

## Delayed Inverse Depth Monocular SLAM

Rodrigo Munguia, Antoni Grau

Department of Automatic Control, Technical University of Catalonia, UPC ,  
c/ Pau Gargallo, 5 E-08028 Barcelona, Spain,  
{rodrigo.munguia;antoni.grau}@upc.edu.

---

**Abstract:** The 6-DOF monocular camera case possibly represents the harder variant in the context of simultaneous localization and mapping problem. In the last years, several advances have been appeared in this area; however the application of these techniques to real world applications it's difficult so far. Recently, the unified inverse depth parametrization has shown to be a good option this challenging problem, in a scheme of EKF for the estimation of the stochastic map and camera pose. In this paper a new delayed initialization scheme is proposed for adding new features to the stochastic map. The results show that delayed initialization can improve some aspects without losing the performance and unified aspect of the original method, when initial reference points are used in order to fix a metric scale in the map.

---

### 1. INTRODUCTION

In recent works, (Montiel, 2006a,b) and (Eade, 2006) have shown that the use of an inverse depth parametrization for monocular SLAM can improve the linearity of the measurement equation even for small changes in the camera position yielding small changes in the parallax angle, this fact allows a Gaussian distribution to cover uncertainty in depth which spans a depth range from nearby to infinity.

In the unified inverse depth method presented by (Montiel, 2006a), transition from partially to fully initialized features need not to be explicitly tackled, making it suitable for direct use in EKF framework for sparse mapping. In this approach the features are initialized in the first frame observed (undelayed initialization) with an initial fixed depth and uncertainty, determined heuristically to cover ranges from nearby to infinity, so distant points can be coded. Due to the clarity and scalability of this method, this approach is a good option for monocular-SLAM implementation.

Particularly, this work is motivated by the problems of vision-based robot map building and localization, therefore, if monocular SLAM wants to be applied in this context, retrieving the metric scale of the world is very important. The experiments with the unified inverse depth method show that, when initial reference points are used for establishing a metric scale in the map, the initial features depths have to be tuned, otherwise, is likely that new features added to the map never converges respect to the metric reference. On the other hand, initializing features distant to the optical camera center can increase the possibility that features depth become negative after a Kalman update step.

Initializing features in the first observed frame (undelayed initialization) avoids the use of pre-initialized features in the state and allows the use of all the information available in the

feature since it is detected, on the other hand, when features are detected in the image with a saliency operator in order to be automatically added to the map, usually the weak long-term image features are added to the map. Therefore it is difficult to match them in subsequent frames. When a minimum number of active image features want to be maintained, it could happen that unnecessary initializations are realized. Every new feature initialization introduces biases to the system (Davison, 2001).

The aforementioned issues suggested for new features, initial inverse depth and their associated initial uncertainty, could be treated before being added to the system state instead of using a fixed initial depth and uncertainty. At the same time features can be tested prior to be added to map in order to prune weak long-term features.

### 2. RELATED WORK

In (Davison, 2003) a multi-hypothesis method based on a particle filter to represent the initial depth of a feature is proposed. This work gives good results. However its application in large environments is not straightforward, as it would require a huge number of particles. In (Lemaire, 2005) is proposed a delayed multi-hypothesis method based in a sum of Gaussian mixture for depth estimation, but it uses odometry as an additional sensor. The work in (Eade 2006) is based in the FastSLAM algorithm, where the pose of the robot is represented by particles and a set of Kalman filters refine the estimation of the features, this approach is unable to code distant points. In the work presented in this paper, a delayed feature initialization is proposed for adding new features to the stochastic map in a context for monocular SLAM using inverse depth parametrization. The experimental results show that delayed initialization can improve some aspects without losing the performance and

unified aspect of the original (undelayed) method presented by (Montiel, 2006a), where initial reference points are used in order to fix a metric scale in the map.

### 3. INVERSE DEPTH MONOCULAR SLAM

#### 3.1 Camera motion model

A free camera moving in any direction in  $\mathbb{R}^3 \times SO(3)$  is considered. The camera state  $x_v$  is defined by:

$$x_v = [r^{WC} q^{WC} v^W \omega^W]^T \quad (1)$$

Where  $r^{WC} = [x, y, z]$  represents the camera optical center position,  $q^{WC} = [q_0, q_1, q_2, q_3]$  represents the camera orientation by a quaternion,  $v^W = [v_x, v_y, v_z]$  and  $\omega^W = [\omega_x, \omega_y, \omega_z]$  denote linear and angular velocities respectively. At every step it is assumed an unknown linear and angular acceleration with zero mean and known covariance Gaussian processes,  $a^W$  and  $\alpha^W$ , producing an impulse of linear and angular velocity such as:

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

The camera motion prediction model is:

$$f_v = \begin{bmatrix} r_{k+1}^{WC} \\ q_{k+1}^{WC} \\ v_{k+1}^W \\ \omega_{k+1}^W \end{bmatrix} = \begin{bmatrix} r_k^{WC} + (v_k^W + V_k^W) \Delta t \\ q_k^{WC} \times q((\omega_k^W + \Omega^W) \Delta t) \\ v_k^W + V^W \\ \omega_k^W + \Omega^W \end{bmatrix} \quad (3)$$

Being  $q((\omega_k^W + \Omega^W) \Delta t)$  the quaternion defined by the rotation vector  $(\omega_k^W + \Omega^W) \Delta t$ .

An Extended Kalman Filter propagates the camera pose and velocity estimates, as well as feature estimates.

#### 3.2 Features definition and measurement

The complete state that includes the features  $y$  is made of:

$$x = [x_v^T, y_1^T, y_2^T, \dots, y_n^T]^T \quad (4)$$

where a feature  $y$  represents a scene 3D point  $i$  defined by the 6-dimension state vector:

$$y_i = [x_i, y_i, z_i, \theta_i, \phi_i, \rho_i]^T \quad (5)$$

which models the 3D point located at:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i, \phi_i) \quad (6)$$

where  $x_i, y_i, z_i$  are the camera optical center coordinates when the feature was first observed; and  $\theta_i, \phi_i$  represent azimuth and elevation (respect to the world reference  $W$ ) for the directional vector  $m(\theta_i, \phi_i)$ . The point depth  $d_i$  along the ray is coded by its inverse  $\rho_i = 1/d_i$ .

The different locations of the camera, along with the location of the already mapped features, are used to predict the feature position  $h_i$ . The observation of a point  $y_i$  from a camera location defines a ray expressed in the camera frame as  $h^C = [h_x, h_y, h_z]$ :

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

$h^C$  is observed by the camera through its projection in the image. The projection is modeled using a full perspective wide angle camera. First the projection is modeled in the normalized retina:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{h_x}{h_z} \\ \frac{h_y}{h_z} \end{pmatrix} \quad (8)$$

The camera calibration model is applied to produce the pixel coordinates for the predicted point:

$$h = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - \frac{f}{d_x} v \\ v_0 - \frac{f}{d_y} v \end{pmatrix} \quad (9)$$

where  $u_0, v_0$  is the camera center in pixels,  $f$  is the focal length and  $d_x, d_y$  the pixel size. Finally, a radial distortion model is applied (Davison, 2004).

$$h_i = \begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} \frac{u - u_0}{\sqrt{1 + 2K_1 r^2}} + u_0 \\ \frac{v - v_0}{\sqrt{1 + 2K_1 r^2}} + v_0 \end{pmatrix} \quad (10)$$

where  $r = \sqrt{(u - u_0)^2 + (v - v_0)^2}$ , and  $K_1$  is the distortion coefficient.

Features search is constrained to elliptical regions around the predicted  $h_i$ . The elliptical regions are defined by the innovation covariance matrix  $S_i = H_i P_{k+1} H_i^T + R$  where  $H_i$  is the Jacobian of the sensor model with respect to the state,  $P_{k+1}$  is the prior state covariance, and measurements  $z$  assumed corrupted by zero mean Gaussian noise with covariance  $R$ .

## 4. DELAYED FEATURE INITIALIZATION

### 4.1 Candidate points

In our work we consider a minimum number of features  $y_i$  to be predicted appearing in the image, otherwise new features have to be added to the map. In this latter case, new points are detected in the image with a saliency operator. Specifically, we use Harris corner detector, although more robust detectors can be used. If the data association problem want to be addressed in a more robust way, features

descriptors could be used, in previous work (Munguía, 2006 a,b) we treat this problem. Only areas in the image free of previously detected points or features already mapped are consider for detecting new points, we call these points in the image that do not have to be added to the map as candidate points,  $\lambda$ . When a point is first detected by the saliency operator in a frame  $k$ , the candidate point is conformed by:

$$\lambda_i = (x_i, y_i, z_i, \sigma_i^x, \sigma_i^y, \sigma_i^z, q_i^0, q_i^1, q_i^2, q_i^3, \sigma_i^{q^0}, \sigma_i^{q^1}, \sigma_i^{q^2}, \sigma_i^{q^3}, u_i, v_i) \quad (11)$$

The values  $x_i, y_i, z_i$  represent the camera optical center position,  $\sigma_i^x, \sigma_i^y, \sigma_i^z$  their associated variances taken from the state covariance matrix  $P_k$ .  $q_i^0, q_i^1, q_i^2, q_i^3, \sigma_i^{q^0}, \sigma_i^{q^1}, \sigma_i^{q^2}, \sigma_i^{q^3}$  is the quaternion representing the current camera orientation and its associated variances taken from the state covariance matrix  $P_k$ , and  $u_i, v_i$  is the current pixel coordinates for the point  $\lambda_i$ . In subsequent frames  $\lambda_i$  is tracked, but practically some  $\lambda_i$  points can not be tracked. This process is used for pruning weakest image features. For tracking purposes any method can be used. The tracking for every candidate point  $\lambda_i$  is realized until is pruned or initialized in the system. In practice for every frame, some new candidate points  $\lambda_i$  could be detected, others points could be pruned and others could be considered to be added to the map. In our experiments an average of 5 to 15 points  $\lambda_i$  are maintained at every step.

4.2 Adding features to the state

As the camera freely moves through its environment, the translation produces parallax in features. Parallax is really the key that allows to estimating features depth. In the case of indoor sequences, centimeters are enough to produce parallax, on the other hand, the more distant the features, the more the camera have to be translated to produce parallax.

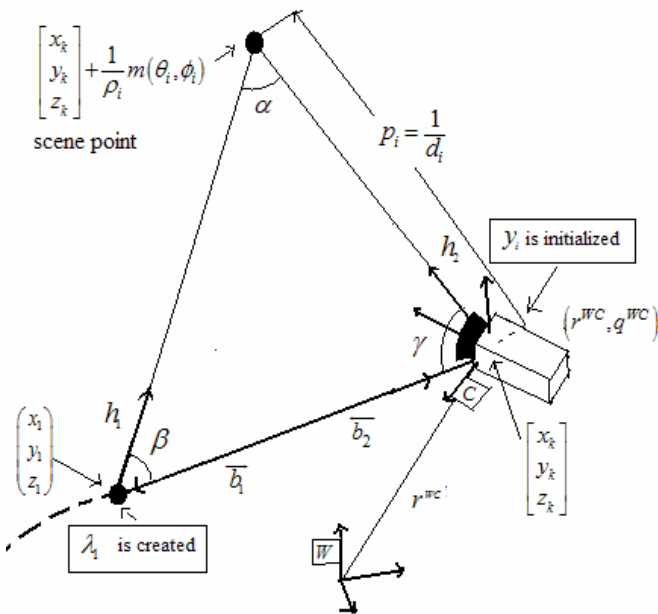


Fig. 1. Feature parametrization and initialization.

In our approach we want dynamically to estimate an initial depth and its associated uncertainty for the features added to

the map. For near features, a small translation is enough to reproduce some parallax. We use a minimum parallax threshold  $\alpha_{min}$  for considering a candidate point  $\lambda_i$  to be added to the map as a feature  $y_i$ . On the other hand distant features will not produce parallax but are useful to estimate the camera orientation, and therefore it is advantageous to include some distant features in the map with big depth uncertainty. Then, a minimum base-line camera translation  $|b|_{min}$  is also considered for adding a candidate point  $y_i$  to the map. Fig. 2 shows a simulation for decrementing uncertainty in feature depth estimation respect with the increase of parallax angle. It can be observed that a few parallax degrees are enough for reducing significantly the depth uncertainty. In the experiments  $\alpha_{min} = 3$  is used. The minimum base-line  $b_{min}$  was heuristically established to be the base-line necessary to produce a parallax  $\alpha \approx 6^\circ$  in the initial reference points. For example if the camera initial position is in average one meter away from the initial reference points then  $b_{min} = 8cm$ .

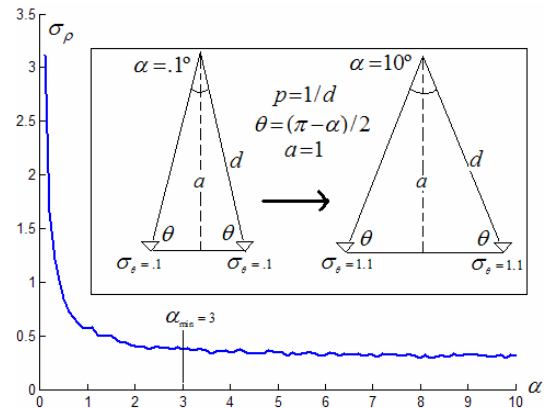


Fig 2. Estimate simulation of uncertainty feature depth  $\sigma_p$  for parallax angle  $\alpha$  from  $0.1^\circ$  to  $10^\circ$ . An increment in the uncertainty  $\sigma_\theta$  of the measurement angle  $\theta$  is considered as the parallax grows. Note that a few degrees parallax is enough to reduce the uncertainty in the estimation.

So far, the uncertainty of the measurements is not considered, and the parallax  $\alpha$  is estimated using i) the base-line  $b$ , ii)  $\lambda_i$  using its associated data  $(x_i, y_i, z_i, q_i^0, q_i^1, q_i^2, q_i^3, u_i, v_i)$ , and iii) the current state  $(x_k, y_k, z_k, q_k^0, q_k^1, q_k^2, q_k^3, u_k, v_k)$ .

The parallax angle for a  $\lambda_i$  can be estimated (Fig. 1):

$$\alpha = \pi - (\beta + \gamma) \quad (12)$$

The angle  $\beta$  is determined by the directional projection ray vector  $h_1$  and the vector  $b_1$  defining the base-line  $b$  in the direction of the camera trajectory by:

$$\beta = \cos^{-1} \left( \frac{h_1 \cdot b_1}{\|h_1\| \|b_1\|} \right) \quad (13)$$

where the directional projection ray vector  $h_1$  expressed in the absolute frame, is computed from the camera position and the coordinates of the observed point when it was first observed, using the data stored in  $\lambda_i$

$$h_1 = R_{wC}(q_1^{wC})h_1^C \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} \quad (14)$$

with  $R_{wC}(q_1^{wC})$  being the rotation matrix depending on the stored camera orientation quaternion  $q_1^{wC} = (q_1^0, q_1^1, q_1^2, q_1^3)$  and  $h_1^C(u_1, v_1)$  is the directional vector in the camera frame using equation 7.  $b_1$  is the vector representing the camera base-line  $b$  between the camera optical center position  $x_1, y_1, z_1$  where the point was first observed and the current optical center  $x_k, y_k, z_k$ .

$$b_1 = [(x_k - x_1), (y_k - y_1), (z_k - z_1)] \quad (15)$$

The angle  $\gamma$  is determined in a similar way as  $\beta$  but using the directional projection ray vector  $h_2$  and the vector  $b_2$  defining the base-line in the opposite direction of the camera trajectory by:

$$\gamma = \cos^{-1} \left( \frac{h_2 \bullet b_2}{\|h_2\| \|b_2\|} \right) \quad (16)$$

The directional projection ray vector  $h_2$  expressed in the absolute frame, is computed in a similar way as (14) but using current camera position  $x_v$  and points coordinates  $u, v$ .  $b_2$  is equal to  $b_1$  but pointing to the opposite direction:

$$h_2 = R_{wC}(q_k^{wC})h_k^C \begin{pmatrix} u \\ v \end{pmatrix} \quad (17)$$

$$b_2 = [(x_1 - x_k), (y_1 - y_k), (z_1 - z_k)] \quad (18)$$

The base-line  $b$  is the module of  $b_2$  or  $b_1$  :

$$b = \|b_2\| = \|b_1\| \quad (19)$$

If  $\alpha > \alpha_{min}$  or  $b > b_{min}$  then  $\lambda_i$  is initialized as a new feature map:

$$\hat{y}_i = [\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{\theta}_i, \hat{\phi}_i, \hat{\rho}_i]^T \quad (20)$$

where the three first elements are obtained directly from the current camera optical center position:

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} \quad (21)$$

The angles can be derived as:

$$\begin{pmatrix} \hat{\theta}_i \\ \hat{\phi}_i \end{pmatrix} = \begin{pmatrix} \arctan \left( -h_2^y, \sqrt{h_2^x{}^2 + h_2^z{}^2} \right) \\ \arctan \left( h_2^x, h_2^z \right) \end{pmatrix} \quad (22)$$

where  $h_2 = [h_2^x, h_2^y, h_2^z]$  is obtained from equation 17. Finally the inverse depth  $\rho_i$  is derived from the sine law

$$\hat{\rho}_i = \frac{\sin \alpha}{b * \sin \beta} \quad (23)$$

#### 4.3 Updating the covariance matrix

The covariance for  $\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{\theta}_i, \hat{\phi}_i$  and  $\hat{\rho}_i$  is derived from the error diagonal covariance matrix  $R_j$  measurement and the state covariance matrix estimate  $P_k$ .

$$R_j = \text{diag}(\sigma_{u2}^2, \sigma_{v2}^2, \sigma_{u1}^2, \sigma_{v1}^2, \dots, \sigma_1^x, \sigma_1^y, \sigma_1^z, \sigma_1^{q^0}, \sigma_1^{q^1}, \sigma_1^{q^2}, \sigma_1^{q^3}) \quad (24)$$

$R_j$  is conformed with the image measurement error variance  $\sigma_{u2}^2, \sigma_{v2}^2, \sigma_{u1}^2, \sigma_{v1}^2$  and the variances stored in  $\lambda_i$   $\sigma_1^x, \sigma_1^y, \sigma_1^z, \sigma_1^{q^0}, \sigma_1^{q^1}, \sigma_1^{q^2}, \sigma_1^{q^3}$ . The state covariance matrix after initialization is:

$$P_k^{new} = J \begin{pmatrix} P_k & 0 \\ 0 & R_j \end{pmatrix} J^T \quad (25)$$

$$J = \begin{pmatrix} I & 0 \\ \frac{\partial y}{\partial x_v}, 0, \dots, 0, & \frac{\partial y}{\partial h_{R_j}} \end{pmatrix} \quad (26)$$

where  $I$  is the identity matrix with the same dimension of  $P_k$ .  $\partial y / \partial x_v$  are the derivatives of  $y_i$  with respect to the state  $x_v$  and  $\partial y / \partial h$  the derivatives of  $y_i$  with respect to measurement equations depending on  $R_j$ . The Jacobian calculation is complicated but a tractable matter of differentiation; we do not present the results here.

## 5. EXPERIMENTAL RESULTS

Real image sequences of  $320 \times 240$  pixels acquired with a monochrome IEEE1394 web-cam camera at 30 fps was used for test the performance of the method. The experiments were developed in MatLab. The part of code related with section 2 was based in the code provided by the author of (Montiel 2006a). The initial reference consists in three spatial points forming a triangle of known dimensions, (see Fig. 3 and 4). Prior to start the first Kalman step, these three points are selected on the image, then their 3D position respect to the camera are calculated using an optimization technique, and finally included in the system state with zero uncertainty.

Several image sequences moving the camera through different trajectories were recorded following a predefined path. The undelayed and delayed initialization has been compared. The trajectories were designed in order that if a feature is left behind by the movement of the camera, this feature will not appear in image again in subsequent frames.

The original method have a drawback when a initial metric reference is used; if the features are initialized with an initial distance close to the optical center with respect to the distance to the reference points, the features never converge respect to the reference, and even the Kalman Filter never converges to an unscaled trajectory.

Figure 3 illustrates the initialization of the first features after the three reference points are introduced in the system for the undelayed and delayed method. The graphics in the center show the undelayed method for an initial feature depth of 50cm, in frame 2 (central upper), it is possible to observe that reference points are located approximate 80 cm from the initial camera position and the first observed points are immediately initialized. However at frame 320 (central lower) the mapped features never converge respect to the metric reference.

Sequence	Method	$\sigma_{x,y,z}$	Nf	%c	Nfc	E	Nf<0
1	Undelayed	4	47	42	114	5.7	0
	Delayed	4.1	35	27	110	1.44	0
2	Undelayed	2.1	46	76	36	11.2	0
	Delayed	2.4	28	82	45	9.12	0
3	Undelayed	1.4	34	44	45	17	2
	Delayed	2.5	27	55	58	19	1

Table 1. The results at the end of the three sequences: ( $\sigma_{x,y,z}$ ): Summed standard deviation for the  $x,y,z$ , position of the camera. (Nf): Total number of features added to the system. (%c): Percentage of features that present convergence. (Nfc): The average number of frames needed for the convergence of the features. (E): The metric error distance in cm from the real to final estimated trajectory. (Nf<0) Number of negative inverse depth estimated at the final of the trajectory.

Camera trajectory either converge, note the 4 points corresponding to the printer located besides the initial three point reference. On the other hand when we use an initial depth equal to 60 cm, (right upper and lower graphics) the map and camera trajectory converge reasonably.

In delayed approach (left graphics) the first feature is added to the map until frame 125, in this case with a huge initial uncertainty (upper left graphic). However at frame 320 (lower left graphic) the map and trajectory converges. Note that the first added feature was initialized very near to its final position, and its uncertainty was minimized.

The condition for detecting new points with the Harris corner detector for both methods is applied if the number of actives features in image goes below 30, in this case the detector is applied over the free features image regions.

Figure 4 shows the results for three different sequences. Real final camera position and trajectory was manually added to the graphics (in black) to make easier the comparison, the initial and final frames are illustrated in the center for each sequence.

Table 1 shows the results for each sequence for the next aspects. In our experiments we consider that a feature converges when its depth uncertainty  $\sigma$  represents less than 5% of its depth, in this way we consider a convergence measurement proportional to the distance. The depth of a near feature should be estimated in a more accurate manner than a distant feature.

## 6. CONCLUSIONS

In this work a method for delayed features initialization for inverse depth parametrization in monocular SLAM is presented. The experimental results show that this method can be a good choice when using monocular SLAM. The method seems to be more robust respect to the undelayed method, when initial metric reference points are used for scaling the map.

In our experiments the resulting camera trajectory estimate using the delayed method was similar to the estimate by the undelayed method. In aspects relating with features depth convergence the results were similar for both methods. Since the delayed method is more restrictive for adding new

features, a reduced percentage of new features are added to the map (20-40%) respect to the undelayed method, without losing the quality of the map. This aspect is desirable, because bigger environments can be mapped with the same number of features. On the other hand is clear that an additional computational cost is added in the delayed method, since the candidate points have to be tested in order to be added to the map. The Jacobian to estimate the new covariance matrix is more complex respect to one used in the undelayed method. On the other hand is known that Kalman filter computation cost scales poorly with the size of the state, and the saving computational cost using 20-40% of the total amount of features can be higher than the computational cost added in the delayed method.

## REFERENCES

- Davison, A. J. Kita N. (2001) *Sequential localization and map-building for real-time computer vision and robotics*. Robotics and Autonomous systems
- Davison, A. J. (2003) *Real-time simultaneous localization and mapping with a single camera*. In Proc. International Conference on Computer Vision.
- Davison, A. J. Gonzales, Y. Kita N. (2004) *Real-Time 3D SLAM with Wide-Angle Vision*. In Proc. of Symposium on Intelligent Autonomous Vehicles.
- Eade, E. Drummond, T. (2006) *Scalable monocular SLAM*. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition.
- Hartley, R. I. Zisserman, A. (2004) *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- Lemaire, T. Lacroix, S. Sola J. (2005) *A practical 3D Bearing-Only SLAM algorithm*. In Proc. International Conference on Intelligent Robots and Systems.
- Montiel, J.M.M. Civera, J. Davison, A.J. (2006) *Unified Inverse Depth Parametrization for Monocular SLAM*. Robotics: Science and Systems Conference.
- Montiel, J.M.M. Davison, A.J. (2006) *A visual compass based on SLAM*. In Proc. Intl. Conf. on Robotics and Automation
- Munguia, R. Grau, A. Sanfeliu A. (2006) *Matching Images Features in a Wide Base Line with ICA Descriptors*. In Proc. of the IEEE International Congress in Pattern Recognition, ICPR
- Munguia, R. Grau, A. (2006) *Learning Variability of Image Feature Appearance Using Statistical Methods*. Lecture Notes in Computer Science, 4225

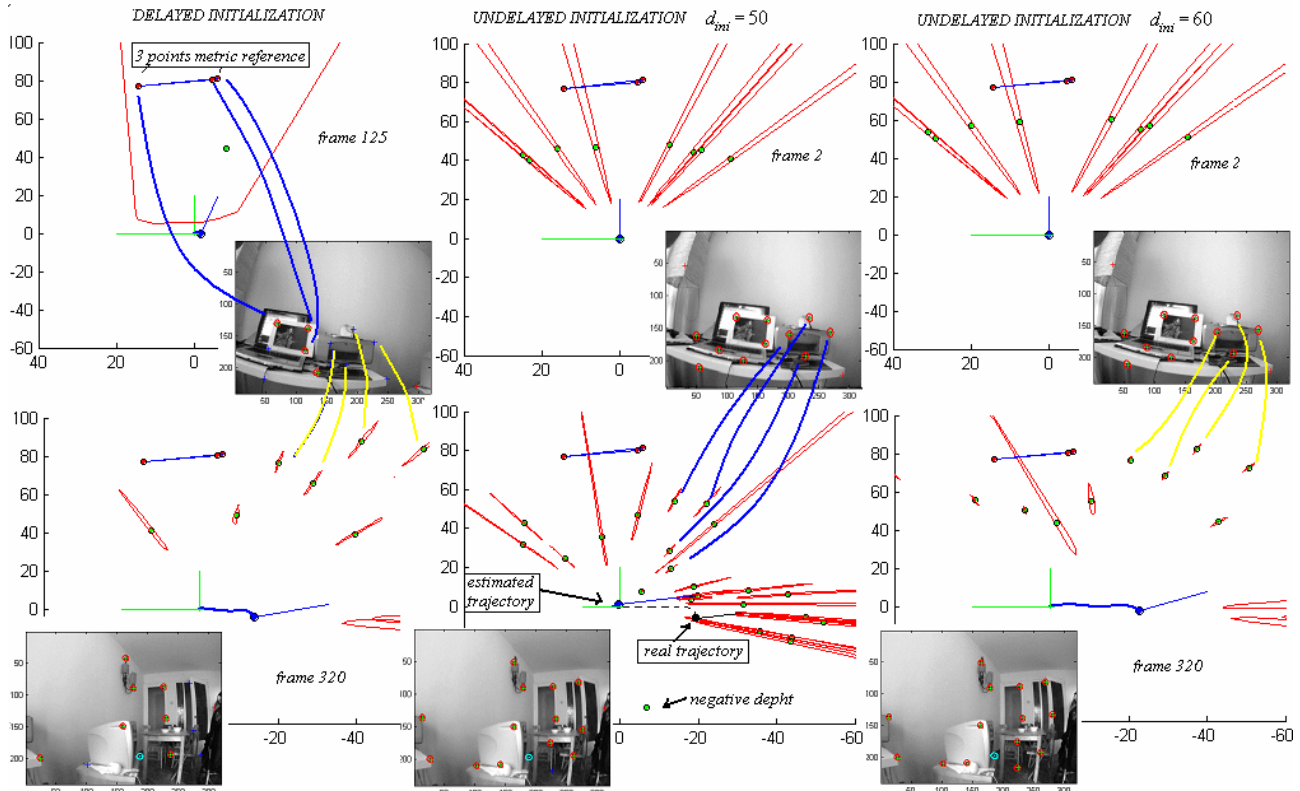


Figure 3. Delayed and undelayed methods, using three point reference to establish metric scale. The features positions are represented by green solid circles and their uncertainty by red ellipses. The camera position is represented by a blue solid circle and its orientation by a blue line emerging from the camera position. The camera trajectory is indicated with the blue path from the initial ( $x=0$   $z=0$ ) to the final camera position. For simplicity all the maps are viewed in  $x$ - $z$  axes.

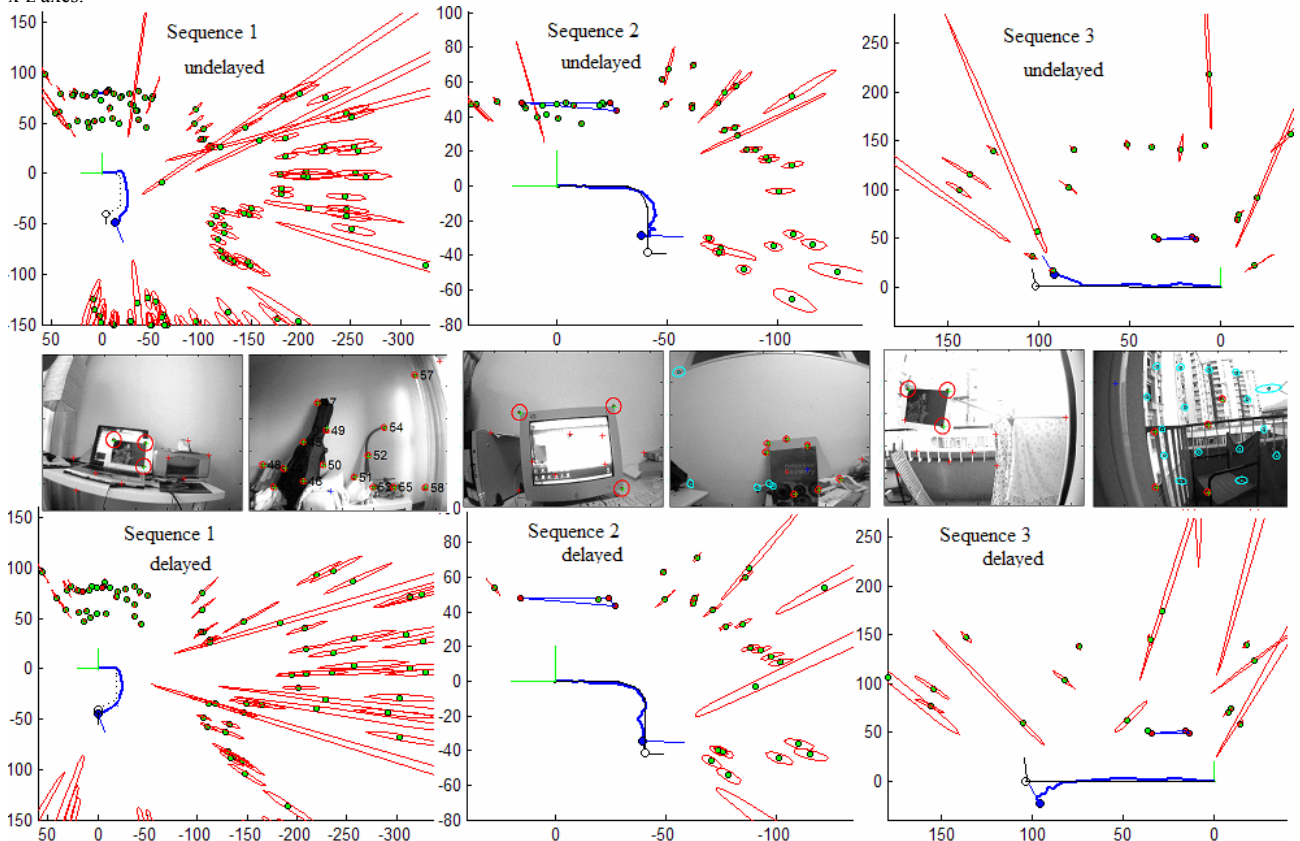


Figure 4. Camera trajectory and map for three sequences. Undelayed method (upper graphics) and delayed method (lower graphics). The first sequence corresponds to 760 frames of a house livingroom and it is the same sequence used in the previous experiment. The second sequence corresponds to 480 frames taken in a laboratory. Note that a PC monitor was used as initial metric reference. The third 360-frame sequence was taken following a simple linear path, but in a more occluded terrace building environment, with very near and very distant features.