Visual planes-based Simultaneous Localization And Model Refinement for augmented reality

Fabien Servant Eric Marchand Pascal Houlier Isabelle Marchal
Orange Labs INRIA Orange Labs Orange Labs
INRIA

Abstract

This paper presents a method for camera pose tracking that uses a partial knowledge about the scene. The method is based on monocular vision Simultaneous Localization And Mapping (SLAM). With respect to classical SLAM implementations, this approach uses previously known information about the environment (rough map of the walls) and profits from the various available databases and blueprints to constraint the problem. This method considers that the tracked image patches belong to known planes (with some uncertainty in their localization) and that SLAM map can be represented by associations of cameras and planes. In this paper, we propose an adapted SLAM implementation and detail the considered models. We show that this method gives good results for a real sequence with complex motion for augmented reality (AR) application.

1 Introduction

The goal of AR is to insert virtual information in the real world providing the end-user with additional knowledge about the scene. The added information, usually virtual objects, must be precisely aligned with the real world. It is then necessary to accurately align real and virtual world and then to compute the full position of the device for each image of the sequence.

Our ultimate goal is to consider augmented reality on mobile devices such as UMPC using their own integrated camera and, if available, inertial sensors. Considering such systems raises many issues. First, the visual sensor (monocular cameras) does not provide any depth information. Furthermore, due to bad quality of the lenses, even the estimated bearing information is far from being perfect. Uncertainty related to visual measurements have thus to be handled. Finally, since the system is moved by hand, it is not possible to make any assumptions about its motion that could make the localization estimation easier. On the other hand, the specific application also provides us some clues to simplify the problem. We have prior information about the vertical surfaces positions (mainly the walls) in the environment, provided by a C.A.D. model (let us note that we do not use a complete scene model as in [6, 4]) or any other database. The first pose at initialization is given by an external method [12].

The presented solution used to solve this problem of "*planes-based*" motion computation considers the monocular Simultaneous Localization and Mapping (SLAM) approach proposed in [5] in order to introduce a constraint on the vertical planes location. By restricting the estimation issue to perform measures on patches belonging to planes registered in the database, and by modifying the models, we show that the use of vertical surfaces allows to reduce complexity and improves the estimation of the pose computed by the Extended Kalman Filter (EKF).

2 S.L.A.M.R.

Simultaneous Localization and Model Refinement (SLAMR) is a fork of EKF-SLAM[1]. Almost all theory of EKF-SLAM applies but, in our case, the map is initially partially known. EKF theory is described in [2] and will not be recalled in this paper. Moreover, our method can be used simultaneously with a standard SLAM, for example when navigating in a partially modeled building.

In this paper, the EKF state vector is represented by

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_c} & \mathbf{x_{f_1}} & \mathbf{x_{f_2}} & \dots & \mathbf{x_{f_n}} \end{bmatrix}^T \quad (1)$$

where $\mathbf{x_c}$ contains the current pose of the camera along with other parameters such as its velocity and acceleration (See section 2.1). $\mathbf{x_{f_i}}$ is an element of the map and can be either a plane defined in the world frame or a reference camera pose (See section 2.4)

2.1 Camera model and prediction equations

The camera pose is represented by the 6 parameters $\left[\begin{array}{cc} \mathbf{t} & \theta\mathbf{u} \end{array}\right]^{\top}$ where $\mathbf{t}$ (resp. $\theta\mathbf{u}$) is the position (resp. the orientation) of $\mathcal{R}_w$ (world frame) in $\mathcal{R}_c$ (camera frame). We chose for rotation the $\theta\mathbf{u}$ representation [10] (where θ is the angle and $\mathbf{u}$ the axis of the rotation) which does not need any additional constraints (eg, normalization).

Considering the prediction step of the EKF, a motion model has to be defined. This model is important in our application where the camera is handheld as we do not have any odometry. Prediction of the camera pose will be used to predict the image features and propagating the uncertainty through time. Hence we need to find a model that can handle a wide range of motions, while avoiding filter instability.

A constant translational acceleration is then considered, constant angular velocity model as proposed in [9]. We consider that the camera will move according to this model but unknown jerk for translation and acceleration for rotation will happen and be modeled as an additive Gaussian zero-centered noise :

$$\mathbf{x}_c \quad = \quad \left[\begin{array}{ccccc} \mathbf{t} & \theta\mathbf{u} & \boldsymbol{v} & \boldsymbol{\omega} & \mathbf{a} \end{array}\right]^T \tag{2}$$

where $\boldsymbol{v}$ is the translational velocity, $\boldsymbol{\omega}$ is the angular velocity and $\mathbf{a}$ is the translational acceleration. All of them are given in the camera frame $\mathcal{R}_c$. Note there is a difference in order between translation and rotation. This is a consequence of the derivation calculus [13, 10].

Let us define f the non linear prediction model which will predict the new state given the motion parameters:

$$f(\mathbf{x}_c) = \left(\begin{array}{c} \mathbf{R}(-\boldsymbol{\omega}dt)\mathbf{t} - \mathbf{S}(-\boldsymbol{\omega}dt)\boldsymbol{v}dt - \mathbf{T}(-\boldsymbol{\omega}dt)\mathbf{a}\frac{dt^2}{2} \\ \phi(\mathbf{R}(\boldsymbol{\omega}dt)^T.\mathbf{R}(\mathbf{r})) \\ \boldsymbol{v} + \mathbf{a}dt \\ \boldsymbol{\omega} \\ \mathbf{a} \end{array} \right) \tag{3}$$

where $\phi(\mathbf{R})$ is a rotation vector corresponding to a rotation matrix $\mathbf{R}$. $\mathbf{R}(\theta\mathbf{u})$ is the Rodrigues formula which computes a rotation matrix given a rotation vector $\theta\mathbf{u}$. $\mathbf{S}(\theta\mathbf{u})$,$\mathbf{T}(\theta\mathbf{u})$ are integrations of $\mathbf{R}(\theta\mathbf{u})$.

2.2 Introducing planar surfaces in SLAM process

In building blueprints, walls are coded as $2^{1/2}$ dimensions shapes. The model provides the footprints and elevation of the wall with a quite good precision. Consequently, the 3D information is only an extrusion of the footprint and there is no information about *verticality* of the wall nor about its straightness. However, we will assume that walls are perfect and thus can be fully described in 3 dimensions up to a certain uncertainty. Indeed, walls imperfections and small objects fixed to them are considered as noise by our method.

A plane is represented by a normal vector $\mathbf{n}$ and a scalar d which codes for the orthogonal distance to the origin so that $\mathbf{nX} + d = 0$ where $\mathbf{X}$ is a point of the plane.

Both $\mathbf{n}$ and d are needed to compute the measurement model. However, we choose to represent the angle defined by $\mathbf{n}$ by its corresponding spherical coordinates θ and ϕ which doesn't need to be normalized after the update step of the EKF and make its initial uncertainty determination easier. A function $n(\mathbf{s})$ is used to transform a spherical coordinates vector $\mathbf{s}$ in a normal vector: $\mathbf{N} = \left[\begin{array}{ccc} \phi & \theta & d \end{array}\right]$. The planes dimensions are not estimated but used as constants to reduce the image search space when looking for planes in the sequence.

2.3 Homography

Let us define $\mathbf{X}$ as a point on the plane $\mathbf{N}$ and w() a function which normalizes a vector in $\mathbb{R}^3$ by dividing it with its third component. There exists a matrix $\mathbf{^bH_a}$ called an homography which transforms a projected point in frame a (the reference camera frame) into its projected coordinates in frame b. This matrix is defined up to a scale factor.

$$^b\mathbf{X} = w(^b\mathbf{H}_a{}^a\mathbf{X}) \tag{4}$$

$$^b\mathbf{H}_a = {}^b\mathbf{R}_a + \frac{^b\mathbf{t}_a}{^ad}n(^a\mathbf{s})^T \tag{5}$$

In our method, we use the ESM [3] homography tracker which gives the homography (in pixels) $^b\mathbf{G}_a$ for a specified patch between a reference image and the current one. Using the camera internal parameters matrix $\mathbf{K}$, the homography in meters is retrieved with

$$^b\mathbf{H}_a \quad = \quad \mathbf{K}^{-1b}\mathbf{G}_a\mathbf{K} \tag{6}$$

This method is inspired by differentials trackers ([11],[7]) and takes advantage of an efficient second order optimization method. This tracking approach has proved to be very efficient in various context such as robotics or augmented reality. The tracker was modified using robust estimation method (M-Estimators) and is able to handle occlusion (Up to 60 percent of the patch in some scenes).

2.4 Adding Features

For this paper, a template recognition method (based on [14]) is used to detect patches of interests. This will soon be replaced by a modified version of the described tracker which is able to estimate only the plane parameters, the others given by the current SLAMR state.

Once we estimate that a given image zone represents one of the planes, we can use this information in our SLAM method. A new homography tracker is instantiated with the current image as the reference one and a series of i points which define the reference image zone to be tracked. EKF needs an estimate of the measurement to compare and update its state. Since we already have an estimate of the current camera pose, we need to add both the reference camera pose and the plane associated to the new patches.

The plane is added to the state vector once for all and may be shared by many tracked patches. Of course, information and uncertainty about the plane is not dependent of cameras or anything else. This information is already in world frame coordinates and uncertainty comes from the database only. Hence there is no covariance at initialisation between the new plane and the other parts of the state vector. Let $g(\widehat{\mathbf{x}}_c, \mathbf{l})$ be the function which augments the state vector estimation $\widehat{\mathbf{x}}$ with the vector $\mathbf{l}$.

$$g_{plane}(\widehat{\mathbf{x}}, \mathbf{N}) = \left[\begin{array}{cc} \widehat{\mathbf{x}} & \mathbf{N} \end{array}\right] \tag{7}$$

Contrary to the plane, the information and uncertainty about the reference camera is strictly correlated with the current camera pose. There may be multiple reference cameras for a same plane :

$$g_{refcam}(\widehat{\mathbf{x}}) = \left[\begin{array}{ccc} \widehat{\mathbf{x}} & \mathbf{t} & \theta\mathbf{u} \end{array}\right] \tag{8}$$

The current camera frame will be noted $c2$ while the reference camera frame will be noted $c1$ in further equations. A map element may then be considered as a combination of one plane and one reference camera.

2.5 Observation Model

The measurement vector is given by $\lambda(\mathbf{^{c2}H_{c1}})$ where the $\lambda(\mathbf{M})$ function scales the $\mathbf{M}$ matrix so that its last component equals 1 and stack the matrix columns into a vector of size 8. This measurement can be estimated using the observation model defined as

$$\begin{aligned}
{}^{c1}\mathbf{n} &= R({}^{c1}\theta\mathbf{u}_o)n(\mathbf{s}) && (9)\\
{}^{c1}d &= d - ({}^{c1}\mathbf{n}^{T\,c1}\mathbf{t}_o) && (10)\\
{}^{c2}\mathbf{R}_{c1} &= R({}^{c2}\theta\mathbf{u}_o)R({}^{c1}\theta\mathbf{u}_o)^T && (11)\\
{}^{c2}\mathbf{t}_{c1} &= -{}^{c2}\mathbf{R}_1{}^{c1}\mathbf{t}_o + {}^{c2}\mathbf{t}_o && (12)\\
h_j(\mathbf{x}_{c1}, \mathbf{x}_{c2}, \mathbf{N}) &= \lambda({}^{c2}\mathbf{R}_{c1} + \frac{{}^{c2}\mathbf{t}_{c1}}{{}^{c1}d}{}^{c1}\mathbf{n}^T) && (13)
\end{aligned}$$

The Jacobian $\mathbf{J_h}$ of the measurement model $h(.)$ will help transfer innovation of the measurement to both cameras and the plane.

$$\mathbf{J_h} = \left[\begin{array}{cccccc} \frac{\partial\mathbf{h}}{\partial\mathbf{x_{c2}}} & \mathbf{0} & \frac{\partial\mathbf{h}}{\partial\mathbf{x_{c1}}} & \mathbf{0} & \frac{\partial\mathbf{h}}{\partial\mathbf{N}} & \mathbf{0} \end{array}\right] \tag{14}$$

Note that a chi-square test on the innovation covariance tells us if the measurement is statistically valid given the current state vector (Thus removing outliers). For new patches, the test can be used to check if there is a plane in our database which produce a homography estimation similar to the measurement. If the test is successful, the new patch is associated to the plane and used in our pose estimation (See section 2.4).

2.6 Measurement noise

The uncertainty of the measurement is a key feature of the EKF. Wrong uncertainty will lead to a bad estimation of the innovation covariance and thus will badly update the whole state. Because each element of the computed homography is tightly related to the others, the homography covariance is not easy to estimate and must be built from a simpler geometric structure.

Let's define $\mathbf{p}$ as a vector containing a list of points uniformly distributed in the reference image tracked patches. Let's define $\mathbf{p}'$ as the points $\mathbf{p}$ transformed through the homography $\mathbf{H}$. The uncertainty (defined by the covariance $\mathbf{\Sigma_{p'}}$) of $\mathbf{p}'$ is only coming from the uncertainty of $\mathbf{H}$ (defined by the covariance $\mathbf{\Sigma_h} \in \mathbb{R}^{8,8}$). Let $\mathbf{J}_h$ be $\frac{\partial\mathbf{w(Hp)}}{\partial\mathbf{H}}$, then

$$\mathbf{\Sigma_{p'}} = \mathbf{J}_h\mathbf{\Sigma_h}\mathbf{J}_h^T \tag{15}$$

Reversing the problem gives the solution. If we can have a coarse estimate of $\mathbf{\Sigma_{p'}}$, $\mathbf{\Sigma_h}$ can be estimated ([8]) using

$$\mathbf{\Sigma_h} = (\mathbf{J}_h^T\mathbf{\Sigma_{p'}}^{-1}\mathbf{J}_h)^{-1} \tag{16}$$

In our experimentations, we set $\mathbf{\Sigma_{p'}}$ to be a diagonal matrix (No correlation between x and y coordinates) and use the same covariance for each point.

3 Results

We first tested the proposed method on a 1 meter large box. Posters were placed on 4 sides of the box. The camera turns around the box and goes back to its original location. Let us note that the camera is handheld leading to a shaky movement. Two poses are computed, the former uses the presented method, the latter one using a simple Kalman filter with same prediction/measurement models but without handling of planes parameters uncertainty. Noise (up to 6 degrees) have been added in the angular parameters of some plane. Figure 1 shows the results for various frames. The pink polygons represent the patches tracked. The red frame represent the estimated pose with our method

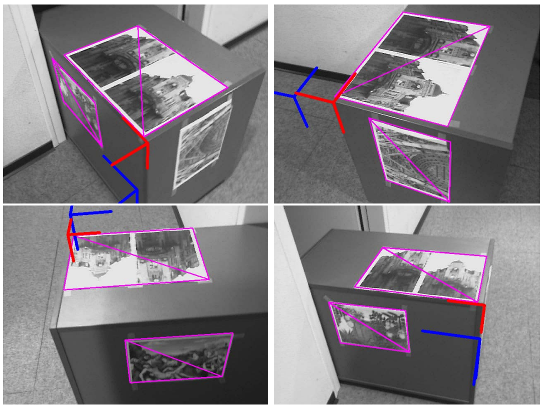

Figure 1. Box result frames

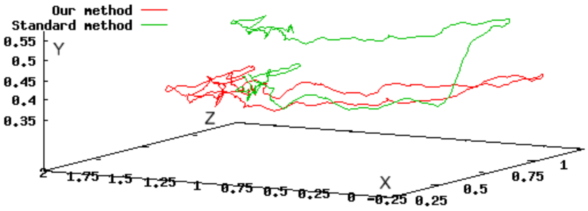

Figure 2. Position of the camera

and the blue with the classical one. Figure 2 plots the pose of the camera for both methods (in meters).

We can see that the proposed method allows a better estimation when there is some errors in the planes models.

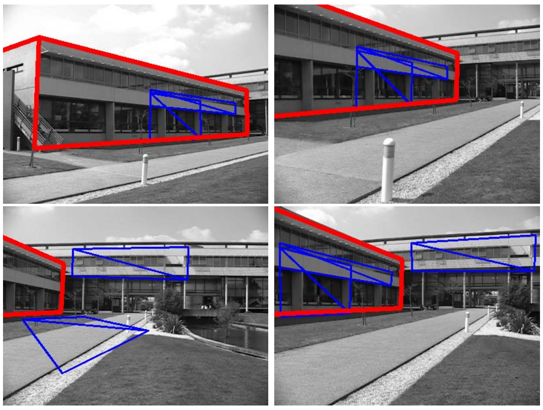

Figure 3. Outdoor result frames

The second scene presented in Figure 3 is an outdoor sequence captured while walking with very coarse knowledge about the walls parameters. Blue polygons are the tracked patches and the red ones are the model of the left wall reprojected using the estimated camera position. This scene shows the robustness of the tracker to non completely planar patches. Moreover, we can see in the fourth image that the patch which was lost previously is found again and reused. This is done by reinitializing at each frame the trackers using the predicted collineation matrices and tracking it again if the patch projection is in the current image.

4 Conclusions

In this paper we have shown that it is possible to integrate a priori structural and absolute information in the SLAM process, just by changing the models and the structure of the map. The approach proved to be efficient on various scene. Our next goal is to automate the selection of patches to track using our database and the statistical tools provided by the described method.

References

[1] T. Bailey, H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Sept. 2006.

[2] Y. Bar-Shalom, T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[3] S. Benhimane, E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ IROS'04*, page 943-948, Sendai, Japan, Oct. 2004.

[4] A. Comport, E. Marchand, M. Pressigout, F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE T. on Visualization and Computer Graphics*, 12(4):615–628, July 2006.

[5] A. Davison, D. Murray. Simultaneous localization and map-building using active vision. *IEEE PAMI*, 24(7):865–880, July 2002.

[6] T. Drummond, R. Cipolla. Real-time visual tracking of complex structures. *IEEE PAMI*, 24(7):932–946, July 2002.

[7] G. Hager, P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE PAMI*, 20(10):1025–1039, Oct. 1998.

[8] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.

[9] D. Koller et al. Real-time vision-based camera tracking for augmented reality applications. In *Int. Symp. on Virtual Reality Software and Technology, VRST'97*, pp. 87–94, Lausanne, Sept. 1997.

[10] Y. Ma et al. *An invitation to 3-D vision*. Springer, 2004.

[11] J. Shi, C. Tomasi. Good features to track. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 593–600, Seattle, Washington, June 1994.

[12] H. Tran, E. Marchand. Real-time keypoints matching: application to visual servoing. In *IEEE ICRA'07*, Roma, Apr. 2007.

[13] Z. Zhang, O. Faugeras. *3D-Dynamic Scene Analysis: A Stereo Based Approach*. Springer-Verlag, 1992.

[14] H. Bay et al. Speeded-Up Robust Features (SURF). In Comput. Vis. Image Underst, 2008.