatively unimportant to the accurate operation of the filter as long as infinity is clearly included in the confidence interval.

The state covariance after feature initialization is

$$\hat{\mathbf{P}}_{k|k}^{\text{new}} = \mathbf{J} \begin{pmatrix} \hat{\mathbf{P}}_{k|k} & 0 & 0 \\ 0 & \mathbf{R}_i & 0 \\ 0 & 0 & \sigma_\rho^2 \end{pmatrix} \mathbf{J}^\top \tag{25}$$

$$\mathbf{J} = \left(\begin{array}{c|c} I & 0 \\ \hline \frac{\partial \mathbf{y}}{\partial \mathbf{r}^{WC}}, \frac{\partial \mathbf{y}}{\partial \mathbf{q}^{WC}}, 0, \ldots, 0, & \frac{\partial \mathbf{y}}{\partial \mathbf{h}}, \frac{\partial \mathbf{y}}{\partial \rho} \end{array} \right). \tag{26}$$

The inherent scale ambiguity in a monocular SLAM has usually been fixed by observing some known initial features that fix the scale (e.g., [8]). A very interesting experimental observation we have made using the inverse depth scheme is that sequential monocular SLAM can operate successfully without *any* known features in the scene, and in fact, the experiments we present in this paper do not use an initialization target. In this case, of course, the overall scale of the reconstruction and camera motion is undetermined, although with the formulation of the current paper, the estimation will settle on a (meaningless) scale of some value. In a very recent work [6], we have investigated this issue with a new dimensionless formulation of monocular SLAM.

VI. Switching From Inverse Depth to *XYZ*

While the inverse depth encoding can be used at both low and high parallax, it is advantageous for reasons of computational efficiency to restrict inverse depth to cases where the *XYZ* encoding exhibits nonlinearity according to the L_d index. This section details switching from inverse depth to *XYZ* for high parallax features.

A. Conversion From Inverse Depth to XYZ Coding

After each estimation step, the linearity index L_d (15) is computed for every map feature coded in inverse depth

$$\mathbf{h}_{XYZ}^W = \hat{\mathbf{x}}_i - \hat{\mathbf{r}}^{WC} \qquad \sigma_d = \frac{\sigma_\rho}{\rho_i^2} \qquad \sigma_\rho = \sqrt{\mathbf{P}_{\mathbf{y}_i\mathbf{y}_i}(6,6)}$$
$$d_i = \left\|\mathbf{h}_{XYZ}^W\right\| \qquad \cos\alpha = \mathbf{m}^\top \mathbf{h}_{XYZ}^W \left\|\mathbf{h}_{XYZ}^W\right\|^{-1}. \tag{27}$$

where $\hat{\mathbf{x}}_i$ is computed using (4) and $\mathbf{P}_{\mathbf{y}_i\mathbf{y}_i}$ is the submatrix 6×6 covariance matrix corresponding to the considered feature.

If L_d is below a switching threshold, the feature is transformed using (4) and the *full state* covariance matrix $\mathbf{P}$ is transformed with the corresponding Jacobian:

$$\mathbf{P}_{\text{new}} = \mathbf{J}\mathbf{P}\mathbf{J}^\top \qquad \mathbf{J} = \text{diag}\left(\mathbf{I}, \frac{\partial \mathbf{x}_i}{\partial \mathbf{y}_i}, \mathbf{I}\right). \tag{28}$$

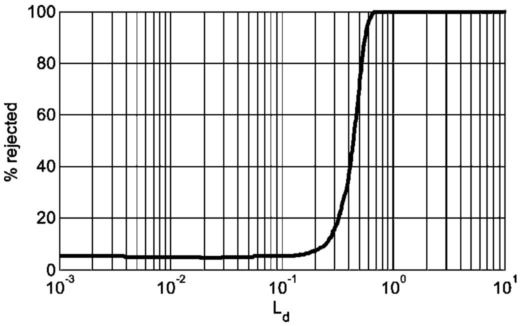

Fig. 4. Percentage of test rejections as a function of the linearity index L_d.

B. Linearity Index Threshold

We propose to use index L_d (15) to define a threshold for switching from inverse depth to *XYZ* encoding at the point when the latter can be considered linear. If the *XYZ* representation is linear, then the measurement u is Gaussian distributed (10), i.e.,

$$u \sim N\left(\mu_u, \sigma_u^2\right) \qquad \mu_u = 0 \qquad \sigma_u^2 = \left(\frac{\sin\alpha}{d_1}\right)^2 \sigma_d^2. \tag{29}$$

To determine the threshold in L_d that signals a lack of linearity in the measurement equation, a simulation experiment has been performed. The goal was to generate samples from the uncertain distribution for variable u, and then, apply a standard Kolmogorov–Smirnov Gaussianty [4] test to these samples, counting the percentage of rejected hypotheses h. When u is effectively Gaussian, the percentage should match the test significance level α_{sl} (5% in our experiments); as the number of rejected hypotheses increases, the measurement equation departs from linearity. A plot of the percentage of rejected hypotheses h with respect to the linearity index L_d is shown in Fig. 4. It can be clearly seen than when $L_d > 0.2$, h sharply departs from 5%. So, we propose the $L_d < 10\%$ threshold for switching from inverse depth to *XYZ* encoding.

Notice that the plot in Fig. 4 is smooth (log scale in L_d), which indicates that the linearity index effectively represents the departure from linearity.

The simulation has been performed for a variety of values of α, d_1, and σ_d; more precisely, all triplets resulting from the following parameter values:

$$\alpha(\text{deg}) \in \{0.1, 1, 3, 5, 7, 10, 20, 30, 40, 50, 60, 70\}$$
$$d_1(\text{m}) \in \{1, 3, 5, 7, 10, 20, 50, 100\}$$
$$\sigma_d(\text{m}) \in \{0.05, 0.1, 0.25, 0.5, 0.75, 1, 2, 5\}.$$

The simulation algorithm detailed in Fig. 5 is applied to every triplet $\{\alpha, d_1, \sigma_d\}$ to count the percentage of rejected hypotheses h and the corresponding linearity index L_d.

VII. Experimental Results

The performance of the new parametrization has been tested on real-image sequences acquired with a handheld-low-cost Unibrain IEEE1394 camera, with a 90° field of view and

```
input: $\alpha, d_1, \sigma_d$
output: $h,\ L_d$
$\sigma_u = \left|\frac{\sin\alpha}{d_1}\right| \sigma_d$; $\mu_u = 0$; //(29)
$\alpha_{sl} = 0.05$; // Kolm. test sign. level
$L_d = \frac{4\sigma_d}{d_1} |\cos\alpha|$
n_rejected=0 ;
N_GENERATED_SAMPLES=1000;
SAMPLE_SIZE=1000;

for j=1 to N_GENERATED_SAMPLES repeat
  $\{d_i\}_j$=random_normal(0,$\sigma_d^2$,SAMPLE_SIZE);
      //generate a normal sample from $N\left(0,\sigma_d^2\right)$;
  $\{u_i\}_j$=propagate_from_dept_to_image($\{d_i\}_j$,$\alpha$,$d_1$);//(14)
  if rejected==Kolmogorov_Smirnov($\{u_i\}_j$, $\mu_u, \sigma_u, \alpha_{sl}$)
    n_rejected=n_rejected+1;
endfor
$h$=100$\frac{[n\_rejected]}{[N\_GENERATED\_SAMPLES]}$;
```

Fig. 5. Simulation algorithm to test the linearity of the measurement equation.

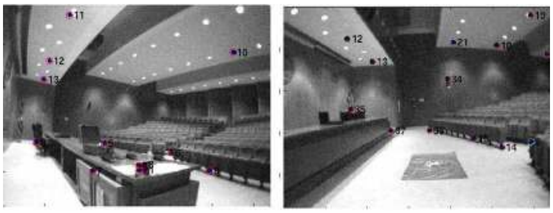

Fig. 6. First (a) and last (b) frame in the sequence of the indoor experiment of Section VII-A. Features 11, 12, and 13 are analyzed. These features are initialized in the same frame but are located at different distances from the camera.

320 × 240 resolution, capturing monochrome image sequences at 30 fps.

Five experiments were performed. The first was an indoor sequence processed offline with a Matlab implementation, the goal being to analyze initialization of scene features located at different depths. The second experiment shows an outdoor sequence processed in real time with a C++ implementation. The focus was on distant features observed under low parallax along the whole sequence. The third experiment was a loop closing sequence, concentrating on camera covariance evolution. Fourth was a simulation experiment to analyze the effect of switching from inverse depth to *XYZ* representations. In the last experiment, the switching performance was verified on the real loop closing sequence. This section ends with a computing time analysis. It is worth noting that no initial pattern to fix the scale was used in any of the last three experiments.

A. Indoor Sequence

This experiment analyzes the performance of the inverse depth scheme as several features at a range of depths are tracked within SLAM. We discuss three features, which are all detected in the same frame but have very different depths. Fig. 6 shows the image where the analyzed features are initialized (frame 18 in the sequence) and the last image in the sequence. Fig. 7 focuses on the evolution of the estimates corresponding to the features, with labels 11, 12, and 13, at frames 1, 10, 25, and 100. Confidence regions derived from the inverse depth representation (thick red line) are plotted in the *XYZ* space by numerical Monte Carlo propagation from the 6-D multivariate Gaussians representing these features in the SLAM EKF. For comparison, standard Gaussian *XYZ* acceptance ellipsoids (thin black line) are linearly propagated from the 6-D representation by means of the Jacobian of (28). The parallax α in degrees for each feature at every step is also displayed.

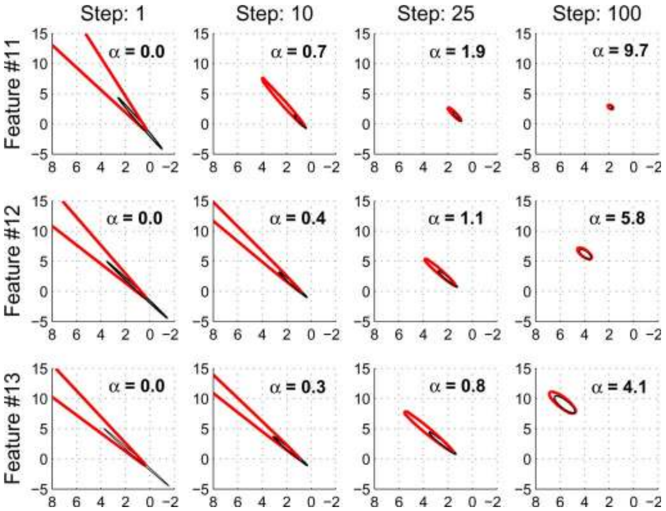

Fig. 7. Feature initialization. Each column shows the estimation history for a feature horizontal components. For each feature, the estimates after 1, 10, 25, and 100 frames since initialization are plotted; the parallax angle α in degrees between the initial observation and the current frame is displayed. The thick (red) lines show (calculated by a Monte Carlo numerical simulation) the 95% confidence region when coded as Gaussian in inverse depth. The thin (black) ellipsoids show the uncertainty as a Gaussian in the *XYZ* space propagated according to (28). Notice how at low parallax, the inverse depth confidence region is very different from the elliptical Gaussian. However, as the parallax increases, the uncertainty reduces and collapses to the Gaussian ellipse.

When initialized, the 95% acceptance region of all the features includes $\rho = 0$, so infinite depth is considered as a possibility. The corresponding confidence region in depth is highly asymmetric, excluding low depths but extending to infinity. It is clear that Gaussianity in inverse depth is not mapped to Gaussianity in *XYZ*, so the black ellipsoids produced by Jacobian transformation are far from representing the true depth uncertainty. As stated in Section IV-D, it is at low parallax that the inverse depth parametrization plays a key role.

As rays producing bigger parallax are gathered, the uncertainty in ρ becomes smaller but still maps to a nonGaussian distribution in *XYZ*. Eventually, at high parallax, for all of the features, the red confidence regions become closely Gaussian and well approximated by the linearly propagated black ellipses—but this happens much sooner for nearby feature 11 than distant feature 13.

A movie showing the input sequence and estimation history of this experiment is available as multimedia data `inverseDepth_indoor.avi`. The raw input image sequence is also available at `inverseDepth_indoorRawImages.tar.gz`.

B. Real-Time Outdoor Sequence

This 860 frame experiment was performed with a C++ implementation that achieves real-time performance

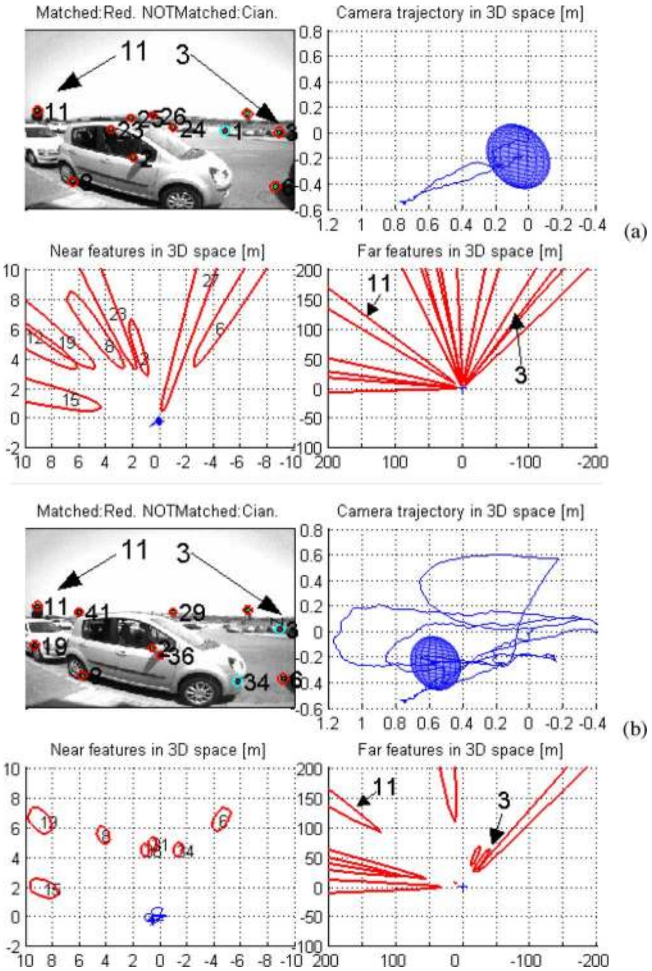

Fig. 8. (a) and (b) show frames #163 and #807 from the outdoor experiment of Section VII-B. This experiment was processed in real time. The focus was two features: 11 (tree on the left) and 3 (car on the right) at low parallax. Each of the two figures shows the current images and top-down views illustrating the horizontal components of the estimation of camera and feature locations at three different zoom scales for clarity: the top-right plots (maximum zoom) highlight the estimation of the camera motion; bottom-left (medium zoom) views highlight nearby features; and bottom-right (minimum zoom) emphasizes distant features.

at 30 fps with the handheld camera. Here, we highlight the ability of our parametrization to deal with both close and distant features in an outdoor setting. The input image sequence is available at multimedia material `inverseDepth_outdoorRawImages.tar.gz`. A movie showing the estimation process is also available at `inverseDepth_outdoor.avi`.

Fig. 8 shows two frames of the movie illustrating the performance. For most of the features, the camera ended up gathering enough parallax to accurately estimate their depths. However, being outdoors, there were distant features producing low parallax during the whole camera motion.

The inverse depth estimation history for two features is highlighted in Fig. 9. It is shown that distant, low parallax features are persistently tracked through the sequence, despite the fact that their depths cannot be precisely estimated. The large depth uncertainty, represented with the inverse depth scheme, is successfully managed by the SLAM EKF, allowing the orientation information supplied by these features to be exploited.

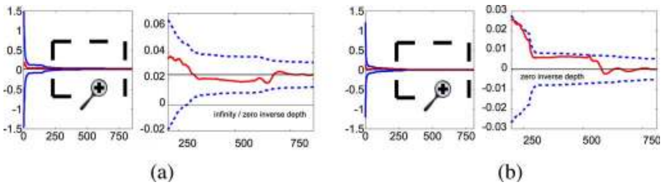

Fig. 9. Analysis of outdoor experiment of Section VII-B. (a) Inverse depth estimation history for feature 3, on the car, and (b) for feature 11, on a distant tree. Due to the uncertainty reduction during estimation, two plots at different scales are shown for each feature. It shows the 95% confidence region, and with a thick line, the estimated inverse depth. The thin solid line is the inverse depth estimated after processing the whole sequence. In (a), for the first 250 steps, zero inverse depth is included in the confidence region, meaning that the feature might be at infinity. After this, more distant but finite locations are gradually eliminated, and eventually, the feature's depth is accurately estimated. In (b), the tree is so distant that the confidence region always includes zero since little parallax is gathered for that feature.

Feature 3, on a nearby car, eventually gathers enough parallax enough to have an accurate depth estimate after 250 images, where infinite depth is considered as a possibility. Meanwhile, the estimate of feature 11, on a distant tree and never displaying significant parallax, never collapses in this way and zero inverse depth remains within its confidence region. Delayed intialization schemes would have discarded this feature without obtaining any information from it, while in our system, it behaves like a bearing reference. This ability to deal with distant points in real time is a highly advantageous quality of our parametrization. Note that what does happen to the estimate of feature 11 as translation occurs is that hypotheses of nearby depths are ruled out—the inverse depth scheme correctly recognizes that measuring little parallax while the camera has translated some distance allows a minimum depth for the feature to be set.

C. Loop Closing Sequence

A loop closing sequence offers a challenging benchmark for any SLAM algorithm. In this experiment, a handheld camera was carried by a person walking in small circles within a very large student laboratory, carrying out two complete laps. The raw input image sequence is available at `inverseDepth_loopClosingRawImages.tar.gz`, and a movie showing the mapping process at `inverseDepth_loopClosing.avi`.

Fig. 10 shows a selection of the 737 frames from the sequence, concentrating on the beginning, first loop closure, and end of the sequence. Fig. 11 shows the camera location estimate covariance history, represented by the 95% confidence regions for the six camera DOF and expressed in a reference local to the camera.

We observe the following properties of the evolution of the estimation, focussing, in particular, on the uncertainty in the camera location.

1) After processing the first few images, the uncertainty in the depth of features is huge, with highly nonelliptical confidence regions in the *XYZ* space [Fig. 10(a)].
2) In Fig. 11, the first peak in the *X* and *Z* translation uncertainty corresponds to a camera motion backward along

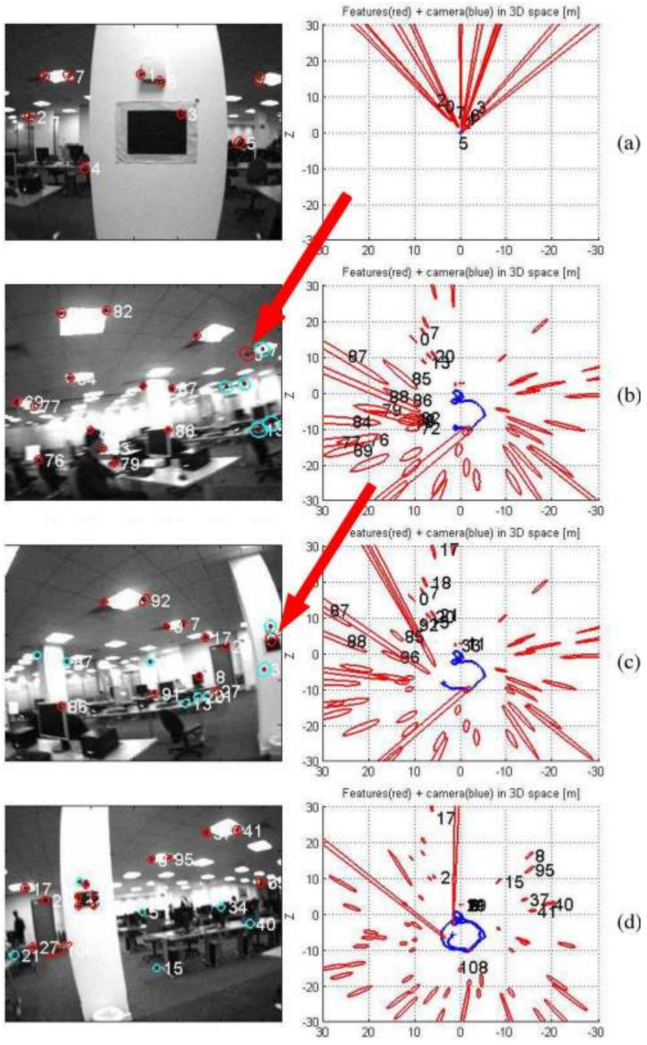

Fig. 10. Selection of frames from the loop closing experiment of Section VII-C. For each frame, we show the current image and reprojected map (left), and a top-down view of the map with 95% confidence regions and camera trajectory (right). Notice that confidence regions for the map features are far from being Gaussian ellipses, especially for newly initialized or distant features. The selected frames are: (a) #11, close to the start of the sequence; (b) #417, where the first loop closing match, corresponding to a distant feature, is detected; the loop closing match is signaled with an arrow; (c) #441, where the first loop closing match corresponding to a close feature is detected; the match is signaled with an arrow; and (d) #737, the last image, in the sequence, after reobserving most of the map features during the second lap around the loop.

the optical axis; this motion produces poor parallax for newly initialized features, and we, therefore, see a reduction in the orientation uncertainty and an increase in the translation uncertainty. After frame #50, the camera again translates in the *X*-direction, parallax is gathered, and the translation uncertainty is reduced.

3) From frame #240, the camera starts a 360° circular motion in the *XZ* plane. The camera explores new scene regions, and the covariance increases steadily as expected (Fig. 11).
4) In frame #417, the first loop closing feature is reobserved. This is a feature that is distant from the camera, and causes an abrupt reduction in the orientation and translation uncertainty (Fig. 11), though a medium level of uncertainty remains.

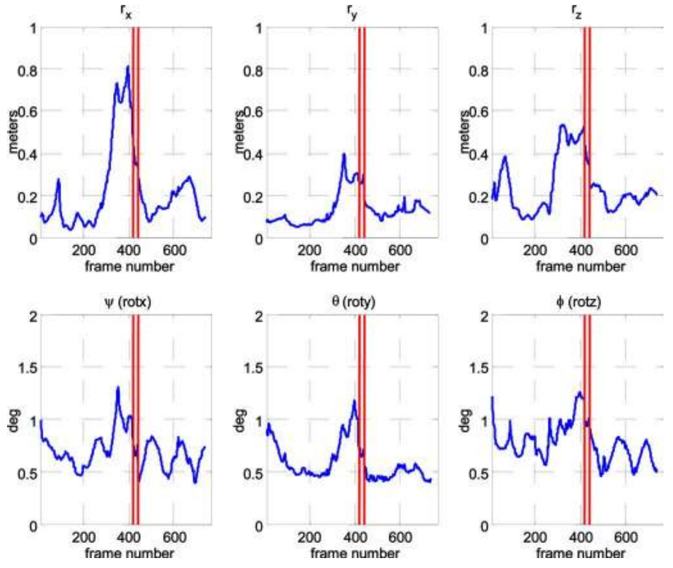

Fig. 11. Camera location estimate covariance along the sequence. The 95% confidence regions for each of the 6 DOF of camera motion are plotted. Note that errors are expressed in a reference local to the camera. The vertical solid lines indicate the loop closing frames #417 and #441.

5) In frame #441, a much closer loop closing feature (mapped with high parallax) is matched. Another abrupt covariance reduction takes place (Fig. 11) with the extra information this provides.
6) After frame #441, as the camera goes on a second lap around the loop, most of the map features are revisited, almost no new features are initalized, and hence, the uncertainty in the map is further reduced. Comparing the map at frame #441 (the beginning of the second lap) and #737, (the end of the second lap), we see a significant reduction in uncertainty. During the second lap, the camera uncertainty is low, and as features are reobserved, their uncertainties are noticeably reduced [Fig. 10(c) and (d)].

Note that these loop closing results with the inverse depth representation show a marked improvement on the experiments on monocular SLAM with a humanoid robot presented in [9], where a gyro was needed in order to reduce angular uncertainty enough to close loops with very similar camera motions.

D. Simulation Analysis for Inverse Depth to XYZ Switching

In order to analyze the effect of the parametrization switching proposed in Section VI on the consistency of SLAM estimation, simulation experiments with different switching thresholds were run. In the simulations, a camera completed two laps of a circular trajectory of radius 3 m in the XZ plane, looking out radially at a scene composed of points lying on three concentric spheres of radius 4.3, 10, and 20 m. These points at different depths were intended to produce observations with a range of parallax angles (Fig. 12).

The camera parameters of the simulation correspond with our real image acquisition system: camera 320×240 pixels, frame rate 30 frames/s, image field of view 90°, measurement uncertainty for a point feature in the image, and Gaussian

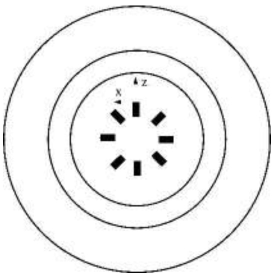

Fig. 12. Simulation configuration for analysis of parametrization switching in Section VII-D, sketching the circular camera trajectory and 3-D scene, composed of three concentric spheres of radius 4.3, 10, and 20 m. The camera completes two circular laps in the (XZ)-plane with radius 3 m, and is orientated radially.

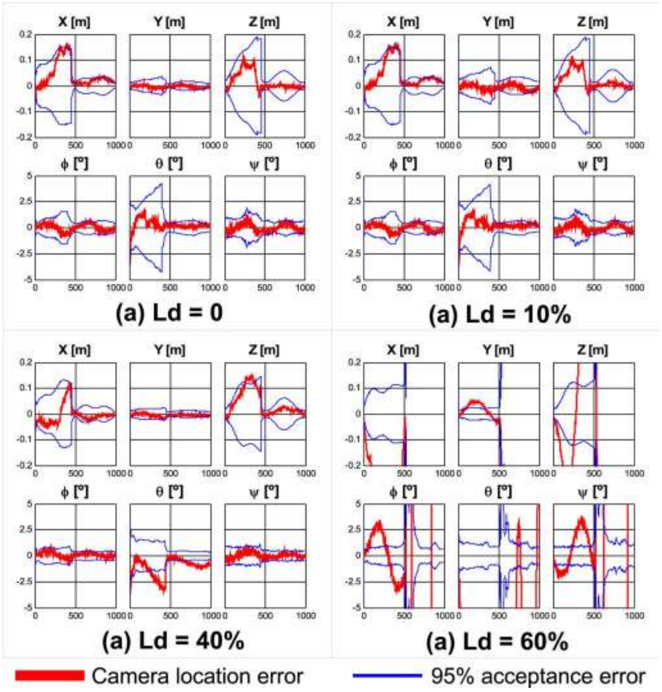

Fig. 13. Details from the parametrization switching experiment. Camera location estimation error history in 6 DOF. (translation in XYZ, and three orientation angles $\psi\theta\phi$) for four switching thresholds: With $L_d = 0\%$, no switching occurs and the features all remain in the inverse depth parametrization. At $L_d = 10\%$, although features from the spheres at 4.3 and 10 m are eventually converted, no degradation with respect to the non-switching case is observed. At $L_d = 40\%$, some features are switched before achieving true Gaussianity, and there is noticeable degradation, especially in θ rotation around the Y axis. At $L_d = 60\%$, the map becomes totally inconsistent and loop closing fails.

$N\left(0, 1\ \text{pixel}^2\right)$. The simulated image sequence contained 1000 frames. Features were selected following the randomized map management algorithm proposed in [8] in order to have 15 features visible in the image at all times. All our simulation experiments work using the same scene features in order to homogenize the comparison.

Four simulation experiments for different thresholds for switching each feature from inverse depth to *XYZ* parametrization were run with $L_d \in \{0\%, 10\%, 40\%, 60\%\}$. Fig. 13 shows the camera trajectory estimation history in 6 DOF (translation in XYZ, and three orientation angles $\psi(\text{Rot}_x), \theta(\text{Rot}_y), \phi(\text{Rot}_z, \text{cyclotorsion})$). The following conclusions can be made.

1) Almost the same performance is achieved with no switching (0%) and with 10% switching. So, it is clearly advantageous to perform 10% switching because there is no penalization in accuracy and the state vector size of each converted feature is halved.
2) Switching too early degrades accuracy, especially in the orientation estimate. Notice how for 40% the orientation estimate is worse and the orientation error covariance is smaller, showing filter inconsistency. For 60%, the estimation is totally inconsistent and the loop closing fails.
3) Since early switching degrades performance, the inverse depth parametrization is mandatory for initialization of every feature and over the long term for low parallax features.

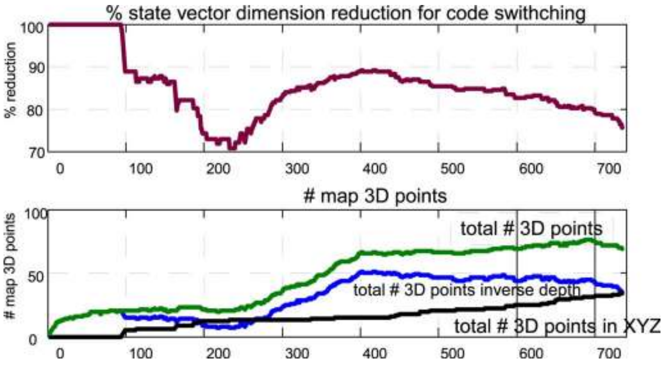

Fig. 14. Parametrization switching on a real sequence (Section VII-E): state vector size history. Top: percentage reduction in state dimension when using switching compared with keeping all points in inverse depth. Bottom: total number of points in the map, showing the number of points in inverse depth and the number of points in *XYZ.*

E. Parametrization Switching With Real Images

The loop closing sequence of Section VII-C was processed without any parametrization switching, and with switching at $L_d = 10\%$. A movie showing the results is available at `inverseDepth_loopClosing_ID_to_XYZ_conversion.avi`. As in the simulation experiments, no significant change was noticed in the estimated trajectory or map.

Fig. 14 shows the history of the state size, the number of map features, and how their parametrization evolves. At the last estimation step, about half of the features had been switched; at this step, the state size had reduced from 427 to 322 (34 inverse depth features and 35 *XYZ*), i.e., 75% of the original vector size. Fig. 15 shows four frames from the sequence illustrating feature switching. Up to step 100, the camera has low translation and all the features are in inverse depth form. As the camera translates, nearby features switch to *XYZ.* Around step 420, the loop is closed and features are reobserved, producing a significant reduction in uncertainty that allows switching of more reobserved close features. Our method automatically determines which features should be represented in the inverse depth or *XYZ* forms, optimizing computational efficiency without sacrificing accuracy.

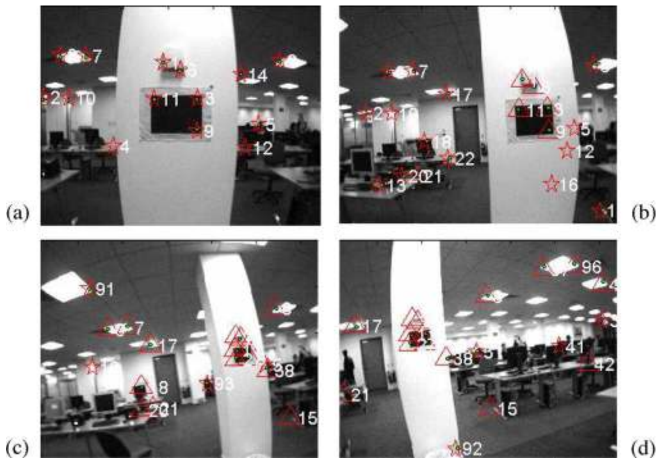

Fig. 15. Parametrization switching seen in image space: points coded in inverse depth ($\star$) and coded in *XYZ* ($\triangle$). (a) First frame, with all features in inverse depth. (b) Frame #100; nearby features start switching. (c) Frame #470, loop closing; most features in *XYZ*. (d) Last image of the sequence.

F. Processing Time

We give some details of the real-time operation of our monocular SLAM system, running on a 1.8 GHz Pentium M processor laptop. A typical EKF iteration would implies the following.

1) A state vector dimension of 300.
2) 12 features observed in the image, a measurement dimension of 24.
3) 30 fps, so 33.3 ms available for processing.

Typical computing time breaks down as follows: image acquisition, 1 ms.; EKF prediction, 2 ms; image matching, 1 ms.; and EKF update, 17 ms. This adds up to a total of 21 ms. The remaining time is used for graphics functions using OpenGL on an NVidia card and scheduled at a low priority.

The quoted state vector size 300 corresponds to a map size of 50 if all features are encoded using inverse depth. In indoor scenes, due to switching maps of up to 60–70, features can be computed in real time. This size is enough to map many typical scenes robustly.

VIII. Conclusion

We have presented a parametrization for monocular SLAM that permits operation based uniquely on the standard EKF prediction-update procedure at every step, unifying initialization with the tracking of mapped features. Our inverse depth parametrization for 3-D points allows unified modeling and processing for any point in the scene, close or distant, or even at "infinity." In fact, close, distant, or just-initialized features are processed within the routine EKF prediction-update loop without making any binary decisions. Due to the undelayed initialization and immediate full use of infinite points, estimates of camera orientation are significantly improved, reducing the camera estimation jitter often reported in previous work. The jitter reduction, in turn, leads to computational benefits in terms of smaller search regions and improved image processing speed.

The key factor is that due to our parametrization of the direction and inverse depth of a point relative to the location from which it was first seen, our measurement equation has low linearization errors at low parallax, and hence, the estimation uncertainty is accurately modeled with a multivariate Gaussian. In Section IV, we presented a model that quantifies linearization error. This provides a theoretical understanding of the impressive outdoor, real-time performance of the EKF with our parametrization.

The inverse depth representation requires a 6-D state vector per feature, compared to three for *XYZ* coding. This doubles the map state vector size, and hence produces a fourfold increase in the computational cost of the EKF. Our experiments show that it is essential to retain the inverse depth parametrization for intialization and distant features, but nearby features can be safely converted to the cheaper *XYZ* representation, meaning that the long-term computational cost need not increase significantly. We have given details on when this conversion should be carried out for each feature to optimize computational efficiency without sacrificing accuracy.

The experiments presented have validated the method with real imagery using a handheld camera as the only sensor, both indoors and outdoors. We have experimentally verified the following the key contributions of our study:

1) real-time performance achieving 30 fps real-time processing for maps up to 60–70 features;
2) real-time loop closing;
3) dealing simultaneously with low and high parallax features;
4) nondelayed initialization;
5) low jitter, full 6-DOF monocular SLAM.

In the experiments, we have focused on a map size around 60–100 features because these map sizes can be dealt with in real time at 30 Hz, and we have focused on the challenging loop closing issue. Useful future work would be a thorough analysis of the limiting factors in EKF inverse depth monocular SLAM in terms of linearity, data association errors, accuracy, map size, and ability to deal with degenerate motion such as pure rotations or a static camera for long-time periods.

Finally, our simulations and experiments have shown that inverse depth monocular SLAM operates well without known patterns in the scene to fix scale. This result points toward further work in understanding the role of scale in monocular SLAM (an avenue that we have begun to investigate in a dimensionless formulation in [6]) and further bridging the gap between sequential SLAM techniques and structure from motion methods from the computer vision literature.

Appendix

To recover the ideal projective undistorted coordinates $\mathbf{h}_u = (u_u, v_u)^\top$ from the actually distorted ones gathered by the camera $\mathbf{h}_d = (u_d, v_d)^\top$, the classical two parameters radial distortion model [17] is applied:

$$\begin{pmatrix} u_u \\ v_u \end{pmatrix} = \mathbf{h}_u \begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} u_0 + (u_d - u_0)\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) \\ v_0 + (v_d - v_0)\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) \end{pmatrix}$$

$$r_d = \sqrt{\left(d_x(u_d - u_0)\right)^2 + \left(d_y(v_d - v_0)\right)^2} \qquad (30)$$

where u_0, v_0 are the image centers and κ_1, κ_2 are the radial distortion coefficients.

To compute the distorted coordinates from the undistorted

$$\begin{pmatrix} u_d \\ v_d \end{pmatrix} = \mathbf{h}_d \begin{pmatrix} u_u \\ v_u \end{pmatrix} = \begin{pmatrix} u_0 + \dfrac{(u_u - u_0)}{(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4)} \\ v_0 + \dfrac{(v_u - v_0)}{(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4)} \end{pmatrix} \tag{31}$$

$$r_u = r_d \left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) \tag{32}$$

$$r_u = \sqrt{\left(d_x (u_u - u_0)\right)^2 + \left(d_y (v_u - v_0)\right)^2} \tag{33}$$

where r_u is readily computed computed from (33), but r_d has to be numerically solved from (32), e.g, using Newton–Raphson method; hence, (31) can be used to compute the distorted point.

Undistortion Jacobian $\partial \mathbf{h}_u / \partial \left(u_d, v_d\right)$ has the following analytical expression:

$$\left(\begin{array}{c|c} \begin{array}{c} \left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) + \\ 2\left(\left(u_d - u_0\right) d_x\right)^2 \times \\ \left(\kappa_i + 2\kappa_2 r_d^2\right) \end{array} & \begin{array}{c} 2d_y^2 \left(u_d - u_0\right)\left(v_d - v_0\right) \times \\ \left(\kappa_1 + 2\kappa_2 r_d^2\right) \end{array} \\ \hline \begin{array}{c} 2d_x^2 \left(v_d - v_0\right)\left(u_d - u_0\right) \times \\ \left(\kappa_1 + 2\kappa_2 r_d^2\right) \end{array} & \begin{array}{c} \left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) + \\ 2\left(\left(v_d - v_0\right) d_y\right)^2 \times \\ \left(\kappa_i + 2\kappa_2 r_d^2\right) \end{array} \end{array}\right) \tag{34}$$

The Jacobian for the distortion is computed by inverting expression (34)

$$\left.\frac{\partial \mathbf{h}_d}{\partial \left(u_u, v_u\right)}\right|_{\left(u_u, v_u\right)} = \left(\left.\frac{\partial \mathbf{h}_u}{\partial \left(u_d, v_d\right)}\right|_{\mathbf{h}_d\left(u_u, v_u\right)}\right)^{-1}. \tag{35}$$

Acknowledgment

The authors are very grateful to D. Murray, I. Reid, and other members of Oxford's Active Vision Laboratory for discussions and software collaboration. They also thank the anonymous reviewers for their useful comments.

References

[1] V. J. Aidala and S. E. Hammel, "Utilization of modified polar coordinates for bearing-only tracking," *IEEE Trans. Autom. Control*, vol. 28, no. 3, pp. 283–294, Mar. 1983.

[2] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., Sep. 14–19, 2003, vol. 2, pp. 1966–1971.

[3] M. Bryson and S. Sukkarieh, "Bearing-only SLAM for an airborne vehicle," presented at the Australasian Conf. Robot. Autom. (ACRA 2005), Sydney, Australia.

[4] G. C. Canavos, *Applied Probability and Statistical Methods*. Boston, MA: Little, Brow, 1984.

[5] A. Chowdhury and R. Chellappa, "Stochastic approximation and rate-distortion analysis for robust structure and motion estimation," *Int. J. Comput. Vis.*, vol. 55, no. 1, pp. 27–53, 2003.

[6] J. Civera, A. J. Davison, and J. M. M. Montiel, "Dimensionless monocular SLAM," in *Proc. 3rd Iberian Conf. Pattern Recogn. Image Anal.*, 2007, pp. 412–419.

[7] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth to depth conversion for monocular SLAM," in *Proc. Int. Conf. Robot. Autom.*, 2007, pp. 2778–2783.

[8] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1403–1410.

[9] A. J. Davison, I. Reid, N. Molton, and O. Stasse, "Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[10] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.* Jun. 17–22, 2006, vol. 1, pp. 469–476.

[11] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proc. Eur. Conf. Comput. Vis.*, Jun. 1998, pp. 311–326.

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[13] D. Heeger and A. Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," *Int. J. Comput. Vis.*, vol. 7, no. 2, pp. 95–117, Jan. 1992.

[14] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proc. IEEE Int. Conf. Robot. Autom.* Sep. 14–19, 2003, vol. 1, pp. 406–411.

[15] N. Kwok and G. Dissanayake, "An efficient multiple hypothesis filter for bearing-only SLAM," in *Proc. IROS*, 28 Sep.–2–Oct. 2004, vol. 1, pp. 736–741.

[16] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *Int. J. Comput. Vis.*, vol. 3, no. 3, pp. 209–238, 1989.

[17] E. Mikhail, J. Bethel, and J. C. McGlone, *Introduction to Modern Photogrammetry*. New York: Wiley, 2001.

[18] J. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," presented at the Robot. Sci. Syst. Conf., Philadelphia, PA, Aug. 2006.

[19] J. Montiel and A. J. Davison, "A visual compass based on SLAM," in *Proc. Int. Conf. Robot. Autom.*, Orlando, FL, May 15–19, 2006, pp. 1917–1922.

[20] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real-time localization and 3-D reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2006, vol. 3, pp. 1027–1031.

[21] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robot.*, vol. 23, no. 1, pp. 3–26, 2006.

[22] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 353–363, Apr. 1993.

[23] M. Pollefeys, R. Koch, and L. Van Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," *Int. J. Comput. Vis.*, vol. 32, no. 1, pp. 7–25, 1999.

[24] J. Sola, "Towards visual localization, mapping and moving objects tracking by a mobile robot: A geometric and probabilistic approach," Ph.D. dissertation, LAAS-CNRS, Toulouse, France, 2007.

[25] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Proc. 2005 IEEE/RSJ Int. Conf. Intell. Robots Syst.,* 2–6 Aug., 2005, pp. 2499–2504.

[26] N. Trawny and S. I. Roumeliotis, "A unified framework for nearby and distant landmarks in bearing-only SLAM," in *Proc. Int. Conf. Robot. Autom.,* May 15–19, 2006, pp. 1923–1929.

Javier Civera was born in Barcelona, Spain, in 1980. He received the M.S. degree in industrial-electrical engineering in 2004 from the University in Zaragoza, where he is currently working toward the Ph.D. degree with the Robotics, Perception, and Real-Time Group.

He is currently an Assistant Lecturer with the Departamento de Informática, University of Zaragoza, where he teaches courses in automatic control theory. His current research interests include computer vision and mobile robotics.

Andrew J. Davison received the B.A. degree in physics and the D.Phil. degree in computer vision from the University of Oxford, Oxford, U.K., in 1994 and 1998, respectively.

He was with Oxford's Robotics Research Group, where he developed one of the first robot simultaneous localization and mapping (SLAM) systems using vision. He was a European Union (EU) Science and Technology Fellow at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan, for two years, where he was engaged in visual robot navigation. In 2000, he returned to the University of Oxford as a Postdoctoral Researcher. In 2005, he joined Imperial College London, London, U.K., where he currently holds the position of Reader with the Department of Computing. His current research interests include advancing the basic technology of real-time localization and mapping using vision while collaborating to apply these techniques in robotics and related areas.

Dr. Davison was awarded a five-year Engineering and Physical Sciences Research Council (EPSRC) Advanced Research Fellowship in 2002. In 2008, he was awarded a European Research Council (ERC) Starting Grant.

J. M. Martínez Montiel was born in Arnedo, Spain. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 1991 and 1996, respectively.

He is currently an Associate Professor with the Departamento de Informática, University of Zaragoza, where he is in charge of Perception and Computer Vision courses. His current interests include computer vision, real-time vision localization and mapping research, and the transference of this technology to robotic and nonrobotic application domains.

Dr. Montiel is member of the the Robotics, Perception, and Real-Time Group. He has been awarded the Spanish Merrimack Education Center (MEC) grants to fund research at the University of Oxford, Oxford, U.K., and Imperial College London, London, U.K.