Real-Time Localization and 3D Reconstruction

Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, Patrick Sayd

Real Time Localization and 3D Reconstruction

E. Mouragnon[1,2], M. Lhuillier[1], M. Dhome[1], F. Dekeyser[2], P. Sayd[2]
[1]LASMEA UMR 6602, Université Blaise Pascal/CNRS, 63177 Aubière Cedex, France
[2]Image and embedded computer lab., CEA/LIST/DTSI/SARC, 91191 Gif s/Yvette Cedex, France

Abstract

In this paper we describe a method that estimates the motion of a calibrated camera (settled on an experimental vehicle) and the tridimensional geometry of the environment. The only data used is a video input. In fact, interest points are tracked and matched between frames at video rate. Robust estimates of the camera motion are computed in real-time, key-frames are selected and permit the features 3D reconstruction. The algorithm is particularly appropriate to the reconstruction of long images sequences thanks to the introduction of a fast and local bundle adjustment method that ensures both good accuracy and consistency of the estimated camera poses along the sequence. It also largely reduces computational complexity compared to a global bundle adjustment. Experiments on real data were carried out to evaluate speed and robustness of the method for a sequence of about one kilometer long. Results are also compared to the ground truth measured with a differential GPS.

1. Introduction

During last years, many works [8, 4] were carried out on the robust and automatic estimate of the movement of a perspective camera (calibrated or not) and points of the observed scene, from a sequence of images. It is still today a very active field of research, and several successful systems currently exist [13, 1, 12, 9, 15]. Interest points are initially detected and matched between successive images. Then, robust methods proceeding by random samples of these points make possible to calculate the geometry of subsequences of 2 and 3 images. Lastly, these "partial" geometries are merged and the reprojection errors (due to the difference between points detected in the images and the reprojections of 3D points through the cameras) are minimized.

This paper deals with the problem of scene reconstruction from images obtained by a moving calibrated camera. The reconstruction consists in finding the 3D model of the environment, by using only the recorded data. Many applications (architecture, navigation of robots, etc.) require the use of such a model. The problem often takes the SFM denomination for Structure From Motion, which was the subject of many works in vision.

One can note several types of approaches for SFM algorithms. First of all, the methods without global optimization of the full geometry are fast but their accuracy is questionable since errors accumulate in time. Among those works of Vision-Based SLAM (Simultaneous Localization and Mapping), Nistér [11] presents a method called "visual odometry". This method estimates the movement of a stereo head or a simple camera in real time from the only visual data: the aim is to guide robots. Davison [2] proposes a real time camera pose calculation but he assumes that number of landmarks is small (under about 100 landmarks). This approach best suite to indoor environments and is not appropriate for long displacements because of algorithmic complexity and growing uncertainty.

With a really different approach, we can find off-line methods carrying out a bundle adjustment optimization of the global geometry in order to obtain a very accurate model (see [18] for a very complete survey of bundle adjustment algorithms). Such an optimization is very computing time expensive and can not be implemented in a real time application. Bundle adjustment is a process which adjusts iteratively the pose of cameras as well as points position in order to obtain the optimal least squares solution.

Most articles refer to Levenberg-Marquardt (LM) to solve the non linear criterion involved in bundle adjustment, a method which combines the Gauss-Newton algorithm and the descent of gradient. The main problem in bundle adjustment is that it is very slow, especially for long sequences because it requires inversion of linear systems whose size is proportional to the number of estimated parameters (even if one benefits from the sparse structure of the systems).

It is also important to have an initial estimate relatively close to the real solution. So, applying a bundle adjustment in a hierarchical way is an interesting idea [8, 16] but it does not solve the computing time problem. Thus, it is necessary to take an alternative method whose purpose is to decrease the number of parameters to be optimized. Shum [16] ex-

ploits information redundancy in images by using two virtual key frames to represent a sequence.

In this paper we propose an accurate and fast incremental reconstruction and localization algorithm. The idea of an incremental method for a 3D reconstruction and motion estimation has already been developed in many ways. Steedly [17] proposes an incremental reconstruction with bundle adjustment where he readjusts only the parameters which have changed. Even if this method is faster than a global optimization, it is not sufficiently efficient and very data dependent. There are also Kalman filters or extended Kalman filters [2], but they are known to provide less accurate results than bundle adjustment. Our idea is to take benefit from both offline methods with bundle adjustment and from speed of incremental methods. In our algorithm, a local bundle adjustment is carried out at each time a new camera pose is added to the system. The nearest approach is proposed by Zhang [19], but in this work, local optimization is done on a triplet of images only, and structure parameters are eliminated from the proposed reduced local bundle adjustment. Taking into account of 2D reprojections of 3D estimated points in more than three images without eliminating the 3D points parameters greatly improves the accuracy of the reconstruction.

The paper is organized as follows. First, we present our complete method to estimate camera motion and 3D structure from a video flow. We explain our incremental method with local bundle adjustment: we propose to only optimize the end of the 3D structure with a set of parameters restricted to the last cameras and 3D points observed by these cameras. In a second part, we present experiments and results on real data, and we compare to a GPS localization.

2. Description of the incremental algorithm

Let us consider a video sequence acquired with a camera settled on a vehicle moving in an unknown environment. The goal of this work is to find the position and the orientation in a global reference frame of the camera at several times t as well as the 3D position of a set of points (viewed along the scene). We use a monocular camera whose intrinsic parameters are known (including radial distortion) and assumed to be unchanged throughout the sequence.
The algorithm begins with determining a first triplet of images that will be used to set up the global frame and the system geometry. After that, a robust pose calculation is carried out for each frame of the video flow using features detection and matching. Some of the frames are selected and become key-frames that are used to 3D points triangulation. The system operates in an incremental way, and when a new key-frame and 3D points are added, we proceed to a local bundle adjustment. The result (see Figure 3) is a set of camera poses corresponding to key-frames and 3D coordinates of points seen in images.

2.1. Interest points detection and matching

The whole method is based on the detection and matching of features points. In each frame, Harris corners [7] are detected. Matching a pair of frames is done as follows:

- For each interest point in *image* 1, we select some candidate corresponding points in a region of interest defined in *image* 2
- Then a Zero Normalized Cross Correlation score is computed between interest points neighborhoods.
- The pairs with the high-scores are selected to provide a list of corresponding point pairs between the two images.

In order to suite to a real time application, the step "detection and matching" has been implemented using SIMD extensions of modern processors. That provides a very efficient solution and not much time consumption.

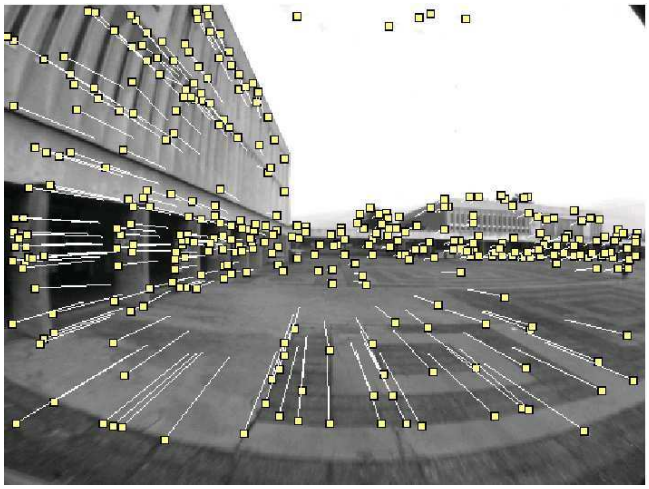

Figure 1. An example of image from the video data. Small squares represent detected interest points, and white lines represent distance covered by matched points.

2.2. Sequence initialization

We have in mind that the motion between two consecutive frames must be sufficiently large to compute the epipolar geometry. So we select frames relatively far from each other but that have enough common points. For that, the first image noted I_1 is always selected as a key frame. The second image I_2 is selected as far as possible from I_1 in the video but with at least M matched interest points with I_1. Then for I_3, we choose the frame most distant from I_2 so that there are at least M matched interest points between I_3 and I_2 and at least M' matched points between I_3 and I_1 (in our experiments, we choose $M = 400$ and $M' = 300$). Actually, this process ensures to have a sufficient number

of points in correspondence between frames to calculate the movement of the camera. The camera coordinate system associated to I_1 is set as the global coordinate system and the relative poses between the first three key frames is calculated using the 5-points algorithm [10] and a RANSAC [5] approach. More details on the initialization process are developed in [15]. Then, observed points are triangulated into 3D points using the first and the third observation. Finally an optimization of estimated poses and 3D points coordinates is done with a Levenberg-Marquardt algorithm (LM).

2.3. Real-time robust pose estimation

Let us suppose that pose of cameras C_1 to C_{i-1} corresponding to selected key-frames I_1 to I_{i-1} have previously been calculated in the reference reconstruction frame. We have also found a set of points whose projections are in the corresponding images. The goal is to calculate camera pose C corresponding to the last acquired frame I. For that, we match I (last acquired frame) and I_{i-1} (last selected key frame) to determine a set of points p whose projections on the cameras (C_{i-2} C_{i-1} C) are known and whose 3D coordinates have been computed before. From 3D points reconstructed from C_{i-2} and C_{i-1}, we use Grunert's pose estimation algorithm as described in [6] to compute the location of camera C. A RANSAC process gives an initial estimate of camera C pose which is then refined using a fast LM optimization stage with only 6 parameters (3 for optical center position and 3 for orientation). At this stage, the covariance matrix of camera C pose is calculated by the hessian inverse and we are able to draw an ellipsoid of confidence at 90% (see Figure 2). If Cov is the covariance matrix of camera C pose, the ellipsoide of confidence is given by $\Delta x^T Cov^{-1} \Delta x \leq 6.25$ since $\Delta x^T Cov^{-1} \Delta x$ obeys a $\mathcal{X}_2$ distribution with 3 dof.

2.4. Key frames selection and 3D points reconstruction

As mentioned before, not all the frames of the input are taken into account for the 3D reconstruction, but only a subsample of the video. For each frame, the normal way is to compute the corresponding localization using the last two key frames. We set up a criterion that indicates if a new frame must be added as a key frame or not. First, if the number of matched points with the last key frame I_{i-1} is not sufficient (typically inferior to a fixed level M, $M = 400$ in experiments), we have to introduce a new key-frame. We have also to take a new key frame if the the uncertainty of the calculated position is too high (for example, superior to the mean inter-distance between two consecutive key positions). Obviously, it is not the frame for which criterion is refused that becomes a key frame but the one which immediately precedes. After that, new points (ie. those which are only observed in I_{i-2}, I_{i-1} and I_i) are reconstructed using a standard triangulation method.

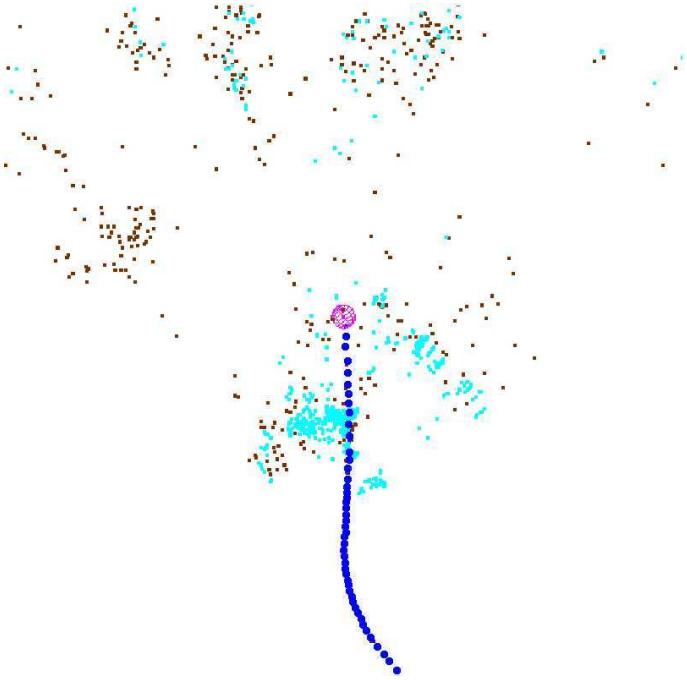

Figure 2. Top view of a processing reconstruction. We can see the trajectory, 3D reconstructed points and the ellipsoid of confi dence for the most recently calculated camera.

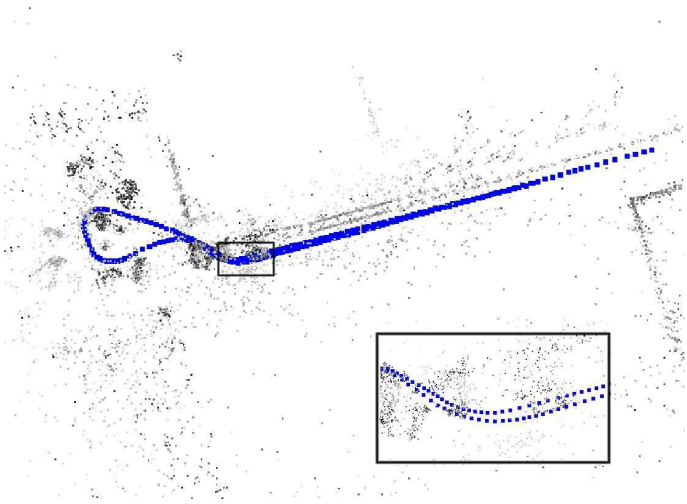

Figure 3. Top view of a complete reconstruction in a urban environment. The distance covered is about 200 meters long including a half-turn. More than 8.000 3D points have been reconstructed for 240 key frames.

2.5. Local bundle adjustment

When the last key frame I_i is selected and added to the system, a stage of optimization is carried out. It is a

bundle adjustment or Levenberg-Marquardt minimization of the cost function $f^i(\mathcal{C}^i, \mathcal{P}^i)$ where $\mathcal{C}^i$ and $\mathcal{P}^i$ are respectively the cameras parameters (extrinsic parameters) and 3D points chosen for this stage i. The idea is to reduce the number of calculated parameters in optimizing only the extrinsic parameters of the n last cameras and taking account of the 2D reprojections in the N (with $N \geq n$) last frames (see Figure 4). Thus, $\mathcal{C}^i = \{C_{i-n+1}..C_i\}$ and $\mathcal{P}^i$ contains all the 3D points projected on cameras $\mathcal{C}^i$. Cost function f^i is the sum of points $\mathcal{P}^i$ reprojection errors in the last frames C_{i-N+1} to C_i:

$$f^i(\mathcal{C}^i, \mathcal{P}^i) = \sum_{C_i \in \{C_{i-N+1}\ ;\ C_i\}} \sum_{p_j \in \mathcal{P}^i} (\varepsilon_{ij}^2)$$

where $\varepsilon_{ij}^2 = d^2(p_{ij}, K_i p_j)$ is the square of Euclidean distance between $K_i p_j$, estimated projection of point p_j through the camera C_i and the measured corresponding observation. K_i is the projection matrix 3×4 of camera i composed of C_i extrinsic parameters and known intrinsic parameters.

Thus, n (number of optimized cameras at each stage) and N (number of images taken into account in the reprojection function) are the 2 main parameters involved in the optimization process. Their given value can influence the result quality and execution speed. Experiments permitted to determine what are values for n and N that provide an accurate reconstruction.

It is important to specify that when the reconstruction process starts, we refine not only the last parameters of the sequence, but the very whole 3D structure. Thus, for $i \leq N_f$, we chose to take $N = n = i$. N_f is the maximum number of cameras so that optimization at stage i is global (in our experiments, we choose $N_f = 20$). That makes it possible to have reliable initial data, which is significant given the recursive aspect of the algorithm, and that does not pose any problem because the parameters number is still relatively restricted at this time.

Comparison with global bundle adjustment:

Because of 3D points independence, we can take advantage [8, 18] of the sparse structure of the Jacobian matrix J of the error measure vector ε. So, we have implemented the sparse minimization algorithm as described in [8]. The system to be solved for each Levenberg-Marquardt iteration is:

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta_{cameras} \\ \Delta_{points} \end{pmatrix} = \begin{pmatrix} Y_{cameras} \\ Y_{points} \end{pmatrix}$$

where U, V, W are sub-matrix composing Hessian matrix $J^T J$, $\Delta_{cameras}$, Δ_{points} are increments to be calculated and $Y_{cameras}, Y_{points}$ are obtained by the product matrix $J^T \varepsilon$. The resolution is carried out with 2 stages:

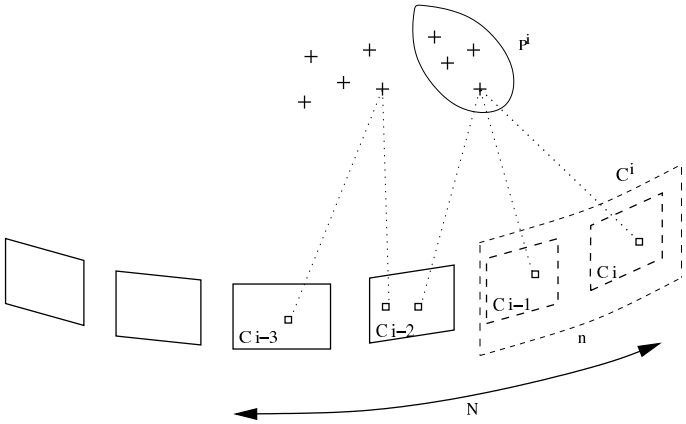

Figure 4. Local bundle adjustment when camera C_i is added. Only surrounded points and cameras are optimized. Nevertheless, we take account of 3D points reprojections in the N last images.

1. Calculation of the increment $\Delta_{cameras}$ to be applied to cameras by resolution of the following system:

$$(U - WV^{-1}W^T)\Delta_{cameras} = Y_{cameras} - WV^{-1}Y_{points} \quad (1)$$

2. Direct calculation of the increment Δ_{points} to be applied to 3D points:

$$\Delta_{points} = V^{-1}(Y_{points} - W^T \Delta_{cameras})$$

Let n and P be the number of cameras and points optimized in bundle adjustment. Let p be the number (considered as constant) of projecting points through each camera.

Once $J^T J$ is calculated (time complexity is proportional to the number $N_r = p.N$ of 2D reprojections taken into account), the two time computing expensive stages of this resolution are:

- The matrix product $WV^{-1}W^T$
- The resolution of cameras linear system (1).

For matrix product $WV^{-1}W^T$, the number of necessary operations can be given by first considering the number of not-null blocks of WV^{-1}. It is the same number as W, i.e. $(p.n)$, number of reprojections in n images, because V^{-1} is block diagonal. Then, in the product $(WV^{-1})W^T$, each not-null 6×3 block of WV^{-1} is used once in the calculation of each block column of $WV^{-1}W^T$. Thus the time complexity of the product $WV^{-1}W^T$ is $\Theta(p.n^2)$. The time complexity of the traditional resolution of the linear system (1), is $\Theta(n^3)$ [14].

So, the time complexity of one bundle adjustment iteration is: $\Theta(p.N + p.n^2 + n^3)$.

Thus, we can see that is very interesting to reduce the number of parameters (n and N) involved in the optimization process. For example, the complexity reduction compared to global bundle adjustment obtained with a sequence of 20 key frames and 150 2D reprojections per image is given in the following table:

Type	p	n	N	gain
global	150	20	20	**1**
reduced 1	150	5	20	**10**
reduced 2	150	3	10	**25**

Table 1. Complexity gain obtained with a reduced local bundle adjustment compared to global bundle adjustment for one iteration.

In practice, we note that the number of necessary iterations is quite low; it is due to the fact that, excepting the last added camera, all the cameras poses have already been optimized at stage $i-1, i-2, ...$

2.6. Method summary

The proposed method is summarized as follow:

1. Select a triplet of images that become first three key frames. Set up the global frame, estimate the relative pose, and triangulate 3D points.

2. For each new frame, match with last key frame and estimate the camera pose and uncertainty. Detect if a new key frame has to be selected. If not, repeat 2.

3. If a new key frame is selected, add precedent frame as new key frame, triangulate new points and make a local bundle adjustment. Repeat from 2.

3. Experiments on real data

We applied our incremental localization and mapping algorithm to a semi-urban scene. The goal is to evaluate robustness to perturbations in a complex environment and accuracy compared to ground truth provided by a Real Time Kinematics Differential GPS. The camera was settled on an experimental vehicle whose velocity is about 0.8 m/s. The covered distance is about 70 $meters$ and the video sequence is 1 min long (Image size is 512×384 $pixels$ at 7.5 fps). More than 4.000 3D points have been reconstructed and 94 images selected as key frames from a series of 445. This sequence is particularly interesting because of images contain (people walking in front of the camera, sunshine, etc...) that does not favors the reconstruction process. Moreover, the environment is more appropriate to a GPS localization because the satellites in the sky are not occulted by high buildings. It is also interesting because of the trajectory: a turn on the right, two turns on the left and a straight line.

Figure 5. 2 frames from real data experiments. We can see some pedestrians

3.1. Processing Time

In our experiments, we used a standard Linux PC (Pentium 4 at 2.8 GHz, 1Go of RAM memory at 800 MHZ). Image processing time through the sequence is reported in Figure 6. Time measured includes feature detection (#1500 Harris points per frame), matching, and pose calculation for all frames. For key frames, treatment time is longer (see Figure 6) because of points 3D reconstruction and local bundle adjustment. In this case, we took $n = 3$ (number of optimized camera poses) and $N = 10$ (number of cameras used for reprojection criterion minimization). We can note that speed results are very interesting with an average of 0.09 s for normal frames and 0.28 s for key frames (let us notice that time between two frames is 0.133 s at 7.5 fps). Results are reported in table 2.

Frames	Max Time	Mean Time	Total
Non-key frames	0.14	0.09	30.69
Key frames	0.43	0.28	26.29

Table 2. Results. Computation times are given in *seconds*.

3.2. Ground truth comparison

The calculated trajectory obtained with our algorithm was compared to data given by a GPS sensor. It is a Real Time Kinematics Differential GPS whose precision is about the inch in the horizontal plane. For the comparison, we applied a rigid transformation (rotation, translation and scale factor) to the trajectory as described in [3] to fit with GPS reference data. Figure 8 shows trajectory registration with GPS reference. As GPS positions are given in a metric frame we can compare camera locations and measure positioning error in meters. For camera key pose i, 3D position error is:

$$E_{i3D} = \sqrt{(x_i - x_{GPS})^2 + (y_i - y_{GPS})^2 + (z_i - z_{GPS})^2}$$

and 2D position error in horizontal plane is:

$$E_{i2D} = \sqrt{(x_i - x_{GPS})^2 + (y_i - y_{GPS})^2}$$

where x_i, y_i, z_i are estimated coordinates for camera pose i and x_{GPS}, y_{GPS}, z_{GPS} are corresponding GPS coordi-

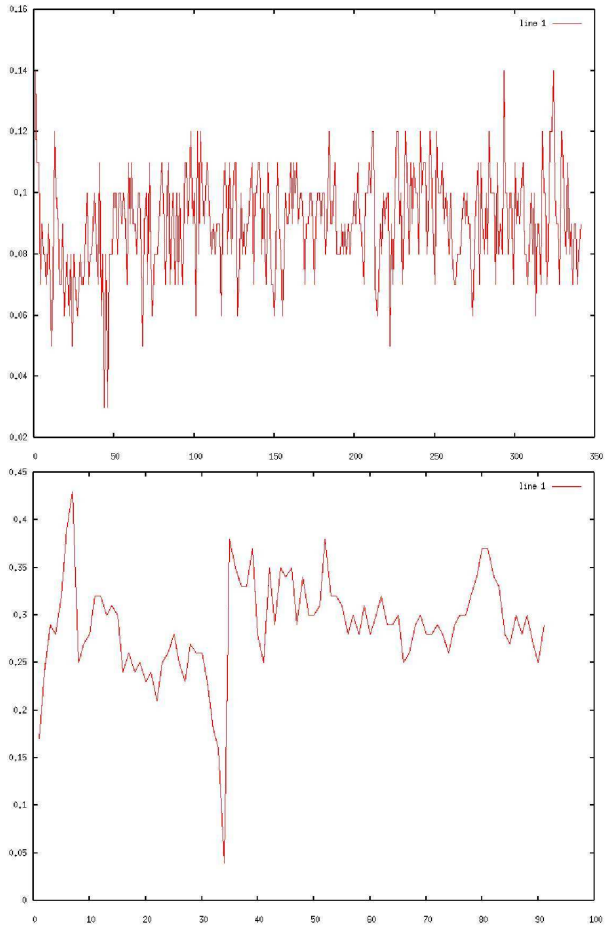

Figure 6. Processing time (in *seconds*) for non-key frames and for key frames.

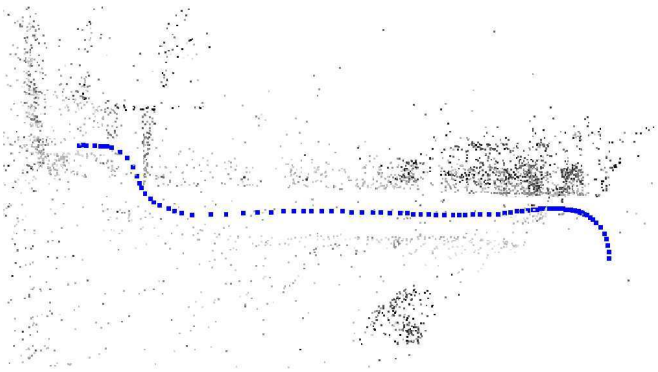

Figure 7. Top view of the reconstructed scene and trajectory (# 4.000 points and 94 key positions).

nates. Figure 9 shows 2D/3D error variations through the 94 frames. The maximum measured error is 2.0 *meters* with a 3D mean error of 41 *centimeters* and a 2D mean error of less than 35 *centimeters*.

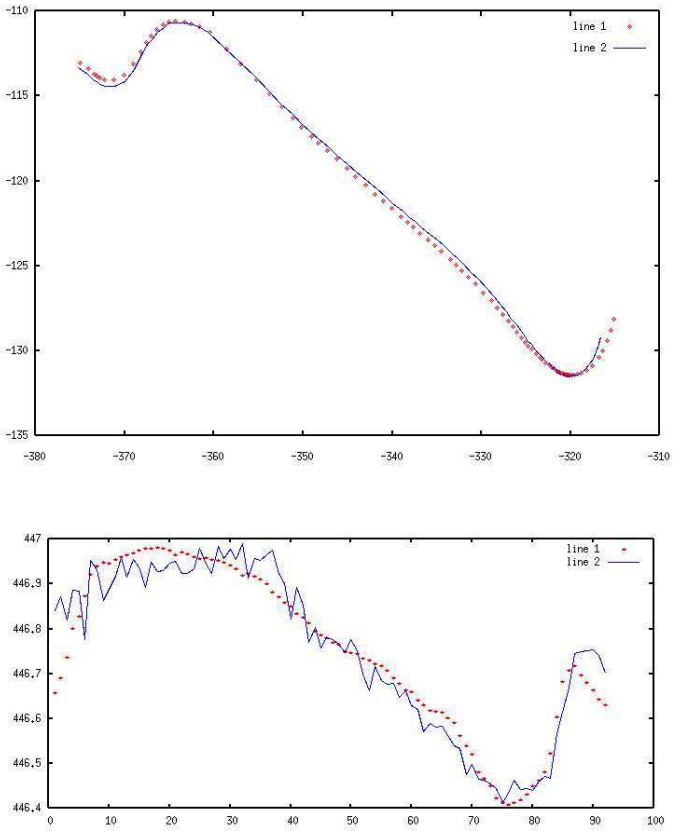

Figure 8. Registration with GPS reference, top: in horizontal plane, bottom: on altitude axis. Continuous line represents GPS trajectory and points represent estimated key positions. Coordinates are expressed in *meters*.

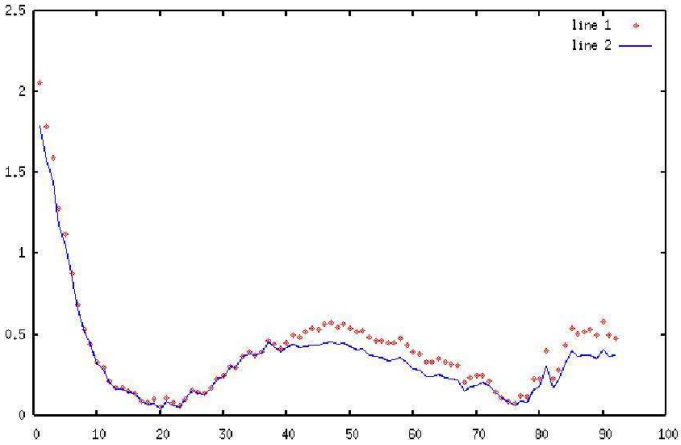

Figure 9. Error in meters. continuous line: 2D error, dotted line: 3D error.

3.3. Parameterizing the local bundle adjustment

In our incremental method, local bundle adjustment consists in optimizing the end of the reconstruction only, so as to avoid useless calculation and very long computing time.

As mentioned before, optimization is applied to the n last estimated camera poses, taking account of points reprojections in a larger number N of cameras. So, we tested several values for n and N as reported in table 3, 4 and 5. Note that we must have $N \geq n + 2$ to fix the reconstruction frame and the scale factor at the sequence end. First, we compared results to GPS trajectory as explained previously, and to a trajectory computed with a global bundle adjustment. We also measured mean time processing for local bundle adjustment in function of n. In practice, it does not vary much with N.

Comparison with GPS

n \ N	n	n+1	n+2	n+3	n+5	n+7
n=2	failed	failed	0.55	0.49	0.85	1.99
n=3	failed	3.28	0.45	**0.41**	**0.41**	**0.41**
n=4	6.53	1.77	0.42	**0.40**	**0.41**	**0.27**
global	0.33					

Table 3. Mean 3D position error (in *meters*) compared to GPS for the incremental method with different n and N, and for a global bundle adjustment.

Comparison with global bundle adjustment

n \ N	n	n+1	n+2	n+3	n+5	n+7
n=2	failed	failed	3.17	0.43	1.56	1.80
n=3	failed	3.61	1.60	**0.43**	**0.30**	**0.47**
n=4	7.67	1.44	1.03	**0.24**	**0.25**	**0.36**

Table 4. Mean 3D position error (in *meters*) compared to global bundle adjustment for different n and N.

n	2	3	4	5	6
Mean Time	0.24	0.31	0.33	0.37	0.44

Table 5. Mean local bundle adjustment computation times in function of n for many N(in *seconds*).

For $N = n$ or $n + 1$, it happened that the reconstruction was not completed because of the process failure before the end of the sequence. That proves that the fact of introducing a number $N \geq n+2$ is very important in a real environment and bring robustness and accuracy to the estimation. Moreover, with Figure 10, one can note that very many points are tracked over more than 3 views. If $n \geq 3$ and N quite larger, we have very accurate results; the problem is then time consumption. In our experiments, we often use $n = 3$ or 4 and $6 \leq N \leq 11$ (values in bold-faced in tables 3 and 4).

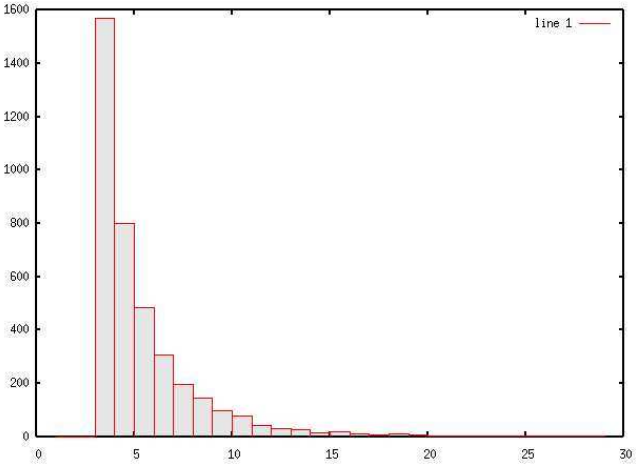

Figure 10. Points distribution with track length.

3.4. Visual comparison with a very long urban scene

Experiments have been carried out in a town-center (see Figure 12) with a camera fixed on a car. The vehicle ran about one kilometer and the video is about 3 min long. With Figure 11, one can visually ensure that reconstruction is not much deformed and drift is very low compared to the covered distance. That shows that our algorithm, very appropriate to long scene reconstruction in term of computing time is also quite precise and robust. The estimated mean 3D position error compared to global bundle adjustment is $0.29m$.

3.5. Conclusion

We presented a nice application of SFM techniques to localization and mapping, for a moving car. The method is very fast and accurate, thanks to the proposed local bundle adjustment technique. The model is built in real-time with 3D points reconstructed from interest points extracted in images and matched through the monocular video sequence. We can envisage to extend the method to more complex 3D primitives such as planes, lines, or curves. We think that the approach can be adapted to many applications in robotics to guide mobile robots, or in augmented reality applications.

References

[1] "Boujou," *2d3 Ltd*, http://www.2d3.com, 2000. 1

[2] A.J. Davison, "Real-Time Simultaneous Localization and Mapping with a Single Camera," *Proc. ICCV*, Nice, 2003. 1, 2

[3] O.D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *International Journal of Robotic Research*, Vol 5, No. 3, pp. 27-52, 1986. 5

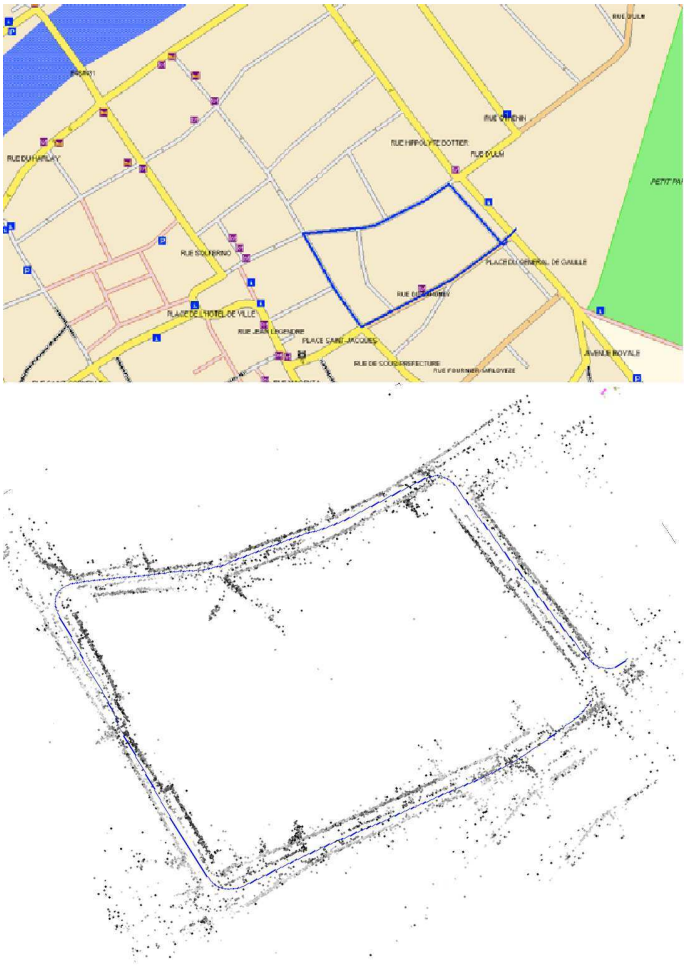

Figure 11. The long urban sequence a) top: a city map with the trajectory in blue b). bottom: reconstruction result (354 key frames, 16.135 3D points).

Figure 12. 2 frames from urban sequence.

[4] O.D. Faugeras and Q.T. Luong, *The Geometry of Multiple Images,* The MIT Press, 2001 1

[5] M. Fischler and R. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis ans Automated Cartography", *Commun. Assoc. Comp. Mach.*, 24:381-395, 1981 3

[6] R.M. Haralick, C.N. Lee, K. Ottenberg and M. Nolle, "Review and analysis of solutions of the three point perspective pose estimation problem," *"International Journal of Computer Vision,"* 1994 3

[7] C. Harris, M. Stephens, "A Combined Corner and Edge Detector", *Alvey Vision Conference* pp. 147-151, 1998. 2

[8] R.Hartley and A.Zisserman, *"Multiple View Geometry in Computer Vision,"* Cambridge University Press, 2000. 1, 4

[9] M. Lhuillier and Long Quan. "A Quasi-Dense Approach to Surface Reconstruction from Uncalibrated Images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(3):418-433, 2005. 1

[10] D. Nister, "An efficient solution to the five-point relative pose problem," *CVPR03*, pp. 195-202, 2003 3

[11] D. Nister, O. Naroditsky and J. Bergen, "Visual Odometry," *CVPR04*, Vol. 1, pp. 652-659, 2004. 1

[12] D. Nister *Automatic Dense Reconstruction from Uncalibrated Video Sequences*, PhD Thesis, Ericsson and University of Stockholms, 2001. 1

[13] M. Pollefeys, R. Koch and L. Van Gool, "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters," *ICCV'98*. 1

[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *"Numerical Recipes in C: The Art of Scientific Computing"*, Cambridge University Press, 1992. 4

[15] E. Royer, M. Lhuillier, M. Dhome and T. Chateau. "Localization in urban environments: monocular vision compared to a differential GPS sensor," *CVPR'05*. 1, 3

[16] H. Shum, Q. KE, and Z. Zhang, "Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion," *CVPR99*, Vol 2, pp. 538-543, 1999. 1

[17] D. Steedly and I. A. Essa, "Propagation of Innovative Information in Non-Linear Least-Squares Structure from Motion," *ICCV*, pp. 223-229, 2001. 2

[18] B. Triggs, P. F. McLauchlan, R. I. Hartley & A. W. Fitzibbon, "Bundle adjustment - A modern synthesis, in Vision Algorithms: Theory and Practice", *LNCS*, pp. 298-375, Springer Verlag, 2000. 1, 4

[19] Z. Zhang and Y. Shan, "Incremental Motion Estimation through Modified Bundle Adjustment", *In Proc. International Conference on Image Processing (ICIP03)*, Vol.II, pp.343-346, 2003. 2