Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Robust Camera Egomotion Estimation from 3D Straight Line-Based Environment model

Fakhreddine Ababsa, *Member, IEEE*
IBISC Laboratory, CNRS FRE 3190
University of Evry-Val-d'Essonne
40, rue du Pelvoux, 91020 Evry, France
absbsa@iup.univ-evry.fr

***Abstract*— In this paper we present a new robust camera pose estimation approach based on 3D lines features. The proposed method is well adapted for mobile augmented reality applications We used an Extended Kalman Filter (EKF) to incrementally update the camera pose in real-time. The principal contributions of our method include first, the expansion of the RANSAC scheme in order to achieve a robust matching algorithm that associates 2D edges from the image with the 3D line segments from the input model. And second, a new powerful framework for camera pose estimation using only 2D-3D straight-lines within an EKF. Experimental results on real image sequences are presented to evaluate the performances and the feasibility of the proposed approach in indoor and outdoor environments**

***Keywords*—augmented reality, markerless tracking, extended kalman filter, Ransac.**

I. INTRODUCTION

Camera ego motion estimation is one of most challenging problem in computer vision. Several approaches based on natural features (corner points, planes, edges, silhouettes, etc.) in the scene have been developed last years. The mean idea of these techniques is to find correspondences between 2D features extracted from the image and 3D features defined in the world frame. The problem is then solved using 2D-3D registration techniques. Numerical nonlinear optimization methods like the Newton-Raphson or Levenberg-Marquardt algorithm are generally used for the minimization. Wuest et al. [1] present a model-based line tracking approach that can handle partial occlusion and illumination changes. The camera pose is computed by minimizing the distances between the projection of the model lines and the most likely matches found in the image. Drummond and Cipolla [2] propose a novel framework for 3D model-based tracking. Objects are tracked by comparing projected model edges to edges detected in the current image. Their tracking system predicts the edge locations in order to rapidly perform the edge search. They have used a Lie group formalism in order to transform the motion problem into simple geometrics terms. Thus, tracking becomes a simple optimization problem solved by means of iterative reweighed least squares. Yoon et al. [21] present a model-based object tracking to compute the camera 3D pose. Their algorithm uses an Extended Kalman Filter (EKF) to provide an incremental pose-update scheme in a prediction-verification framework. In order to enhance the accuracy and the robustness of the tracking against occlusion, they take into account the measurement uncertainties associated with the location of the extracted image straight-lines. Recently, Comport et al. [4] propose a real-time 3D model-based tracking algorithm. They have used a visual servoing approach to formulate the pose estimation problem. A local moving edges tracker based on tracking of points normal to the object contours is implemented. In order to make their algorithm robust, they have integrated a M-estimator into the visual control law. Other approaches have also been applied where different features have been combined to compute the camera pose. Ababsa and Mallem [5] propose to combine point and line features in order to handle partial occlusion. They integrated a M-estimator into the optimization process to increase the robustness against outliers. Koch and Teller [6] describe an egomotion estimation algorithm that takes as input a coarse 3D model of an environment . Their system uses a prior visibility analysis to speed initialization and accelerate image/model matching. Other approaches use Simultaneous Localization And Mapping (SLAM) to track the camera pose while building a 3D map of the unknown scene [7][8]. The main problem with most existing monocular SLAM techniques is a lack of robustness when rapid camera motions, occlusion and motion blur occur.

In this paper we present an original robust camera pose tracking using only straight lines and which differs from existing work. We propose to combine an EKF with a RANSAC scheme in order to achieve a robust 2D-3D lines matching. This gives an efficient solution for outliers rejection. To our knowledge such solution has not been explored before. Furthermore, we have combined the 2D-3D lines correspondence constraints for object pose estimation, developed by Phong et al. [9], with an EKF in order to update recursively the camera pose. We have compared our results with classical approaches where pose estimation is solved using least square approaches [10][11][12]. Our method requires no training phase, no artificial landmarks, and uses only one camera.

The rest of the paper is structured as fellows: In section II, we describe the camera pose estimation problem formulation when using straight lines, and we also give a complete implementation of the Extended Kalman Filter to update the camera pose recursively over the time using 2D and 3D lines features. In section III, we explain how we have expended the RANSAC scheme [13] in order to achieve robust 2D-3D lines matching. In section IV, we show experimental results and evaluations, we discuss also the merits and the limitations of

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

the proposed approach. Conclusion and future work are presented in section V.

II. Camera Pose Estimation Algorith

In any Kalman Filter implementation, the system state is stored as a vector. In our algorithm the state is represented by the position and the orientation of the camera with respect to the world coordinate system. For computational we use a unit quaternion to represent the rotation. Thus, the state vector is given by:

$$X = \lfloor q_0 \quad q_x \quad q_y \quad q_z \quad t_x \quad t_y \quad t_z \rfloor \tag{1}$$

where $\left(q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1\right)$.

We denote the camera state at time t by the vector X_t. The EKF is used to maintain an estimate of the camera state X in the form of a probability distribution $P\left(X_t \middle| X_{t-1}, Z_t\right)$, where Zt is the measurement vector at time t. The Kalman filter models the probability distribution as Gaussian, allowing it to be represented by a covariance matrix Σ. In an extended Kalman filter, the non linear measurements and motion models are linearised about the current state estimate as Jacobian matrices.

Our algorithm follows the usual predict-refine cycle, whereby the state X is predicted at timestep t, and measurements are used to refine the prediction.

A. Time update

The time update model is employed in order to predict the camera pose at the following time step. In our case, the time update is simple because of the fact that we estimate the camera pose at each frame of the images sequence. Therefore the 3D camera pose between two successive frames changes very little. The time update equation is then given by :

$$X_t^- = A \cdot X_{t-1} \tag{2}$$

Where A is 7×7 identity matrix.

The time update step also produces estimates of the error covariance matrix Σ from the previous time step to the current time step t. To perform this prediction we use the general update equation of the Kalman filter:

$$\Sigma_t^- = A \cdot \Sigma_t \cdot A' + Q_{t-1} \tag{3}$$

Where Qt represents the covariance matrix of the process noise. Σ reflects the variance of the state distribution.

B. Measurement model and estimate update

The measurement update model relates the state vector to the measurement vector. Since our goal is to estimate the camera pose using only straight lines, we will first describe the constraint equation which relates the state vector to the 3D model lines and their corresponding 2D image edges. We choose to base our technique on line features, rather than points, because this approach is relatively unexplored in the vision literature. We consider a pin-hole camera model and we assume that the intrinsic camera parameters are known. The world coordinate frame is a reference frame. All the 3D model lines are defined with respect to it. Let Li be a 3D line. Li is represented with the Cartesian coordinates of its two end-points P_1^i and P_2^i (see figure 1). The points P_1^i and P_2^i in world coordinates can be expressed in the camera frame as well :

$$\begin{cases} P_{1/C}^i = R \cdot P_{1/W}^i + T \\ P_{2/C}^i = R \cdot P_{2/W}^i + T \end{cases} \tag{4}$$

Where the 3×3 rotation matrix R and the translation vector T describe the rigid body transformation from the world coordinate system to the camera coordinate system and are precisely the components of the camera state vector.

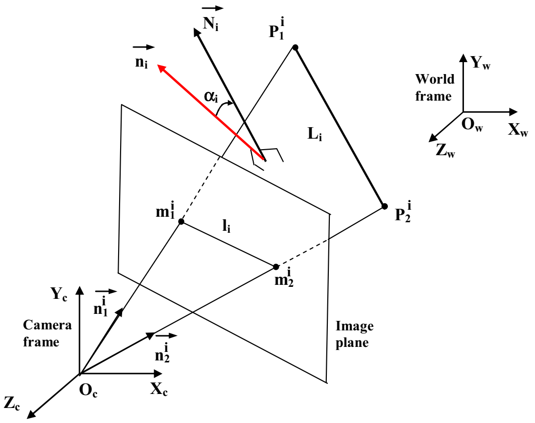


Figure 1. Projection plane. The model line, its projection onto the image and the center of projection OC are coplanar.

We can see that the points $P_{1/C}^i$, $P_{2/C}^i$ and the center of projection OC are coplanar. $\vec{N}_i$ is the unit vector normal to this plane. $\vec{N}_i$ can be expressed in the camera coordinates frame as follows :

$$\vec{N}_i = \frac{\overrightarrow{O_C P_{1/C}^i} \times \overrightarrow{O_C P_{1/C}^i}}{\left\| \overrightarrow{O_C P_{1/C}^i} \times \overrightarrow{O_C P_{1/C}^i} \right\|} \tag{5}$$

Furthermore, a measurement input of the normal vector $\vec{N}_i$ can be obtained from the image data. Indeed, image line matched with model line belongs also to the projection plane defined above. Let li be a 2D image line corresponding to the 3D line L_i. In similar manner l_i is represented by its two extremities $m_1^i = \left[u_1^i \quad v_2^i\right]^T$ and $m_2^i = \left[u_2^i \quad v_2^i\right]^T$ defined in the 2D image coordinates frame. The points m_1^i and m_2^i can be expressed in the camera coordinate frame as follows:

$$\begin{cases} m_{1/C}^{i} = K^{-1} \cdot \begin{bmatrix} u_1^i & v_1^i & 1 \end{bmatrix}^T \\ m_{2/C}^{i} = K^{-1} \cdot \begin{bmatrix} u_2^i & v_2^i & 1 \end{bmatrix}^T \end{cases} \tag{6}$$

Where the matrix K contains camera calibration parameters, such as focal length, aspect ration and principal point coordinates.

A measurement $\vec{n}_i$ of the unit vector $\vec{N}_i$ normal to the projection plane is thus given by (see figure 1):

$$\vec{n}_i = \vec{n}_1^i \times \vec{n}_2^i \tag{7}$$

Where

$$\vec{n}_1^i = \frac{\overrightarrow{O_C m_{1/C}^i}}{\left\| \overrightarrow{O_C m_{1/C}^i} \right\|} \quad et \quad \vec{n}_2^i = \frac{\overrightarrow{O_C m_{2/C}^i}}{\left\| \overrightarrow{O_C m_{2/C}^i} \right\|}$$

Combining equations (5) and (7), a measurement equation can be written, for each matching event $L_i \rightarrow l_i$:

$$z_t = h(X_t) + v_t \tag{8}$$

Where

$$\begin{cases} z_t = \vec{n}_i = \begin{bmatrix} n_x^i & n_y^i & n_z^i \end{bmatrix}^T \\ h(X_t) = \vec{N}_i = \begin{bmatrix} N_x^i & N_y^i & N_z^i \end{bmatrix}^T \end{cases} \tag{9}$$

v_t represent the noise term in the measurement input with covariance R_t. The noise is due to the uncertainty in the measured image position of the end points of the extracted 2D lines. The non linear function $h(X)$ in measurement equation (8) relates the state to the measurement input. Three 2D-3D line correspondences are sufficient in theory to recover 6-DOF camera pose [14] through in practice mores line may be required to increase accuracy.

The state estimate and covariance are refined after each feature measurement z_t using the standard equation of the EKF as follows:

$$\begin{aligned} K_t &= \Sigma_t^- \cdot H_t^T \cdot \left(H_t \cdot \Sigma_t^- \cdot H_t^T + R_t\right)^{-1} \\ X_t &= X_t^- + K_t \cdot \left(z_t - h\left(X_t^-\right)\right) \\ \Sigma_t &= \Sigma_t^- - K_t \cdot H_t \cdot \Sigma_t^- \end{aligned} \tag{10}$$

Where H_t is the Jacobian matrix defined by:

$$H_t = \left. \frac{\partial h(X)}{\partial X} \right|_{X = X_t^-} \tag{11}$$

The measurement update model is executed once a set of 2D-3D matched lines become available.

C. Iterated EKF

The standard EKF method does not consider errors due to the linearization of the non linear function $h(X)$ in the vicinity of X_t^-. However, theses errors can lead to wrong estimates and/or divergence of the camera pose. Since the nonlinearity is only in measurement equation, the Iterated Extended Kalman Filter (IEKF) is the best technique to deal with it. The IEKF uses the same prediction equation as EKF, namely (2) and (3). The measurement update relations are replaced setting $X_t^0 = X_t^-$ and doing iteration on :

$$\begin{aligned} H_t^k &= \left. \frac{\partial h(X)}{\partial X} \right|_{X = X_t^k} \\ K_t^k &= \Sigma_t^- \cdot H_t^{k^T} \cdot \left(H_t^k \cdot \Sigma_t^- \cdot H_t^{k^T} + R_t\right)^{-1} \\ X_t^{k+1} &= X_t^k + K_t^k \cdot \left(z_t - h\left(X_t^k\right)\right) \end{aligned} \tag{12}$$

For iteration number $k = 0,1,\cdots,N-1$. At the end of all iterations, $X_t = X_t^N$. The covariance matrix is then updated based on X_t^N according to :

$$\Sigma_t = \Sigma_t^- - K_t^N \cdot H_t^N \cdot \Sigma_t^- \tag{13}$$

The iteration could be stopped when consecutive values X_t^k and X_t^{k+1} differ by less than a defined threshold

III. Robust 2D-3D Lines Matching Algorith

In this section we explain the expansion of the RANSAC scheme that we have developed in order to achieve a robust matching algorithm that associates 2D edges from the image with the 3D line segments from the input model, and without using any verification algorithm.

Let $\{l_i\}, i = 1,...,N$ be a set of 2D edges extracted from the image and $\{L_j\}, j = 1,...,M$ a set of 3D model lines. Our robust lines matching algorithm is summarized as follows :

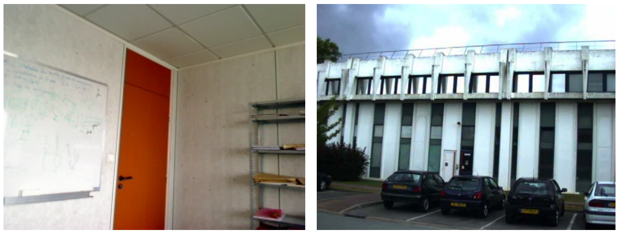
1. Randomly sample subsets of four {li ↔ Lj} pairs of 2D and 3D lines. In theory a minimum three pairs are of lines are sufficient to compute an accurate rigid transformation.
2. For each sample, compute the camera pose Π(R,T) using the IEKF algorithm described in section II.
3. Each candidate Π is tested against all the correspondences $l_i \rightarrow L_j$ by computing, in the camera frame, the angle between the normal vector $\vec{n}_i$ (see figure 1) associated with the image line l_i and the transformed line $R \cdot L_j$. If this match is wrong with respect the pose Π, then the co sinus of the angle should be significantly larger than zero.

4. We choose the pose Π which has the highest number of inliers, i.e the Π for which all the pairs are within a fixed angle threshold.

Hence, the obtained camera pose for the current image is robustly updated using only inliers of correspondences.

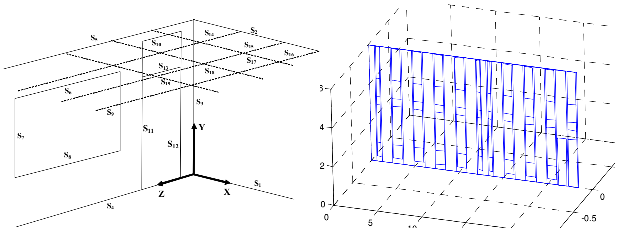
IV. Experimental Results

The proposed camera pose estimation algorithm have been tested in real scenes and the registration accuracy was analyzed. The results are presented for both indoor and outdoor images (Figure 2). The indoor scene consists of a camera moving in an office room whereas the outdoor one corresponds to a moving camera pointing towards one frontage of a building. The frame rate of the recorded image sequences is about 25 frames/s and the resolution of the video images is 320×240 pixels. The 3D models of the office room and the building frontage are known, they are composed, respectively, of 19 lines and 120 line defined by the 3D coordinates of their end points within the world coordinates frame (Figure 3).



(a) (b)

Figure. 2. Two frames from the recorded images sequence. (a) indoor scene. (b) outdoor scene.



(a) (b)

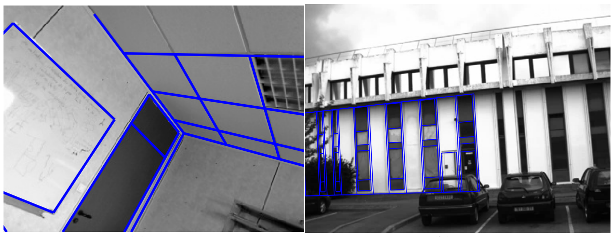
Figure. 3. 3D models of indoor and outdoor scenes used for experiments. (a) the office room model. (b) The frontage building model.

In order to estimate the camera pose accuracy we defined, in the camera space, the registration error ξ. Given a set of correspondences between image edges and model segments the error ξ corresponds to the normalized square sum of the sinus of the angular disparities α_i for each correspondence between image edge and the re projected model segments (Figure 1):

$$\xi = \frac{1}{M}\sum_{i=1}^{M}\sin(\alpha_i)^2 = \frac{1}{M}\sum_{i=1}^{M}\left\|\vec{n}_i \times \vec{N}_i\right\|^2 \qquad (14)$$

Where M is the number of correspondences and α_i the angle between the two planes spanned by the camera center, the observed image edge l_i and the model segment L_i (see figure 1).

In our experiment we took M=10 correspondences. We have first considered that the data set has no outliers (100 % inliers or good matching) and we computed the registration error for several frames of the image sequence. Our algorithm was run on both the indoor and outdoor scenes. Similar results were obtained in both situations, Thus, for the indoor scenario, the mean error is about $\xi_m = 4.01\times10^{-5}$ which corresponds to the mean angular disparities $\alpha_m = 0.28°$. For the outdoor scene, we obtained $\xi_m = 3.43\times10^{-5}$ and $\alpha_m = 0.28°$. Figure (4) shows the projection of the office model using the camera pose estimated by our algorithm, and as can see it is quite skewed. All the lines are fairly well aligned.



(a) (b)

Figure 4.. Projection of the 3D models using the final camera pose estimate when the input data has no outliers. (a) indoor scene (b) outdoor scene

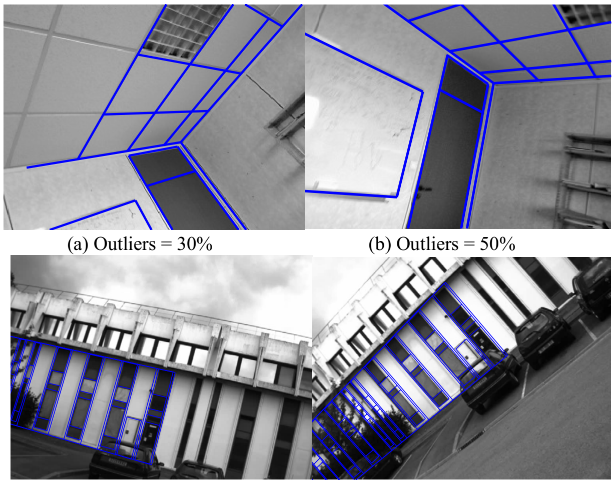
In the second experiment, we have evaluated the capacity of our robust algorithm to reject outliers in observed data. In our case, an outlier corresponds to a wrong feature matching between a 2D and a 3D line. For that, we have contaminated the input data set with different percentage of outliers, for both indoor and outdoor scenes, and have computed the corresponding registration error. The obtained results are summarized in table 1.

TABLE I

Experimental Results of the Robust Algorithm

Indoor scene			
Outliers (%)	ξ_m	α_m(°)	Number of trials
30%	4.93×10^{-5}	0.37	4
40%	6.95×10^{-5}	0.39	12
50%	7.84×10^{-5}	0.42	50
60%	8.18×10^{-5}	0.44	240
Outdoor scene			
Outliers (%)	ξ_m	α_m(°)	Number of trials
25 %	3.81×10^{-5}	0.34	4
37 %	5.57×10^{-5}	0.37	6
50 %	8.64×10^{-5}	0.44	56

As can be seen, our robust algorithm succeeded in all the cases to detect and delete the outliers. The camera pose is then estimated using the final data consensus which contains only the good 2D-3D lines correspondences. For example, in the worst case when 60% of the input data for the indoor scene (i.e. 6 lines correspondences among 10) are labeled as outliers, our algorithm was been able to identify the four inliers in the data. The camera pose returned using this inliers gives a registration error about 8.18×10^{-5}. This result demonstrates the robustness and the accuracy of the proposed approach. Furthermore, we note that the number of trials needed to get the best solution increase with number of outliers (for example 240 trials for 60% of outliers for the indoor scenario and 56 trails for 50% of outliers for the outdoor scene). This means more processing time and will decrease the real time performance of the algorithm. 40% of outliers (15 trials on average) represents a good compromise. Figure 5 shows the projection of the models using the pose estimated by our algorithm for different frames of the indoor and outdoor images sequence and when run with different percentage of outliers. We can see that all lines are well aligned, and in this cases, the outliers have not affected the registration performance of our robust algorithm.


(a) Outliers = 30%

(b) Outliers = 50%

(c) Outliers = 37%

(d) outliers = 50%

Figure. 5. Camera pose estimation results

Another advantage of our approach is its robustness to severe lines occlusion. Indeed, as the line constraint equation (see section II-B) for the camera pose parameters was developed in the case of "infinite image line". Any image points on the 2D line can be used to construct the corresponding projection plane. So, when partial occlusion occurs, it is enough to detect only small parts of the image edges to estimate the camera pose. In figure 6 we can see that several image edges are partially occluded (table and door for the indoor scene and because of the trees and the cars for the outdoor scene), in spite of that, the camera pose was successfully estimated.

We analyzed the processing time needed for camera pose estimation on a Pentium IV with 3GHz. All computations were performed in Matlab. The pose estimation process using IEKF does not take much time. Indeed, we have tuned the parameters of the IEKF in such manner so that it converges in few iterations (20 at the maximum). The processing time strongly depends only on the number of outliers in the current field of view. For example, the average time is about 28 millisecond per frame when having 40% of outliers in 10 input data for indoor scene. 3 milliseconds are used to estimate the camera pose with the IEKF and 25 milliseconds are measured for the time needed to reject outliers.

We have also evaluated the 3D camera localization accuracy. For that, we considered the outdoor scene, and we compared the computed camera trajectory obtained with our algorithm with the ground truth data. The ground truth for position are provided by a Trimble Pathfinder ProXT GPS receiver giving a submeter accuracy, The covered distance is about 30 meters. The ground truth for rotation was obtained using an Xsens MTx gyro mounted rigidly with the camera [32]. The results of the experiments are given in figure 6 which shows the error between the ground truth data and the recovered position and orientation of the camera for different frames of the image sequence. Table 2 reports the mean values and the standard deviation of the position and the orientation errors. It can be seen that our algorithm allows a good 3D localization performance, especially for orientation components.

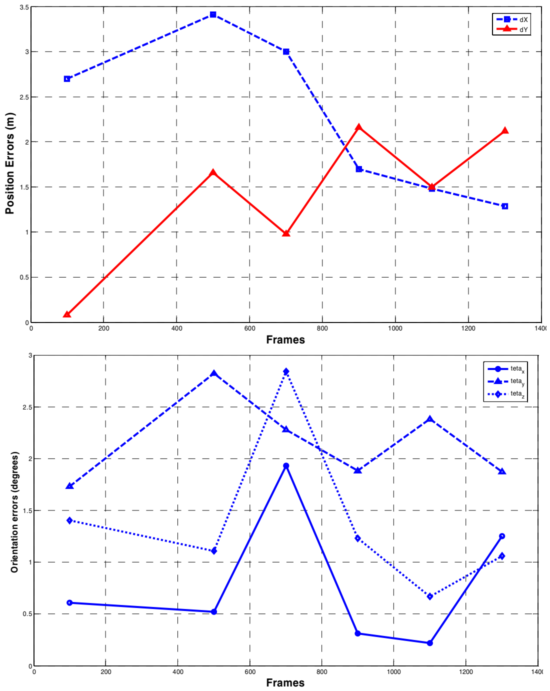


Figure. 6. Position and orientation accuracy with respect to ground truth

TABLE 2
3D LOCALIZATION PERFORMANCE

	Position errors (m)		Rotation errors (deg)		
	ΔX	ΔY	θ_x	θ_y	θ_z
Mean	2.26	1.42	0.81	2.61	1.38
Std	0.88	0.77	0.66	0.41	0.75

V. Conclusion

In this paper, we proposed a new approach for 6-DOF camera localization based on matching between 2D image edges and 3D model segments. We performed a generic camera pose estimation framework based only on lines features using an Iterated Extended Kalman Filter. We also achieved significant improvements on robust 2D/3D lines matching scheme by adapting the well-know RANSAC algorithm to our application. The experimental results confirm the robustness of our approach against severe occlusion and outliers for both indoor and outdoor applications. We also showed the good performance of our algorithm to localize a moving camera in 3D environment. Future work will be devoted to extend our approach by using other outdoor sensors (e.g. an inertial sensor and a GPS). Thus, the system could be used for navigation and localization in large-scale outdoor environments. An hybrid algorithm will the fuse the data provided by the three sensors in order to refine the camera pose.

References

[1] H. Wuest, F. Vial, and D. Stricker, "Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality". In Proceedings of ACM/IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR 2005), Vienna, Austria, 2005, pp. 62-69.

[2] T. Drummond and R. Cipolla, "Real-Time Visual Tracking of Complex Structures," *IEEE Trans. Patt. Anal. and Mach. Intell*, vol. 24, no. 7, pp. 932-946, July 2002.

[3] Y. Yoon, A. Kosaka, J. B. Park and A. C. Kak, "A New Approach to the Use of Edge Extremities for Model-based Object Tracking". In Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA'05). Barcelonna, Spain, 2005, pp. 1883-1889.

[4] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework". *IEEE Trans. on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 615-628, July/August 2006.

[5] F. Ababsa and M. Mallem, "Robust camera pose estimation combining 2D/3D points and lines tracking". In Proceedings of the 2005 IEEE International Symposium on Industrial Electronics (ISIE'08). Cambridge, UK, 2008, pp. 774-779.

[6] O. Koch and S. Teller, "Wide-Area Egomotion Estimation from Known 3D Structure". In Proceedings of the 2007 IEEE International Conference on Vision and Pattern Recognition (CVPR'07), Minneapolis, USA, 2007, pp.1-8.

[7] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM", *IEEE Trans. Patt. Anal. and Mach. Intell*., vol. 2, no. 6, pp. 1052-1067, June 2007.

[8] B. Williams, P. Smith and I. Reid,"Automatic Relocalisation for a Single Camera Simultaneous Localisation and Mapping System". In Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA'07). Roma, Italy, 2007, pp. 1883-1889.

[9] T. Q. Phong, R. Horaud, A. Yassine and P. D. Tao,"Object Pose from 2D to 3D Point and Line Correspondences". *International Journal of Computer Vision*, vol. 15, pp. 225-243, 1995.

[10] D.G. Lowe. "Fitting Parameterized Three-Dimensional Models to Images". *IEEE Trans. Patt. Anal. and Mach. Intell*, vol. 13, pp. 441-450, 1991.

[11] R.M. Haralick, "Pose Estimation From Corresponding Point Data". *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1426-1446, 1989.

[12] C. P. Lu, G. Hager, and E. Mjolsness. "Fast and globally convergent pose estimation from video images". *IEEE Trans. Patt. Anal. and Mach. Intell*, vol. 22, no 6, pp. 610–622, June 2000.

[13] M. A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381- 395, June 1981

[14] M. Dhome, M. Richetin, and J-T. Lapreste, "Determination of the attitude of 3D objects from a single perspective view". *IEEE Trans. Patt. Anal. and Mach. Intell*, vol. 11, no. 12, pp. 1265-1278, 1989.