Detection of the Intersection Lines in Multiplanar Environments: Application to Real-Time Estimation of the Camera-Scene Geometry

Gilles Simon and Marie-Odile Berger
LORIA - Nancy-Université - INRIA Nancy Grand-Est
{*gsimon,berger*}*@loria.fr*

Abstract

This paper describes an integrated system for building a multiplanar model of the scene as the camera is localized on the fly. The core of this system is a robust and accurate procedure for detecting the intersection line between two planes. User cues are used to assist the system in the mapping tasks. Synthetic results and a long video demonstrate the relevance of the method.

1 Introduction

Recent years have seen the emergence of vision-based algorithms for performing localization and mapping in unknown environments [3, 2, 6]. These techniques tend to avoid the need of pre-calibrated environments in real-time applications such as augmented reality. However, they are still very sensitive to data-association errors which can irretrievably corrupt the maps generated by incremental systems.

The method we propose differs in several ways from standard works. Firstly, we consider multiplanar environments (urban or indoors) and aim at building planar surfaces instead of clouds of points. Planar surfaces are natural supports for objects insertion and are easy to track when well textured [7]. Secondly, user cues are used to assist the system in the mapping tasks and visual information are provided to help him make informed decisions about the scene.

The core of our system is an accurate and robust procedure for detecting the projection of the intersection line between two planes based on their apparent motion. Projections of the intersection lines are intermediate results toward the Euclidean reconstruction of the planes. Most of all, their computation can be visually assessed by the user, which is of great interest to prevent map corruptions.

2 Preliminaries

We first set out some theoretical results that will be useful. Suppose we have two images, I_1 and I_2, of a scene consisting of two planes, π_1 and π_2. We further assume that the related homography matrices, $\mathbf{H}_1$ and $\mathbf{H}_2$, from I_1 to I_2 also are known. Using the duality between points and lines in the plane, we deduce from [5] the following result.

Result 1. *The 3×3 matrix $\mathbf{T} = \mathbf{H}_1^T \mathbf{H}_2^{-T}$ is a homology, which admits a pencil of globally fixed lines intersecting at the epipole $\mathbf{e}$ and a distinct fixed line corresponding to the projection $\mathbf{l}$ of the intersection line between π_1 and π_2.*

$\mathbf{l}$ is therefore the eigenvector associated to the simple eigenvalue of $\mathbf{T}$. However, algebraic computations of the eigenvectors of $\mathbf{T}$ is very unstable in practice as $\mathbf{T}$ is a non-symmetric matrix. Particle filtering will therefore be used to detect $\mathbf{l}$ from several subsequent images. The following result will help to distinguish between the two kinds of fixed lines:

Result 2. *Any point on line $\mathbf{l}$ is fixed by $\mathbf{T}^T$ while any point on a line passing through $\mathbf{e}$ is generally transformed by $\mathbf{T}^T$ to a distinct point on the same line.*

The first assertion is straightforward. The second one is illustrated in Fig. 1 ($\mathbf{c}_1$ and $\mathbf{c}_2$ are the camera centers): $\mathbf{p}$ is generally distinct from $\mathbf{p}' = \mathbf{T}^T\mathbf{p}$, unless $\pi_1 = \pi_2$ or $\mathbf{p}$ is on line $\mathbf{l}$.

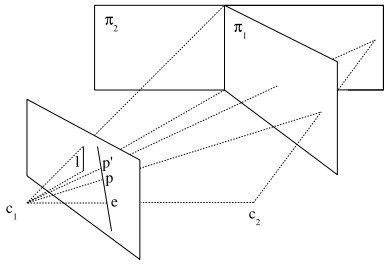
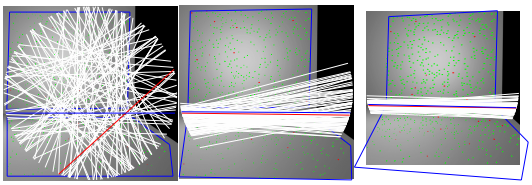


Figure 1. Fixed lines of the homology.

Preprint submitted to 19th International Conference on Pattern Recognition.
Received April 7, 2008.



frame #1 frame #10 frame #20

Figure 2. Images of a synthetic sequence.

3 Particle filtering of the intersection line

We now assume we are able to compute several pairs of homographies $\mathbf{H}_1^i$, $\mathbf{H}_2^i$ between I_1 and subsequent images I_i. Temporal consistency can therefore be exploited to get an accurate and robust computation of line $\mathbf{l}$. We use particle filtering (PF) which is a well known technique for implementing a recursive Bayesian filter by Monte Carlo simulations [1]. The key idea is to represent the required posterior density function $p(\mathbf{x}_i|\mathbf{z}_{1:i})$, where $\mathbf{z}_{1:i}$ is the set of all available measurements up to time i, by a set of random samples with associated weights w_i^j, and to compute estimates based on these samples and weights:

$$p(\mathbf{x}_i|\mathbf{z}_{1:i}) \approx \sum_{j=1}^{N} w_i^j \delta(\mathbf{x}_i - \mathbf{x}_i^j), \sum_{j=1}^{N} w_i^j = 1. \qquad (1)$$

We implement the generic PF according to the framework described in [1]. Resampling is used whenever a significant degeneracy is observed (i.e., when the effective sample size N_{eff} falls below some threshold N_t). Our implementation has the following characteristics:

(i) particles are homogeneous coordinates of lines. The first mode of distribution (1) is taken as the estimated line $\mathbf{l}$ (in image 1) at time i. Initially, the particles are uniformly distributed inside the largest ellipse $\mathcal{E}$ contained in the image (Fig.2, first frame);

(ii) the prior $p(\mathbf{x}_i|\mathbf{x}_{i-1})$ is the normal distribution centered at $\mathbf{x}_{i-1}$ with covariance matrix $\mathbf{V}$ ($\mathbf{V} = \text{diag}^2([0.01, 0.01, 5])$ in our experiments); the importance density is the prior;

(iii) the likelihood density at time i is given by:

$$p(z_i|\mathbf{x}_i) = p(z_i^g|\mathbf{x}_i)p(z_i^p|\mathbf{x}_i),$$

where z_i^g and z_i^p are (assumed independent) geometric and (resp.) photometric measurements we now detail.

Geometric likelihood. Measuring "how fixed" a line is when transformed by $\mathbf{T}$ provides a geometric measure of the likelihood of the related particle. In order not to confuse between the two kinds of fixed lines of $\mathbf{T}$ and according to Result 2, we actually measure the fixity of some points on the line. In practice, we found enough discriminant to measure the fixity of the intersection points $\mathbf{p}_1$ and $\mathbf{p}_2$ of the line with the ellipse $\mathcal{E}$:

$$p(z_i^g|\mathbf{x}_i) = \exp\left(-\frac{D^2}{2\sigma_g^2}\right), D = \frac{1}{2}\sqrt{\sum_{k=1}^{2} ||z(\mathbf{p}_k) - z(\mathbf{T}^T\mathbf{p}_k)||^2}$$

where $||.||$ denotes the L2 norm of a vector, $z\left([x,y,z]^T\right) = [x/z, y/z]^T$ and $\sigma_g = 3$ in our experiments.

Photometric likelihood. Accuracy and convergence of the PF can be increased by also measuring the distance of the particles to the highest gradients of the image. This is done by Sobel filtering, hysteresis thresholding and lines detection using a fast Hough Transform (HT). A significant pruning is obtained by computing a single global HT updated from frame to frame by transferring the line candidates of image i to image 1, using the homography $\mathbf{H}_k^{i\,-T}$, $k = 1$ or $k = 2$: doing that, only the projections of the lines on plane π_k contribute to the local maxima of the HT (other lines are transferred to unstable coordinates of the HT). Finally, we keep the lines $\mathbf{m}_i^j$ corresponding to the M greatest local maxima of the HT ($M = 100$ in our experiments). This leads to:

$$p(z_i^p|\mathbf{x}_i) = \exp\left(-\frac{D^2}{2\sigma_p^2}\right), D = \frac{1}{2}\min_{j=1}^{M}\sqrt{\sum_{k=1}^{2}(\mathbf{m}_i^j|\mathbf{p}_k)^2}, \quad (2)$$

where $(.|.)$ denotes the dot product, $\mathbf{m}_i^j$ is expressed under the form $[\cos(\theta), \sin(\theta), -\rho]^T$ and $\sigma_p = \sigma_g$ in our experiments. This measure benefits from the robustness of the HT and can therefore tolerate partial occlusions of the intersection line.

4 Euclidean reconstruction

Knowing the projection $\mathbf{l}$ in I_1 of the intersection line between two planes π_1 and π_2 and a pair of related homographies $\mathbf{H}_1^i$ and $\mathbf{H}_2^i$ allows to reconstruct the planes in the view coordinate system. Here and in the rest of the paper, we assume that the camera intrinsic parameters of the camera are known and the image coordinates are affine-transformed using the inverse intrinsic matrix [4].

When the equation of one plane (say π_1) and the camera motion $\mathbf{R}, \mathbf{t}$ between I_1 and I_i are known, computation of π_2 is straightforward: as shown in Fig. 1, π_2 belongs to a sheaf of planes passing through the 3D intersection line between π_1 and the plane passing through $\mathbf{l}$ and the camera center $\mathbf{c}_1$. This is algebraically expressed as:

$$\mathbf{\Pi}_2 = \mathbf{\Pi}_1 + \lambda[\mathbf{l}^T 0]^T \qquad (3)$$

Preprint submitted to 19th International Conference on Pattern Recognition.
Received April 7, 2008.

where $\mathbf{\Pi}_1 = [\mathbf{n}_1^T, d_1]^T$ and $\mathbf{\Pi}_2 = [\mathbf{n}_2^T, d_2]^T$ are the equation vectors of the planes expressed in the first view coordinate system. In the common case where π_2 is orthogonal to π_1, we directly obtain $\mathbf{\Pi}_2$ using $\lambda = -1/(\mathbf{n}_1|\mathbf{l})$. In the general case, λ can be determined performing a 1-parameter LMS optimization of the transfer error of $n \geq 4$ points $\mathbf{v}_i$ on π_2 (for instance, the vertices of the planar region). Indeed, any value of λ induces a homography $\mathbf{H}(\lambda) = d_2(\lambda)\mathbf{R} + \mathbf{t}\mathbf{n}_2(\lambda)^T$ [4], providing transfer errors $||z\left(\mathbf{H}(\lambda)\mathbf{v}_i\right) - z\left(\mathbf{H}_2\mathbf{v}_i\right)||$.

When no information is available about the camera-planes geometry, structure and motion are computed using a higher degree optimization that requires an initial estimate. It is shown in [4] that the simultaneous estimate of the camera motion and the plane pose corresponding to a homography has in general two physical solutions which can be obtained using SVD. As we know two homographies, this twofold ambiguity can be removed by finding the common solution for camera motion: this provides initial values for $\mathbf{R}, \mathbf{t}$ and the structure $\mathbf{n}_1, \mathbf{n}_2, d_2$ of the planes (d_1 determines the scale of the scene and is set to the assumed value of the height of the camera in I_1). These values are then refined performing a 9 or 8-parameters optimization (parameters are $\mathbf{R}, \mathbf{t}, \mathbf{n}_1$ and λ when the angle between π_1 and π_2 is unknown) of the transfer error of $n \geq 4$ points on π_1 and $m \geq 4$ points on π_2 using the Levenberg-Marquardt algorithm. It is shown in section 5 that a 9-parameter optimization converges faster and more accurately than a 11-parameter optimization that does not handle the knowledge of line $\mathbf{l}$.

5 Synthetic results

Filtering parameters. Synthetic tests have been performed in order to assess the effects of the number of particles N and the resampling threshold N_t over the convergence of the PF. A 80-frame sequence was used in which the camera followed a circular path while pointing toward a horizontal and a vertical plane (Fig. 2). A Gaussian noise of standard deviation 0.3 was added to the coordinates of the points used to compute the inter-image homographies. Fig. 2 shows examples of particle distributions obtained in three images of the sequence. Figure 3 shows the mean number of frames (over 100 tests) needed to reach the convergence, for N varying from 20 to 1000 and N_t from 0 to N. The convergence is always reached, even for small values of N (except for $N_t = 0$ due to the degeneracy problem). However, the convergence is faster for high values of N and when N_t is closer to N. These results led us to use $N_t = N = 1000$ in our real experiments.

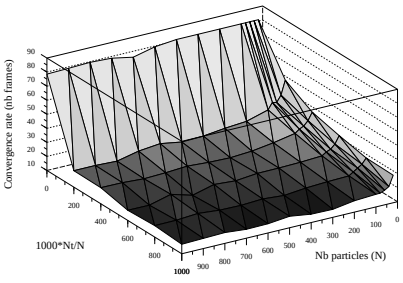


Figure 3. Convergence of the PF.

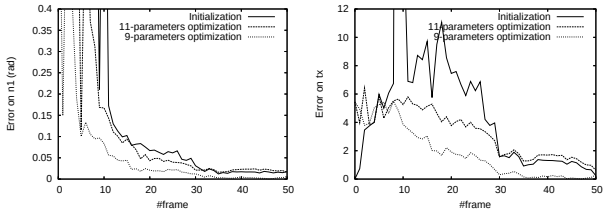


Figure 4. Structure and motion errors.

Euclidean reconstruction. Figure 4 shows the errors obtained on the normal to the horizontal plane and the x-coordinate of the camera translation when computing structure and motion between the first frame of the synthetic sequence and the next 50 frames. Errors are shown for the SVD, the 11-parameters optimization and the 9-parameters optimization (in that case the intersection line is extracted from a HT in the first frame). This graphic shows that the 9-parameters optimization can substantially improve the accuracy, especially when the baseline is small (except for too small baselines which lead to unstable results). Moreover, the mean number of iterations of the Levenberg-Marquardt algorithm is 3.9 for the 9-parameters optimization, against 6.7 for the 11-parameters optimization.

6 On-the-fly map building

The previous theoretical results have been used to design an integrated system for building a multiplanar model of the scene as the camera is localized on the fly. The main characteristic of this system is that the user is able to assist the mapping tasks. In particular, visual assessments of the filtered intersection lines greatly reduce map corruptions. An intuitive interface controlled by only four keys is used to define blobs, indicate the PF convergence, and validate or invalidate the initial and further Euclidean reconstructions. All these interactions can be done on the fly while the camera is moving. As no mouse interaction is needed, the interface

Preprint submitted to 19th International Conference on Pattern Recognition.
Received April 7, 2008.

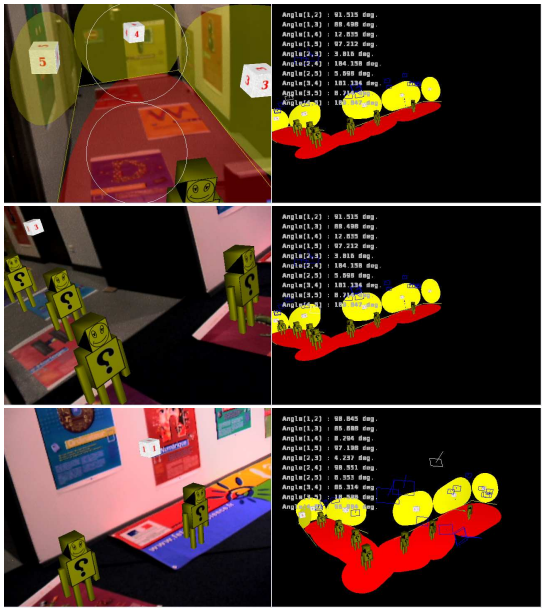


Figure 5. On-the-fly map building.

is particularly well adapted to applications of AR that run on modern devices such as PDA or mobile phones equipped with digital cameras.

A "ground and wall" type of scene is considered: two circles are displayed, one on the bottom half and the other on the top half of the screen, that allow the user to "capture" blobs on the ground plane and (resp.) the vertical planes (see Fig. 5, top frame). These blobs are tracked using the method presented in [7]. Initially, planes poses and camera motion are computed from a vertical and a horizontal blob by filtering their intersection line and performing the SVD + 8-parameters optimization. Then the camera pose is updated in real time using the existing planes in the map. When a new horizontal blob is defined, it is back-projected using the known equation of the horizontal plane in the camera coordinate system. When a new vertical blob is defined, the intersection line with the ground plane is filtered. If this line is aligned with another intersection line in the map, merging with the related plane is proposed. Otherwise, a new blob is added to the map using equation (3). Keyframes with SIFT features are saved during the process and these are used upon user request for global relocalization of the camera and bundle adjustments of the poses of all the planes in the map (in the spirit of [6]).

This system has been used in a two-room scene (Fig. 5). Computation rates were about 12 Hz in tracking mode and 8 Hz in tracking + filtering mode on a PC Dell Precision 390, 2.93 Ghz. A hand-held Sony camera DFW-VL500 was used at resolution 320x240. The

#frame	N_K	N_B	Event	Error angles (deg)			
				(1,2)	(1,3)	(1,4)	(1,5)
675	3	3	P2 added	-10.3			
891	7	3	Bundle (0.3 s)	1.5			
2710	14	10	P3 added	1.5	-1.5		
2954	16	11	P4 added	1.5	-1.5	12.6	
5648	17	12	P5 added	1.5	-1.5	12.6	7.2
7842	18	13	Bundle (2.1 s)	0.9	-3.2	0.6	7.3

Table 1. Error angles between the planes.

session lasted about 10 minutes: a video is associated to the paper[1] showing the most interesting parts of this session. 13 blobs on 6 different planes were defined and successfully reconstructed. Virtual objects were automatically added on the middle of each blob. As one can see in the video, these appear firmly anchored in the scene, and camera tracking performs well despite erratic motions of the hand-held camera. Table 1 shows the error angles obtained between the first vertical plane added to the map and the other vertical planes (plane numbers are those shown in the video).

7 Conclusion

We presented a method for tracking and mapping in multiplanar environments, which has been validated on both synthetic and real-size (spatially and temporally) experiments. This method may be extended to allow detection and reconstruction of other kinds of features like the edges of the scene. Using the system in larger environments (a complete level of a building for instance) would also require improvements: in order to keep a reasonable rate of exploration, the system should be allowed to perform local bundle adjustments as in [6].

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[3] E. Eade and T. Drummond. Scalable monocular SLAM. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–476, Washington, DC, USA, 2006. IEEE Computer Society.

[4] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Rapport de recherche 856, INRIA, 1988.

[5] B. Johansson. View synthesis and 3d reconstruction of piecewise planar scenes using intersection lines between the planes. In *ICCV*, pages 54–59, 1999.

[6] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07), Nara, Japan*, November 2007.

[7] F. Vigueras, M.-O. Berger, and G. Simon. Iterative multi-planar camera calibration: Improving stability using model selection. In E. Association, editor, *Vision, Video and Graphics (VVG'03), Bath, UK*, Jul 2003.

[1] `http://www.loria.fr/~gsimon/icpr08`