2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Visual Odometry Priors for robust EKF-SLAM

Pablo F. Alcantarilla, Luis M. Bergasa, Frank Dellaert

Abstract— **One of the main drawbacks of standard visual EKF-SLAM techniques is the assumption of a general camera motion model. Usually this motion model has been implemented in the literature as a constant linear and angular velocity model. Because of this, most approaches cannot deal with sudden camera movements, causing them to lose accurate camera pose and leading to a corrupted 3D scene map. In this work we propose increasing the robustness of EKF-SLAM techniques by replacing this general motion model with a visual odometry prior, which provides a real-time relative pose prior by tracking many hundreds of features from frame to frame. We perform fast pose estimation using the two-stage RANSAC-based approach from [1]: a two-point algorithm for rotation followed by a one-point algorithm for translation. Then we integrate the estimated relative pose into the prediction step of the EKF. In the measurement update step, we only incorporate a much smaller number of landmarks into the 3D map to maintain real-time operation. Incorporating the visual odometry prior in the EKF process yields better and more robust localization and mapping results when compared to the constant linear and angular velocity model case. Our experimental results, using a handheld stereo camera as the only sensor, clearly show the benefits of our method against the standard constant velocity model.**

I. Introduction

One of the most successful real-time monocular SLAM systems (MonoSLAM) was introduced by Davison *et al.* [2]. In this approach, camera poses and an incremental map of 3D landmarks are computed using a standard Extended Kalman Filter (EKF). Ever since the seminal work by Broida et al. in the early nineties [3], [4], EKF-SLAM strategies have been widely used and have been improved significantly in different aspects such as: undelayed initialization of features [5], moving objects avoidance [6] and automatic re-localisation [7]. With monocular SLAM, recovering the scale of a map is one of the main limitations due to observability problems in recovering 3D information from 2D projections. Stereo sensors are an appealing alternative, since they directly provide the scale of a point using the information from the two camera views. EKF-SLAM strategies have been applied successfully to stereo vision in large environments, for example in [8][9]. Although good results are obtained due to the use of sub-mapping strategies, these approaches cannot handle sudden camera motions since both of them use a constant velocity model. We hypothesize that results can be improved considerably if a better motion model is used.

In common EKF-SLAM strategies, the state vector X comprises of camera pose and 3D landmarks and it is updated in two consecutive ways:

- Prediction: For modeling the camera motion between two consecutive frames, use a general motion model to predict the camera pose in the next frame.
- Update: given the predicted camera pose, search for matches in a high probability search area, and update the state of the filter for matched features.

The use of a generic constant linear and angular velocity motion model that assumes smooth camera motion as in [2] is an important drawback, since this general motion model cannot deal properly with sudden movements. If camera motion is not smooth enough the pose can easily get lost, corrupting the quality of the 3D reconstruction.

In order to improve the robustness of Davison's MonoSLAM, different authors have proposed alternatives to replace this smooth motion model. Williams *et. al* presented in [10] a relocalisation algorithm for monocular SLAM that increases robustness against camera shakes and occlusions. In [11] MonoSLAM's robustness to sudden or erratic camera motions is improved by means of a high-frame rate camera (200 Hz) in conjunction with an extended motion model, taking into account linear and angular velocities and accelerations. This extended motion model provides additional robustness against handheld jitter when throwing or shaking the camera.

Klein and Murray [12] show a fast method of estimating camera rotation between two frames for monocular SLAM, assuming that the camera is purely rotating between frames. In order to estimate rotation between two frames, a comparison between subsampled and blurred images is performed. Using second order minimization techniques they obtain a good pose prior for landmark tracking. With this simple pose estimation and the addition of *edgelets* they increase the robustness of monocular SLAM under fast camera rotations and motion blur.

Our approach is related to the previous one, and considers how visual odometry schemes can be used in conjunction with EKF-SLAM to provide accurate camera pose priors that can be incorporated into the

Pablo F. Alcantarilla and Luis M. Bergasa are with Department of Electronics, University of Alcalá. Alcalá de Henares, Madrid, Spain. e-mail: pablo.alcantarilla, bergasa@depeca.uah.es
Frank Dellaert is with School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. e-mail: dellaert@cc.gatech.edu

EKF process. Visual odometry is a well-known technique to estimate the relative camera motion between two consecutive frames. Nistér *et al.* [13] presented one of the first real-time visual odometry systems using a five and three-point algorithm, respectively for monocular and stereo vision. A drawback of visual odometry is the drift of the pose estimate over time. Taking into account long-range constraints or appearance descriptors, this drift can be reduced as shown in [14][15].

In this paper we propose to use visual odometry priors in conjunction with EKF-SLAM showing that this improves the robustness of localization considerably, satisfying real-time demands. It is important to notice that in EKF-SLAM only a small number of highly textured features are part of the map. Typically no more than 15 features are tracked every frame, which means that there is still a lot of useful information in the image that can be used for a more robust pose estimation.

We have done our experiments considering stereo vision, although the basis of our method can be adapted to the monocular case assuming that the camera is purely rotating between frames (e.g. [12]). First, we extract features from consecutive frames and find the matches between them using a fast multi-scale optical flow algorithm [16]. After this, we run a two-stage visual odometry similar as the one proposed by Kaess *et. al* [1], separating the flow between far and close features for recovering rotation and translation respectively using a 2-point and 1-point RANSAC procedure. This algorithm can also deal with nearly degenerate situations. Finally we incorporate this pose estimate in the prediction step of EKF-SLAM replacing the constant linear and angular velocity motion model.

All the processing can be done in no more than 10 ms, satisfying the real-time constraints of handheld visual SLAM approaches. The only computations that are necessary are: feature extraction, i.e. normally finding corners in the image which can be done in less than 4 ms with the method described in [17], multi-scale optical flow, and pose estimation. For the stereo vision case, it is necessary to find stereo correspondences between the left and the right view for every frame. This can be an important computational burden, although there are some fast stereo implementations for obtaining the disparity map such as [18], [19] that are amenable to a GPU implementation. Also, many commercial stereo sensors provide disparity map implementations on-chip.

Our algorithm can work using a standard 30Hz camera providing robust localization results without any need of using a costly 200Hz camera as in [11]. In addition our algorithm is fully integrated into the EKF process and keeps real-time demands as contrary to the work of [10].

The paper is organized as follows: In Section II the two-stage stereo visual odometry is briefly discussed. In Section III we show how to incorporate our visual odometry priors into EKF-SLAM. Finally we show some experimental results and conclussions in Sections IV and V respectively.

II. Stereo Visual Odometry by Sparse Flow Separation

For camera pose estimation between frames, we take a similar approach as the one described in [1]. The algorithm has the following steps:

1) Perform sparse stereo and putative matching
2) Separate features based on depth threshold
3) Recover rotation with two-point RANSAC
4) Recover translation with one-point RANSAC

A. Sparse Stereo and Putative Matching

Features are extracted in the current frame and stereo correspondences between the left and right views of the stereo pair are established by means of a correlation search. For finding the putatives between two consecutive frames, a fast implementation of multi-scale KLT tracker has been used, as described in [16]. This algorithm typically provides enough matches to estimate camera motion between two consecutive frames with a high level of accuracy.

B. Features Separation based on depth threshold

For selecting which features are more useful for estimating the rotation and translation components between two consecutive frames, we separate the set of putative matches in two sets: $M = \{M_{Rot}, M_{Trans}\}$ according to a depth threshold θ. This threshold takes into account the maximum expected camera velocity, camera frame rate, camera intrinsics and extrinsics. For a better explanation about how to obtain this threshold can be seen in [1].

C. Rotation recovery: Two-point RANSAC

The rotational component of the camera motion is computed based on the set of putative matches that are not influenced by the translation, i.e. those points that can be considered to be at infinity. Considering pure rotation ($t = 0$),

$$R_{k-1}^{k} = \underset{R_{k-1}^{k}}{\arg\min} \sum_{i,\tau \in \{k,k-1\}} \left\| Z_{i,\tau}^{R} - v^{R}\left(R^{\tau}, X_i\right) \right\|^2 \quad (1)$$

where R_{k-1}^{k} is the rotation matrix between two consecutive frames parametrized by a unit quaternion, $Z_i = (u_L, v_L)^t$ is the monocular projection of a 3D point from the putative matches set onto the left image, $X_i = (x_i, y_i, z_i)^t$ represents the 3D coordinates of one of the putative matches in the camera coordinate frame and $v^{(R)}(R, X)$ is the monocular projection of a 3D point given the rotation matrix R.

The putative matches dataset contain outliers, therefore a random sample consensus (RANSAC) is used in order to obtain a robust model as described in [20]. Since a rotation matrix has only 3DoF and each of the points yields two constraints, only two points are necessary for the rotation estimate. From the set with smallest reprojection error, the set of inliers is computed and

all the inliers are used for a refinement of the rotation estimate, obtaining the final rotation estimate $\widehat{R}$.

D. Translation recovery: One-point RANSAC

Based on the camera rotation estimate $\widehat{R}$, the translation is recovered only from the set of putative matches that are close to the camera.

$$\mathbf{t}_{k-1}^{k} = \underset{\mathbf{t}_{k-1}^{k}}{\arg\min} \sum_{i,\tau\in\{k,k-1\}} \left\| Z_{i,\tau}^{\mathbf{t}} - v^{\mathbf{t}}\left(\widehat{R}_{\tau}, \mathbf{t}_{\tau} - X_i\right)\right\|^2 \tag{2}$$

where t_{k-1}^{k} is the translation vector between two consecutive frames, $Z_i = (u_L, v_L, u_R, v_R)^t$ are the stereo projections of a 3D point in the left and right camera respectively, $X_i = (x_i, y_i, z_i)^t$ represents the 3D coordinates of the point relative to the camera coordinate frame and $v^t(R, t, X)$ is the stereo projection of a 3D point given the rotation matrix R and the translation vector t. Since the translation vector has three degrees of freedom, and each of the points yields three constraints (if the stereo is rectified $v_L = v_R$), only one point is necessary for the translation estimate. Again a RANSAC procedure is performed obtaining the final translation estimate from the set of all the inliers.

E. Rotation and Translation recovery: 3-point RANSAC

Depending on the application and the environment, it may happen that there are not enough putative matches for accurate rotation or translation estimates. A common case is where the camera is facing a wall in an indoor environment, in which case all the putative matches are close to the camera. For those occasions the standard three-point algorithm can be used for estimating simultaneously the rotation and camera translation, according to the next equation:

$$R, \mathbf{t}_{k-1}^{k} = \underset{R,\mathbf{t}_{k-1}^{k}}{\arg\min} \sum_{i,\tau\in\{k,k-1\}} \left\| Z_{i,\tau}^{\mathbf{t}} - v^{\mathbf{t}}\left(K\left[R_{\tau}\ \mathbf{t}_{\tau}\right] X_i\right)\right\|^2 \tag{3}$$

III. Visual Odometry priors for EKF SLAM

In MonoSLAM, it is assumed that the camera linear and angular velocities may change in every frame, but they are expected to be constant on average. Using visual odometry priors, we can have any kind of 6DoF camera motion, since we recover accurately the relative camera motion between two consecutive frames. The camera vector state comprises of:

$$\mathbf{X_v} = \left(r_{cam}^{W},\ q_{cam}^{WC},\ v_{cam}^{W},\ \omega_{cam}^{C}\right)^t \tag{4}$$

where the vectors r_{cam}^{W} and v_{cam}^{W} encode the 3D metric position and linear velocity of the camera with respect to the world coordinate frame W, q_{cam}^{W} represents the orientation of the camera by an unit quaternion with respect to the world coordinate frame W and the current camera frame C, and ω_{cam}^{C} encodes the angular velocity estimated in the camera frame.

In each time step we have a certain error in the pose estimate, obtained from a non-linear least squares minimization problem. Hence, the noise vector will be a function of this pose estimate and have a certain covariance given by the residuals of the minimization problem. The noise vector can be expressed as

$$\mathbf{n} = \begin{pmatrix} r_{vo} \\ q_{vo} \end{pmatrix} = \begin{pmatrix} r_{k-1}^{k} \\ q_{k-1}^{k} \end{pmatrix} \tag{5}$$

with an associated covariance matrix P_n. The covariance matrix of the noise vector P_n can be derived from the estimated regression coefficients of the nonlinear model estimate. For more details about how to obtain this covariance matrix, see the implementation notes in [21]. The constant velocity model in MonoSLAM assumes this covariance matrix to be diagonal, representing uncorrelated noise in linear and rotational components. In our approach the recovered covariance matrix is not necessary diagonal, since usually translation and rotational components are correlated. In MonoSLAM approach, the camera state update is computed as follows:

$$\left.\begin{array}{rcl} r_{new}^{W} &=& r_{old}^{W} + \left(v^{W} + V^{W}\right)\cdot\Delta t \\ q_{new}^{WC} &=& q_{old}^{WC} \times q\left[(\omega + \Omega)\cdot\Delta t\right] \\ v_{new}^{W} &=& v_{cam} + V \\ \omega_{new}^{C} &=& \omega_{cam} + \Omega \end{array}\right\} \tag{6}$$

We can express in Eq. 6 the new camera pose as a function of the computed visual odometry priors:

$$\left.\begin{array}{rcl} r_{new}^{W} &=& r_{old}^{W} - R^{WC}\cdot r_{k-1}^{k} \\ q_{new}^{WC} &=& q_{old}^{W} \times q_{k}^{k-1} \end{array}\right\} \tag{7}$$

where r_{k-1}^{k} is the camera translation and q_{k-1}^{k} is the quaternion representing the rotation between frames k and $k-1$, and obtained from R_{k-1}^{k}. The rotation matrix R^{WC} represents the rotation between the current camera frame and the world coordinate frame. This rotation matrix is directly obtained from the quaternion q_{new}^{WC}. The process noise covariance Q_v can be expressed as a function of the non-linear state update function f_v and the noise vector covariance matrix P_n as:

$$\mathbf{Q_v} = \frac{\partial f_v}{\partial n}\cdot P_n \cdot \left(\frac{\partial f_v}{\partial n}\right)^t \tag{8}$$

where the Jacobian $\frac{\partial f_v}{\partial n}$ is computed as follows:

$$\frac{\partial f_v}{\partial n} = \begin{pmatrix} \frac{\partial r_{new}^{W}}{\partial r_{vo}} & \frac{\partial r_{new}^{W}}{\partial q_{vo}} \\ 0 & \frac{\partial q_{new}^{WC}}{\partial q_{vo}} \\ \frac{\partial v_{new}^{W}}{\partial r_{vo}} & \frac{\partial v_{new}^{W}}{\partial q_{vo}} \\ 0 & \frac{\partial \omega_{new}^{C}}{\partial q_{vo}} \end{pmatrix} \tag{9}$$

Considering Eq. 7, the Jacobians in (9) are computed as follows:

$$\frac{\partial r_{new}^{W}}{\partial r_{vo}} = -R^{WC} \tag{10}$$

$$\frac{\partial r^W_{new}}{\partial q_{vo}} = -\frac{\partial R^{WC}}{\partial q^{k-1}_k} \cdot r^k_{k-1} \cdot \frac{\partial q^{k-1}_k}{\partial q_{vo}} \quad (11)$$

$$\frac{\partial q^{WC}_{new}}{\partial q_{vo}} = \frac{\partial q}{\partial q^{k-1}_k} \cdot \frac{\partial q^{k-1}_k}{\partial q^k_{k-1}} \quad (12)$$

$$\frac{\partial v^W_{new}}{\partial r_{vo}} = -\frac{R^{WC}}{\Delta T} \quad (13)$$

$$\frac{\partial v^W_{new}}{\partial q_{vo}} = -\frac{\partial R^{WC}}{\partial q^{k-1}_k} \cdot r^k_{k-1} \cdot \frac{\partial q^{k-1}_k}{\partial q_{vo}} \cdot \frac{1}{\Delta T} \quad (14)$$

$$\frac{\partial q^{WC}_{new}}{\partial r_{vo}} = \frac{\partial \omega^C_{new}}{\partial r_{vo}} = 0 \quad (15)$$

The computation of the Jacobian $\frac{\partial \omega^C_{new}}{\partial q_{vo}}$ is not as direct as the others, but can be computed easily taking into account some quaternion properties. Given an axis of rotation $\omega \cdot \Delta T = (\theta_x, \theta_y, \theta_z)$ the unit quaternion that represents this rotation is:

$$\left.\begin{array}{rcl} q_0 & = & \cos(\frac{\beta}{2}) \\ q_x = \sin(\frac{\beta}{2}) \cdot \frac{\theta_x}{\beta} & = & \sin(\frac{\beta}{2}) \cdot \frac{\omega_x \cdot \Delta T}{\beta} \\ q_y = \sin(\frac{\beta}{2}) \cdot \frac{\theta_y}{\beta} & = & \sin(\frac{\beta}{2}) \cdot \frac{\omega_y \cdot \Delta T}{\beta} \\ q_z = \sin(\frac{\beta}{2}) \cdot \frac{\theta_z}{\beta} & = & \sin(\frac{\beta}{2}) \cdot \frac{\omega_z \cdot \Delta T}{\beta} \end{array}\right\} \quad (16)$$

where $\beta = \sqrt{\theta_x^2 + \theta_y^2 + \theta_z^2}$. This normalization is necessary in order to represent a valid unitary axis of rotation. If we take the approximation that $\omega \cdot \Delta T \approx 0$ we can simplify the trigonometric expressions from Eq. 16 and obtain the final expression of the desired Jacobian:

$$\frac{\partial \omega^C_{new}}{\partial q_{vo}} = \begin{pmatrix} 0 & \frac{2}{\Delta T} & 0 & 0 \\ 0 & 0 & \frac{2}{\Delta T} & 0 \\ 0 & 0 & 0 & \frac{2}{\Delta T} \end{pmatrix} \quad (17)$$

IV. Results and Discussion

In our experiments the acquisition frame-rate is set to 30 f.p.s. and the image resolution is 320×240 pixels. The stereo baseline is 15 cm. Since the camera is carried in hand, maximum expected velocities are in the range of common velocities of a person walking in the range between ($3\ Km/h - 5\ Km/h$). Considering these values we use a depth threshold $\theta = 5.70\ m$ for sparse stereo flow separation. The approximation $\omega \cdot \Delta T \approx 0$ for obtaining the Jacobian values in Eq. 17 is valid for our experiments, since the stereo pair is carried in hand by a normal person walking speeds and the camera frame rate is 30 f.p.s. So we expect the product $\omega \cdot \Delta T \approx 0$ to be small between two consecutive frames. However if the camera motion is much faster or the camera frame rate is decreased, this approximation for computing the Jacobian may be not valid. The Levenberg-Marquardt algorithm [21] has been used for all non-linear optimizations and a 99% confidence value is used for RANSAC procedure.

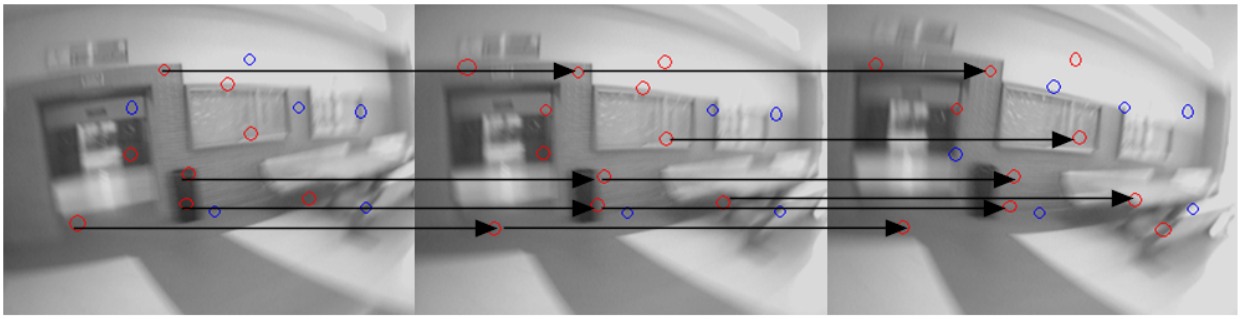
The Harris corner detector [22] is used to detect salient image regions, and a patch of size 11×11 pixels centered on the interest point is used as a feature descriptor. Once the EKF prediction step is done (considering the constant velocity motion model or visual odometry priors), *active search* [2] is performed to find potential matches between the initial 2D template of a feature when it was initialized and the new feature's appearance according to the current camera viewpoint.

We show experimental results for two indoor sequences. Two well known problems in EKF-SLAM are that the processing time associated with the EKF update is $O(n^2)$, where n is the number of landmarks in the map, and divergence over time due to non-linearity problems. All experiments in this paper have been performed in small scenarios in order to disengage intrinsic EKF problems of results due to motion model. The first one is a sequence where the camera moves straight into a corridor and several fast camera rotations are performed. The purpose of this sequence is to show how the constant velocity model fails tracking this fast rotation yielding a completely wrong camera pose, and how visual odometry priors can deal with this fast rotation. We compare our visual odometry priors with two different configurations of the constant velocity model. In the *Experiment 1* linear acceleration noise components in P_n were set to a standard deviation of 0.3 m/s^2, and angular components to a standard deviation of 0.8 rad/s^2. In the *Experiment 2* linear and angular accelerations are set to 1.0 m/s^2 and 10.0 rad/s^2 in order to cope with the rapid changes in orientation of the camera.

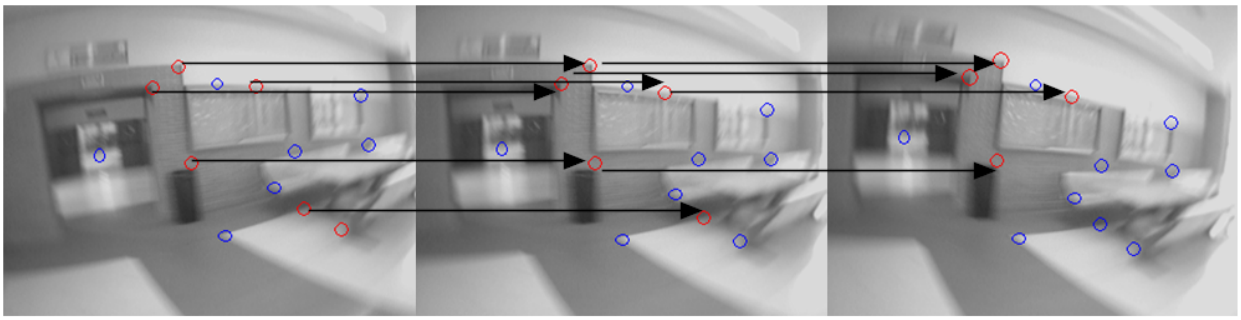
Fig. 1 depicts a comparison between the *Experiment 1* (a) and visual odometry priors (b), where we show *active search* in three consecutive frames where a fast camera rotation is performed. Ellipses in red colour means that the feature has been correctly matched (high 2D templates correlation value) whereas blue colour means that the feature has not been matched correctly. As can be observed, with visual odometry priors *active search* is correctly performed whereas with the constant velocity model the search areas are not in the correct position due to the fast camera rotation yielding bad features estimates corrupting the pose and the map.

Fig. 2 depicts a comparison for the first sequence between the two configurations of the constant velocity model and visual odometry priors with respect to an approximated ground truth, obtained with a batch bundle adjustment implementation. We show results in translation and two components of the quaternion (q_X, q_Z) where the rotation was performed. The results considering visual odometry priors are much better than the other two experiments that yield completely wrong pose estimates, both in translation and rotation. As expected, the obtained results in the *Experiment 2* are better than in the *Experiment 1* since the angular standard deviation was higher for the second case.

The second sequence is a small loop where the camera motion is smooth, but 6DoF since the camera is carried in hand. In this case, in the *Experiment 2* linear and angular accelerations are set to 1.0 m/s^2 and 4.0 rad/s^2 since

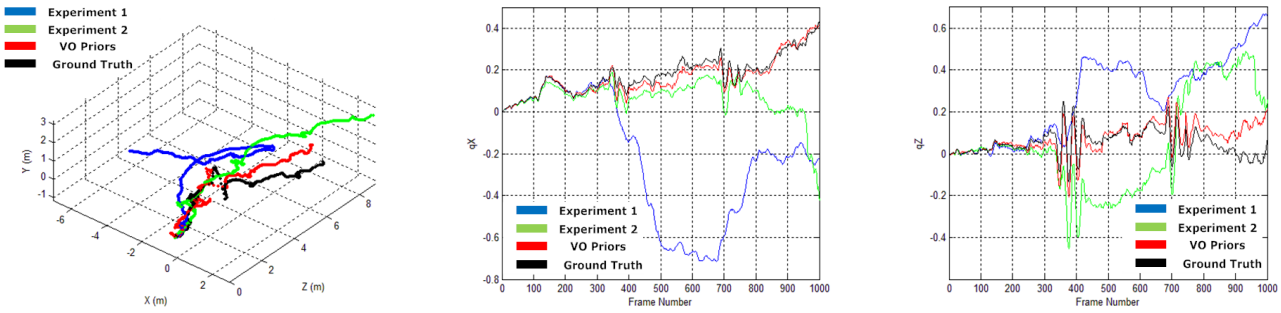


(a) Constant Velocity motion model: Experiment 1



(b) Visual Odometry priors

Fig. 1. Sequence 1: Fast camera rotation in three consecutive frames



(a) Translation (b) q_X (c) q_Z

Fig. 2. Sequence 1: Translation and Orientation Results

Case	$\epsilon_x(m)$	$\epsilon_y(m)$	$\epsilon_z(m)$	ϵ_{q_0}	ϵ_{q_X}	ϵ_{q_Y}	ϵ_{q_Z}
Experiment 1	2.5446	0.3374	1.8508	0.3051	0.3168	0.3525	0.0846
Experiment 2	0.9189	0.3479	0.8710	0.1320	0.1112	0.1153	0.0805
VO Priors	0.8184	0.1288	0.2679	0.0573	0.1037	0.0669	0.0431

TABLE I

SEQUENCE 2: LOCALIZATION ERRORS WITH RESPECT TO GROUND TRUTH

camera rotation is smaller in this sequence. Fig. 3 depicts a comparison of the camera translation results for this sequence. The constant velocity model experiments are not able to close the loop, whereas the case considering visual odometry priors is able to close the loop. Table I shows the mean squared error with respect to the ground truth for both translation and rotation. We performed a timing evaluation which reveals that the performance can function in real-time. On 320×240 frames, feature extraction takes around 4 ms with subpixel precision, extracting the pyramidal optical flow takes around 1 ms and the two-stage visual odometry takes around 4 ms. All timing results were obtained on a Core 2 Duo 2.2GHz laptop computer.

V. CONCLUSIONS

In this paper we have shown how visual odometry priors can be used in conjunction with EKF-SLAM,

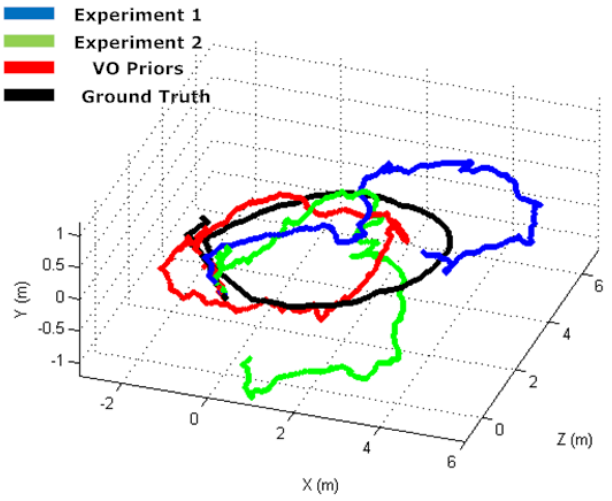


Fig. 3. Sequence 2: Translation Results

improving considerably the accuracy in localization and mapping with respect to using a standard motion model while continuing to meet real-time demands. Although we have presented results for the stereo vision case, the basis of our algorithm can be used for monocular vision, for which at least a good prior in the camera rotation can be obtained assuming that the camera is purely rotating between frames. The main advantage of our method it is that it can handle any kind of camera motion providing good pose priors, whereas for the constant velocity model case setting the standard velocities deviations (linear and angular) can be problematic and dependent on the camera motion.

As future work we are interested in the extension of our method to large environments and difficult scenarios (such as stairs) and compare to other alternatives. In addition, the use of our method in conjunction with JCBB [23] for data association can yield a very robust visual SLAM approach. A comparison with the second order motion model described in [11] could be of interest.

Acknowledgements

This work was supported in part by the Spanish Ministry of Education and Science (MEC) under grant TRA2008-03600 (DRIVER-ALERT Project) and by the Community of Madrid under grant CM: S-0505/DPI/000176 (RoboCity2030 Project).

References

[1] M. Kaess, K. Ni, and F. Dellaert, "Flow separation for fast and robust stereo odometry," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2009.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, 2007.

[3] T. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 4, pp. 639–656, Jul 1990.

[4] T. Broida and R. Chellappa, "Estimating the kinematics and structure of a rigid object from a sequence of monocular images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 6, pp. 497–513, Jun 1991.

[5] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robotics*, 2008.

[6] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2009, pp. 375–380.

[7] B. Williams, G. Klein, and I. Reid, "Real-time SLAM relocalisation," in *Intl. Conf. on Computer Vision (ICCV)*, 2007.

[8] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large scale 6DOF SLAM with stereo-in-hand," *IEEE Trans. Robotics*, vol. 24, no. 5, 2008.

[9] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and E. López, "Real-time hierarchical outdoor SLAM based on stereovision and GPS fusion," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 440–452, Sep. 2009.

[10] B. Williams, P. Smith, and I. Reid, "Automatic relocalisation for a single camera simultaneous localisation and mapping system," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2007, pp. 2784–2790.

[11] P. Gemeiner, A. Davison, and M. Vincze, "Improving localization robustness in monocular SLAM using a high-speed camera," in *Robotics: Science and Systems (RSS)*, 2008.

[12] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *Eur. Conf. on Computer Vision (ECCV)*, Marseille, France, 2008.

[13] D. Nistér, O. Naroditsky, and J. Bergen, "Visual Odometry," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[14] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. Sawhney, "Ten-fold improvement in visual odometry using landmark matching," in *Intl. Conf. on Computer Vision (ICCV)*, 2007.

[15] K. Konolige and M. Agrawal, "Frame-frame matching for realtime consistent visual mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Apr 2007, pp. 2803–2810.

[16] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker. Description of the Algorithm," Intel Corporation. Microprocessor Research Labs, Tech. Rep., 2000.

[17] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Machine Intell.*, 2009. [Online]. Available: http://lanl.arXiv.org/pdf/0810.2434

[18] K. Konolige, "Small vision system: Hardware and implementation," *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, pp. 111–116, 1997.

[19] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide baseline stereo," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 99, no. 1, 2009.

[20] R. Bolles and M. Fischler, "A RANSAC-based approach to model fitting and its application to finding cylinders in range data," in *Intl. Joint Conf. on AI (IJCAI)*, Vancouver, Canada, 1981, pp. 637–643.

[21] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, 2004.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[23] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.