Inverse Depth to Depth Conversion for Monocular SLAM

Javier Civera
Dpto. Informatica.
Universidad de Zaragoza
jcivera@unizar.es

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

J.M.M Montiel
Dpto. Informatica
Universidad de Zaragoza
josemari@unizar.es

***Abstract*— Recently it has been shown that an inverse depth parametrization can improve the performance of real-time monocular EKF SLAM, permitting undelayed initialization of features at all depths. However, the inverse depth parametrization requires the storage of 6 parameters in the state vector for each map point. This implies a noticeable computing overhead when compared with the standard 3 parameter XYZ Euclidean encoding of a 3D point, since the computational complexity of the EKF scales poorly with state vector size.**

In this work we propose to restrict the inverse depth parametrization only to cases where the standard Euclidean encoding implies a departure from linearity in the measurement equations. Every new map feature is still initialized using the 6 parameter inverse depth method. However, as the estimation evolves, if according to a linearity index the alternative XYZ coding can be considered linear, we show that feature parametrization can be transformed from inverse depth to XYZ for increased computational efficiency with little reduction in accuracy.

We present a theoretical development of the necessary linearity indices, along with simulations to analyze the influence of the conversion threshold. Experiments performed with with a 30 frames per second real-time system are reported. An analysis of the increase in the map size that can be successfully managed is included.

I. Introduction

Real-time SLAM (Simultaneous Localization and Mapping) using a 6 DOF agile monocular camera as the only sensor has been proven feasible since the work of Davison [1]. In this and other related work, a sparse map of 3D points is built on the fly as the camera's motion is simultaneously estimated. Each 3D point is parametrized by its 3 Euclidean coordinates XYZ. However, the weakness of the XYZ point encoding is its inability to deal with low parallax configurations, corresponding to two common cases: feature initialization and distant points. In both cases, the camera translation is small compared with observed feature depth; more precisely, the bundle of rays from different camera locations whose intersection defines the 3D point are all nearly parallel relative to the camera's bearing sensing accuracy. Since it is well understood that this situation leads to depth uncertainties which are not well characterized by a Gaussion distribution in 3D space, [1] used a 'delayed' initialization scheme, where full inclusion of a new feature in the map was postponed until significant parallax enabled a fairly accurate depth estimate to be accumulated via an auxiliary particle filter method. If little parallax was detected, features were never included in the map.

Recently, Montiel et al. [2] proposed an alternative 'inverse depth' parametrization for map features within monocular SLAM, noting that with this encoding the Gaussianity of the measurement equation is significantly improved features at all depths. They showed that when the inverse depth parametrization is used, a standard EKF (Extended Kalman Filter) algorithm can successfully deal with low parallax cases. This permits direct, non-delayed initialization, and map points at extreme, potentially 'infinite' depths. The non-delayed inclusion of every feature in the map allows even low-parallax features with unknown depths to provide orientation information immediately; jitter in the camera location estimation is noticeably reduced. The only drawback of the inverse depth approach is its computational cost because every map point is encoded using 6 parameters rather than the 3 of the more usual XYZ scheme. The extra 3 parameters come from the need to save the position of the camera from which the feature was first observed, since it is relative to this position that the inverse depth encoding has advantageous properties. Within the EKF, this is significant because the computational cost of each update scales with the square of the total size of the state vector.

Our contribution is to improve efficiency of the inverse depth parametrization scheme by proposing to transform features to an XYZ encoding as soon as this more efficient parametrization becomes well-behaved, meaning that a Gaussian distribution in these coordinates is a good fit for the uncertainty in the point location. So, retaining the inverse depth method of [2], features are initialized with six parameters — important at low parallax — but as the estimation evolves, if the 3 parameters XYZ encoding becomes well-behaved the feature is transformed from inverse depth to XYZ. We propose a test for transformation, relating to the feature parallax and estimation accuracy, which is tested individually for each feature at every estimation step. We show that algorithm performance does not degrade when compared to keeping every feature with an inverse depth encoding, but computational efficiency is greatly increased by decreasing the state vector size.

Other authors have also recently proposed novel initialization schemes which are closely related to the approach of [2]. Sola et al. in [3] proposed an initialization scheme based on maintaining several depth hypotheses combined with an approximated Gaussian Sum Filter. Eade and Drummond in [4] also proposed the inverse depth concept for feature initialization in a FastSLAM based approach. Trawny and Roumeliotis in [5] proposed a feature encoding using the concept of a virtual camera which allowed non-delayed initialization for 2D monocular vision.

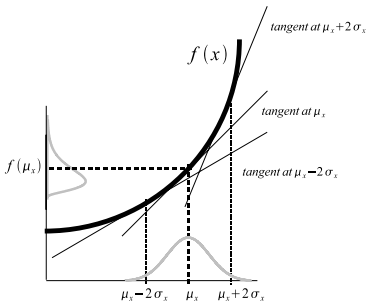

Fig. 1. The first derivative variation in $[\mu_x - 2\sigma_x, \mu_x + 2\sigma_x]$ codes the departure from Gaussianity in the propagation of the random variable through a function

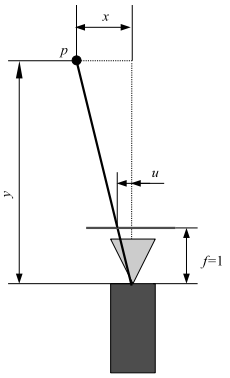

Fig. 2. Observation of a feature with a projective camera

The current paper propose a linearity analysis with a simplified model which ends up achieving the same linearity indices proposed in [2]. The analysis proposed in the current paper is simpler to understand and hence makes clearer the model's assumptions. Simulation are used to determine thresholds for the transformation of feature based on the proposed linearity indices.

Section II analyzes the linearized propagation of a Guaussian through a function and proposes a dimensionless linearity index to determine the departure from linearity. Next, in Section III, the linearity indices are evaluated for the inverse depth point encoding and for the XYZ coding such that the two can be compared directly. A simulation experiment is performed to determine the computed linearity index threshold for XYZ linearity. Section IV details the transformation from inverse depth to XYZ coding. Sections V and VI are devoted to simulation experiments to determine the degradation of estimation with respect to the transformation threshold and to real-time experiments with a hand-held camera. The paper ends with a Conclusions section.

II. Linearized propagation of a Gaussian

Let be x a gaussian random variable:

$$x \sim N\left(\mu_x, \sigma_x^2\right) \tag{1}$$

the image of x through the function f is a random variable y that can be approximated as Gaussian:

$$y \sim N\left(\mu_y, \sigma_y^2\right) \tag{2}$$

where:

$$\mu_y = f\left(\mu_x\right) \tag{3}$$

$$\sigma_y^2 = \left.\frac{\partial f}{\partial x}\right|_{\mu_x} \sigma_x^2 \left.\frac{\partial f}{\partial x}\right|_{\mu_x} \tag{4}$$

If the function f is linear in an interval around μ_x, then y is Gaussian (Fig 1.) It should be noticed that the interval size in which the function has to be linear is related with σ_x, the bigger the σ_x the wider the interval has to be in order to cover a significant fraction of the random variable x values. In this work we fix the linearity interval to the typical 95% region defined by $[\mu_x - 2\sigma_x, \mu_x + 2\sigma_x]$.

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*:

$$\frac{\partial f}{\partial x}\left(\mu_x + \Delta x\right) \approx \left.\frac{\partial f}{\partial x}\right|_{\mu_x} + \left.\frac{\partial^2 f}{\partial x^2}\right|_{\mu_x} \Delta x \tag{5}$$

We propose to compare the derivative value in the interval center, μ_x:

$$\left.\frac{\partial f}{\partial x}\right|_{\mu_x} \tag{6}$$

with the derivative value in the interval extrema $\mu_x \pm 2\sigma_x$(where the deviation is assumed to be maximal):

$$\left.\frac{\partial f}{\partial x}\right|_{\mu_x} \pm \left.\frac{\partial^2 f}{\partial x^2}\right|_{\mu_x} 2\sigma_x \tag{7}$$

The comparison is made by the next dimensionless linearity index:

$$L = \left| \frac{\left.\frac{\partial^2 f}{\partial x^2}\right|_{\mu_x} 2\sigma_x}{\left.\frac{\partial f}{\partial x}\right|_{\mu_x}} \right| \tag{8}$$

that compares the two summands in (7). When $L \approx 0$, the function can be considered linear in the interval, and hence the Gaussianity is preserved.

III. Measurement equation linearity

This section is devoted to presenting an simplified measurement equation model for a mobile camera observing a scene point. Despite its simplicity the model successfully codes the departure from linearity.

Fig. 2 sketches the image u of the point p with a normalized camera (a camera with unitary focal length).

$$u = \frac{x}{y} \tag{9}$$

In sections III-B and III-A we consider that the scene point is observed by two cameras (Figs. 3, 4) both cameras are gazing to the observed point. The first camera detects the ray where the point is. The only location error for the point is in depth. A second camera, observes the same point at a distance d_1; the parallax angle α is the angle between the

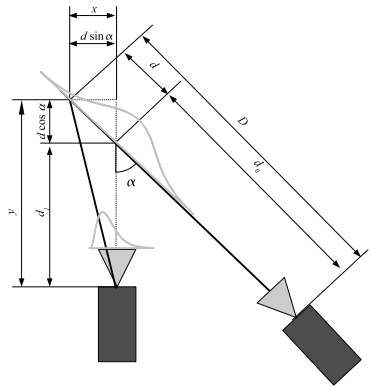

Fig. 3. Uncertainty propagation from the scene point to the image when the point is coded in depth.

two rays observing the point. It is approximated by the angle between the cameras optical axes.

We are interested in analyzing the linearity of the measurement equation for two different codings of an scene point. The inverse depth coding and the depth coding. Next two subsections deal with these two codings.

A. Depth Coding Linearity

We consider that the scene point is observed by two cameras, both cameras are pointing to the observed point (Fig 3.) The first camera detects the ray where the point is. The location error, d, of the point is coded as Gaussian in depth:

$$D = d_0 + d, \quad d \sim N\left(0, \sigma_d^2\right) \tag{10}$$

Next it is detailed how the error d is propagated to the image of the point in the second camera.

$$u = \frac{d \sin\alpha}{d_1 + d\cos\alpha} \tag{11}$$
$$x = d\sin\alpha \tag{12}$$
$$y = d_1 + d\cos\alpha \tag{13}$$

To analyze the Gaussianity of u, it is computed the linearity index L_d (8) as:

$$L_d = \frac{4\sigma_d}{d_1}\left|cos\alpha\right| \tag{14}$$

its detailed computation is given next:

$$L_d = \left|\frac{\left.\frac{\partial^2 u}{\partial d^2}\right|_{d=0} 2\sigma_d}{\left.\frac{\partial u}{\partial d}\right|_{d=0}}\right| \tag{15}$$
$$\frac{\partial u}{\partial d} = \frac{d_1 \sin\alpha}{\left(d_1 + d\cos\alpha\right)^2} \tag{16}$$
$$\frac{\partial^2 u}{\partial d^2} = \frac{-2d_1 \sin\alpha\cos\alpha}{\left(d_1 + d\cos\alpha\right)^3} \tag{17}$$

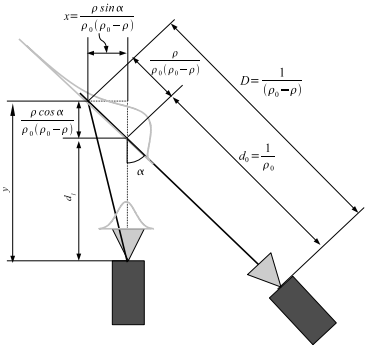

Fig. 4. Uncertainty propagation from the scene point to the image when the point is coded in inverse depth.

B. Inverse Depth Coding Linearity

The inverse depth coding, is based on the same scene geometry as the direct depth coding, but the depth error is coded as Gaussian in inverse depth (Fig 4):

$$D = \frac{1}{\rho_0 - \rho}, \quad \rho \sim N\left(0, \sigma_\rho^2\right) \tag{18}$$
$$d = D - d_0 = \frac{\rho}{\rho_0\left(\rho_0 - \rho\right)} \tag{19}$$
$$d_0 = \frac{1}{\rho_0} \tag{20}$$

So the image of the scene point is computed as:

$$u = \frac{\rho\sin\alpha}{\rho_0 d_1\left(\rho_0 - \rho\right) + \rho\cos\alpha} \tag{21}$$
$$x = d\sin\alpha = \frac{\rho\sin\alpha}{\rho_0\left(\rho_0 - \rho\right)} \tag{22}$$
$$y = d_1 + d\cos\alpha = d_1 + \frac{\rho\cos\alpha}{\rho_0\left(\rho_0 - \rho\right)} \tag{23}$$

So, the linearity index L_ρ is now:

$$L_\rho = \frac{4\sigma_\rho}{\rho_0}\left|1 - \frac{d_0}{d_1}cos\alpha\right| \tag{24}$$

Given that:

$$L_\rho = \left|\frac{\left.\frac{\partial^2 u}{\partial \rho^2}\right|_{\rho=0} 2\sigma_\rho}{\left.\frac{\partial u}{\partial \rho}\right|_{\rho=0}}\right| \tag{25}$$
$$\frac{\partial u}{\partial \rho} = \frac{\rho_0^2 d_1 \sin\alpha}{\left(\rho_0 d_1\left(\rho_0 - \rho\right) + \rho\cos\alpha\right)^2} \tag{26}$$
$$\frac{\partial^2 u}{\partial \rho^2} = \frac{-2\rho_0^2 d_1\left(\cos\alpha - d_1\rho_0\right)}{\left(\rho_0 d_1\left(\rho_0 - \rho\right) + \rho\cos\alpha\right)^3} \tag{27}$$

C. Simulation to Select a Linearity Index Threshold

Our proposal is to use index (14) to define a threshold to switch from the inverse depth coding to the depth coding. So, we use the depth coding when it can be considered linear.

If the depth coding is linear, then the measurement u is a Gaussian:

$$u \sim N\left(\mu_u, \sigma_u^2\right) \tag{28}$$

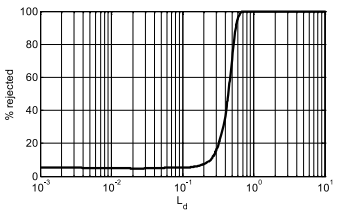

Fig. 5. Percentage of reject test with respect to the linearity index L_d

where according to (3), (4), and (16):

$$\mu_u = 0 \tag{29}$$

$$\sigma_u^2 = \left(\frac{\sin\alpha}{d_1}\right)^2 \sigma_d^2 \tag{30}$$

To determine the linearity threshold a simulation experiment applying a Kolmogorov-Smirnov test to verify the Gaussianty of u for a set of α, d_1 and σ_d values. The next simulation algorithm is applied to every $\{\alpha, d_1, \sigma_d\}$ triplet.

1) For 1000 random samples, repeat steps 2-4.
2) A Gaussian random sample $\{d_i\}$ size 1000 is drawn from $N\left(0, \sigma_d^2\right)$.
3) The sample is propagated to the image according to expression (11) to obtain $\{u_i\}$, a sample for the image measurements.
4) A Kolmogorov-Smirnov test is applied ($\alpha = 0.05$, significance level), the null hypothesis is $\{u_i\}$ follows a $N\left(\mu_u, \sigma_u^2\right)$ distribuion (28).
5) Compute the fraction of rejected null hypotheses h.
6) Compute the linearity index L_d (14) for the triplet $\{\alpha, d_1, \sigma_d\}$.

It the random sample is perfectly Gaussian, the fraction of null hypotheses rejected h should be the significance level 5%. Fig. 5, shows a plot of h with respect to L_d. It can be clearly seen how when $L_d > 0.2$ h abruptly departs from 5%. The simulation has been performed for all the triplets resulting from the next selected parameters values:

$$\alpha(\text{deg}) \in \{0.1, 1, 3, 5, 7, 10, 20, 30, 40, 50, 60, 70\} \tag{31}$$

$$d_1(\text{m}) \in \{1, 3, 5, 7, 10, 20, 50, 100\} \tag{32}$$

$$\sigma_d(\text{m}) \in \{0.050.10.250.50.75125\} \tag{33}$$

So our threshold for switching from inverse depth to depth is fixed in:

$$L_d = \frac{4\sigma_d}{d_1}\left|\cos\alpha\right| < 10\% \tag{34}$$

Notice that plot in Fig. 5 is smooth, what indicates that the linearity index effective codes the departure from linearity.

IV. Switch from Inverse Depth to Depth

This section is devoted to detailing the switch for a scene point coding. After processing an image, we have a joint estimate for the camera location and each scene feature. We focus on the camera translation and the inverse depth

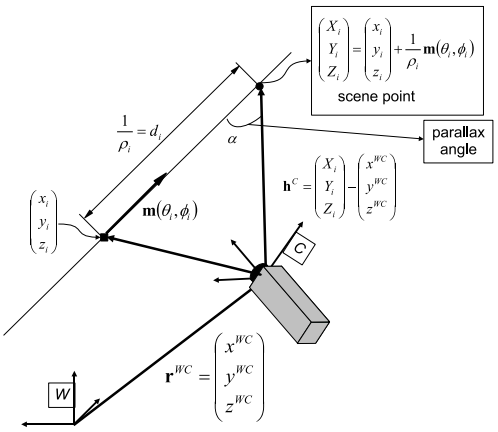

Fig. 6. Inverse depth point coding

coding of a point, so the relevant parts of the state vector and covariance are:

$$\mathbf{x} = \left(\mathbf{r}^{WC\top}, \ldots, \mathbf{y}_i^\top, \ldots\right)^\top \tag{35}$$

$$\mathbf{P} = \begin{pmatrix} \mathbf{P_{rr}} & \cdots & \mathbf{P_{ry}}_i & \cdots \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{\mathbf{ry}_i}^\top & \cdots & \mathbf{P}_{\mathbf{y}_i\mathbf{y}_i} & \cdots \\ \vdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{36}$$

where (Fig. 6): $\mathbf{r}^{WC\top} = \left(x^{WC}, y^{WC}, z^{WC}\right)$ is the camera translation estimate. $\mathbf{y}_i$ is the inverse depth point coding. It relates to the XYZ coding, $\mathbf{x}_i$ as:

$$\mathbf{x}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right) \tag{37}$$

$$\mathbf{y}_i = \begin{pmatrix} x_i & y_i & z_i & \theta_i & \phi_i & \rho_i \end{pmatrix}^\top \tag{38}$$

$$\mathbf{m} = \left(\cos\phi_i \sin\theta_i, -\sin\phi_i, \cos\phi_i\cos\theta_i\right)^\top \tag{39}$$

After each estimation step, the linearity index L_d (14) can be computed from the available estimate using (39), (37), (35), and (36) as:

$$d_i = \left\|\mathbf{h}^C\right\|, \quad \mathbf{h}^C = \mathbf{x}_i - \mathbf{r}^{WC} \tag{40}$$

$$\sigma_d = \frac{\sigma_\rho}{\rho_i^2}, \quad \sigma_\rho = \sqrt{\mathbf{P}_{\mathbf{y}_i\mathbf{y}_i}(6,6)} \tag{41}$$

$$\cos\alpha = \frac{\mathbf{m}^T\mathbf{h}^C}{\left\|\mathbf{h}^C\right\|} \tag{42}$$

If L_d is under the switching threshold, the feature in the state vector is switched using (37) and the covariance is transformed with the corresponding jacobian:

$$\mathbf{P}_{\text{new}} = \mathbf{J}\mathbf{P}\mathbf{J}^\top \tag{43}$$

$$\mathbf{J} = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & \frac{\partial \mathbf{x}_i}{\partial \mathbf{y}_i} & 0 \\ 0 & 0 & \mathbf{I} \end{pmatrix} \tag{44}$$

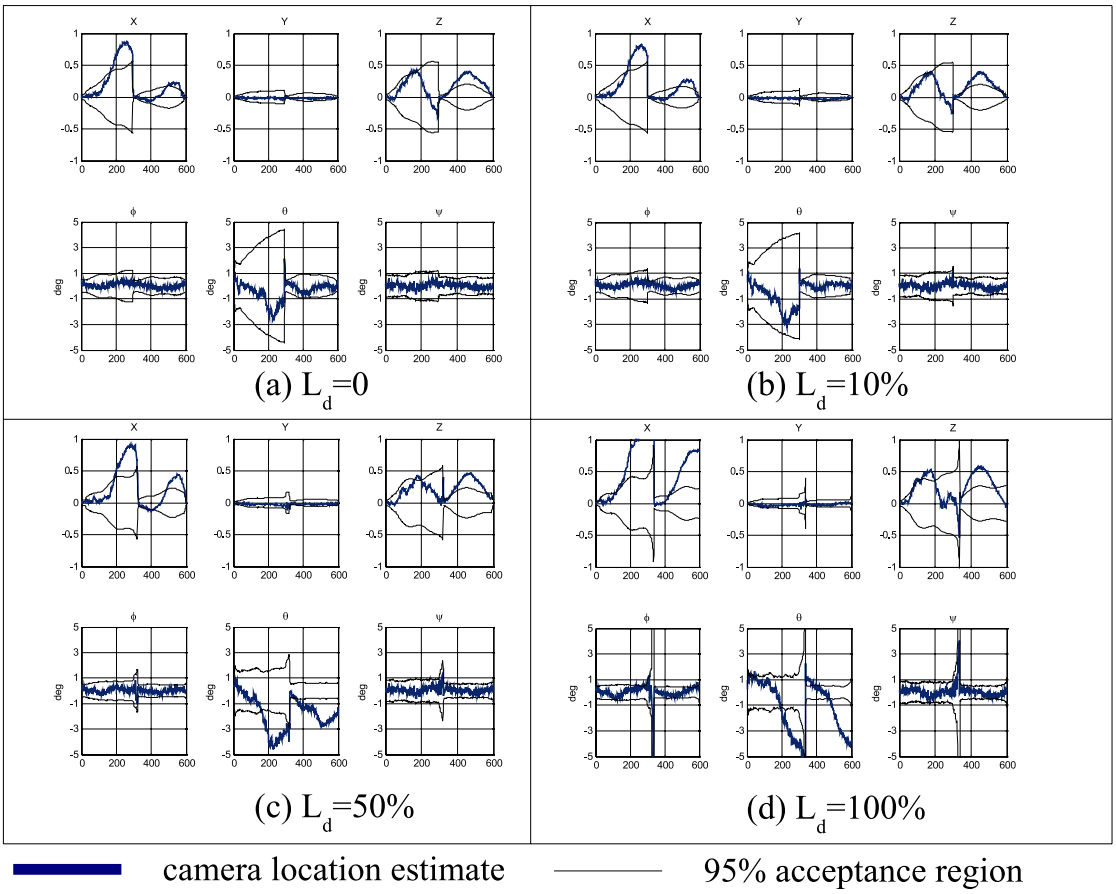

Fig. 8. Camera location estimation error history in 6 d.o.f. (translation in XYZ, and three orientation angles $\psi\theta\phi$) for four switching thresholds: $L_d = 0\%$, no switch, the features are always coded in inverse depth. $L_d = 10\%$ despite features over spheres 4.3 and 10 are eventually converted, no degradation with respect to the non-switch case is observed. $L_d = 50\%$ and $L_d = 100\%$ coding is switched before achieving a Gaussianity, noticeable degradation, especially in the θ rotation around Y axis.

V. Simulation Results

In order to analyze the effect of the coding switching on the consistency of the estimation, simulation experiments with different switch thresholds have been run. The estimation is computed in 3D, i.e full 6 d.o.f. for the camera motion and scene points are 3D points.

The camera parameters correspond with our real image acquisition system: camera 240×320 pixels, frame rate 30 frames/sec, image field of view $90°$. Measurement error for a point feature in the image, Gaussian $N\left(0, 1\text{pixel}^2\right)$, the image sequence is composed of 600 frames. Features are selected following the randomized map management algorithm proposed in [1] in order to have 15 features visible in the image. All the simulation experiments work using the same scene features, in order to homogenize the comparison. The camera trajectory describes two laps on a planar circumference radius 3m in the XZ plane; the camera orientation is always radial (Fig. 7.)

The scene is composed of points laying on 3 concentric spheres radius 4.3m, 10m and 20m. Points at different depths are intended to produce observations with a range of parallax angles.

Four simulation experiments, for different switching thresholds have been run, $L_d \in \{0\%, 10\%, 50\%, 100\%\}$. Fig. 8 shows the camera trajectory estimation history in 6 d.o.f. (translation in XYZ, and three orientation angles $\psi(\text{Rot}_x), \theta(\text{Rot}_y), \phi(\text{Rot}_z, \text{cyclotorsion})$). Next conclusions are derived:

- Almost the same performance is achieved with no switching (0%), and with 10% switching. So it is clearly advantageous to perform this switching because there is no penalization in performance and the computational cost per feature is divided by two.
- An early switching degrades the performance, especially in the orientation estimate. Notice how for 50%, and 100%, the orientation estimate is worse and the estimate for the orientation error covariance is smaller, so inconsistent.
- Early switching degrades the performance, so inverse depth coding is mandatory for initialization of every feature and low-parallax features.

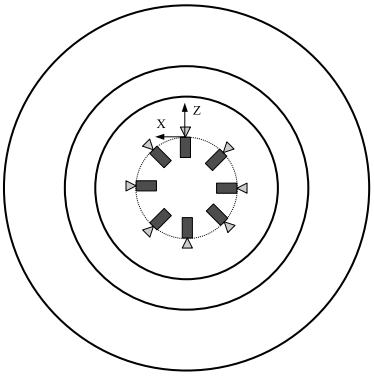

Fig. 7. Top view outline for the 6 d.o.f camera trajectory and 3D scene.The scene is composed of 3 concentric spheres radius 4.3m, 10m and 20m. The camera trajectory describes two laps on a planar circumference (XZ) plane, radius 3m, camera orientation is radial.

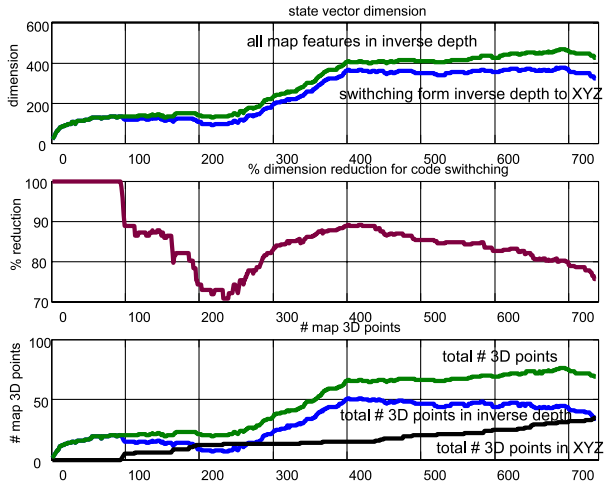

Fig. 9. State vector size history. Middle plot shows the size reduction compared with the original inverse depth coding.

VI. Real Image Experiments

A 737 image loop closing sequence acquired at 30 frames per second has been processed without any switching, and switching at $L_d = 10\%$. According to the simulation results, no significant change has been noticed in the estimated trajectory or map.

Fig. 9 shows the history of the state size. Fig. 10 shows 4 frames illustrating the feature switching. Up to step 100 the camera has a low translation and all the features are in inverse depth. As camera translates close features switch to XYZ. About step 420, the loop is closed, so the features are reobserved, producing a reduction in their uncertainty, what implies the switching of the reobserved close features. At the last estimation step about half of the features has been switched; at this step the state size has reduced from 427 to 322 what implies 75% of the original vector size.

Real-time experiments were run on a 1.8 GHz. Pentium M processor laptop with OpenGL accelerated graphics card. A typical EKF iteration at 33.3ms might imply: 300 state

Fig. 10. Points coded in inverse depth plotted as $\star$ and coded in XYZ plotted as $\triangle$. (a) first frame, all features are inverse depth ones. (b) #100, close features start switching. (c) # 470, loop closing, most features in XYZ. (d) last image of the sequence.

vector size, 12 features observed in the image. So reduction due to the coding increases the number of map features and now the system is able to close a loop at 30 frames/s with a hand-held camera in a room size scenario.

VII. Conclusions

The switch from inverse depth to XYZ parametrization can reduce the point coding from 6 parameters to 3 without degrading the estimation accuracy. A dimensionless index has been proposed to define the threshold for switching. An early switch has proven to degrade the performance, so the inverse depth coding at initialization and for distant features is mandatory.

The approach has been validated with real image experiments. 30 Hz real time performance is achieved up to a 300 sized state vector.

VIII. Acknowledgments

This research was supported by Spanish CICYT DPI2003-07986, EPSRC GR/T24685, Advanced Research Fellowship to A. J. Davison 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 Paul Smith for discussions and software collaboration.

References

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

[2] J. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Robotics Science and Systems Conference. Philadelphia.*, 2006(accepted).

[3] 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.

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

[5] N. Trawny and S. I. Roumeliotis, "A unified framework for nearby and distant landmarks in bearing-only slam," in *IEEE Int. Conf. Robotics and Automation*, 2006, pp. 1923–1929.