3D reconstruction of complex structures with bundle adjustment: an incremental approach

Etienne Mouragnon*, Maxime Lhuillier*, Michel Dhome*, Fabien Dekeyser†, Patrick Sayd†
*LASMEA UMR 6602, Université Blaise Pascal/CNRS, 63177 Aubière Cedex, France
†Lab. Calculateurs Embarqués et Image, CEA Saclay, 91191 Gif s/Yvette, France

***Abstract*— This paper introduces an incremental method for "Structure From Motion" of complex scenes from a video sequence. More precisely, we estimate the 3D positions of the viewed points in images and the camera positions and orientations through the sequence. The method can be seen as a fast but accurate alternative to classical reconstruction methods that use bundle adjustment, and that can become slow and computation time expensive for very long scenes. Our results are compared to the reconstruction obtained by the classical hierarchical bundle adjustment method. They have also been successfully used as a reference sequence for the vision based localization of an autonomous mobile robot.**

I. Introduction

During last years, many works [7], [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, starting from a sequence of images. It is still today a very active field of research, and several successful systems currently exist [12], [1], [11], [8], [14]. 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 two types of approaches for SFM algorithms. First of all, the methods without global optimization of the full geometry are fast but their accuracy is not very good; errors accumulate in time. Among those works of Vision-Based SLAM (Simultaneous Localization and Mapping), Nistér [10] presents a method called "visual odometry". It is about the estimate of the movement starting from the only visual data. This method estimates the movement of a stereo head or a simple camera in real time: the aim is to guide robots. Davison [2] proposes a real time camera pose calculation but he assumes that the number of landmarks is small (under about 100 landmarks) which is not appropriate for long displacements. With a really different approach, we can find algorithms carrying out a bundle adjustment optimization [17], of the global geometry in order to obtain a very accurate model. Such an optimization is computing time expensive and is always carried out in an off line calculation. 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 system).

It is also important to have an initial estimate relatively close to the real solution. So, it could be an interesting idea to carry out a bundle adjustment in a hierarchical way [7], [15] but it does not solve the computing time problem. It is then necessary to take an alternative method whose purpose is to decrease the number of parameters to be optimized. Shum [15] exploits information redundancy in images by using two virtual key frames to represent a sequence. Steedly [16] 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. Zhang [18] presents an incremental method where a local optimization is done on a triplet of images only.

The goal of our study is to find a fast and reliable method for the reconstruction of long sequences. First, we present our incremental bundle adjustment: we propose to only optimize the end of the 3D structure with a set of parameters restricted to the last cameras and points observed by these cameras. In a second part, we evaluate this method and we compare it to a hierarchical bundle adjustment. Finally, we experiment this technique and test it on real data for mobile robot localization.

II. Reconstruction of complex scenes with bundle adjustment: an incremental approach

Let us consider a video sequence obtained from a camera moving in an unknown environment. The goal of the reconstruction 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

sequence of images acquired by a monocular camera whose intrinsic parameters are known. A preliminary calibration stage was carried out to determine the intrinsic parameters of the camera, and those parameters are assumed to be unchanged throughout the sequence.

In this study, we compare the incremental method suggested in the paper to a highly reliable method that makes use of a hierarchical bundle adjustment. In both cases, the reconstruction is based on the matching between image pairs. This matching is carried out from the detection of interest points (of Harris [6]) in each image. For each interest point in an image, we select possible corresponding points in an area of research of the following image. For each possible matching, we calculate a normalized and centered score of correlation to retain only the couples with the best score. These points will be reconstructed in 3 dimensions.

1) Key frames selection: For the reconstruction, we do not keep all the video frames but only one subsample of images. Indeed, the movement between two consecutive camera poses must be sufficiently large to ensure the calculation of the epipolar geometry. However, one must verify that the movement is not too large to calculate the matching between frames. The selection of key frames is carried out in an automatic way. 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 $n > 1$, we choose the key frame I_{n+1} most distant from I_n so that there are at least M matched interest points between I_{n+1} and I_n and at least M' matched points between I_{n+1} and I_{n-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 two key frames to calculate the movement of the camera.

2) Principle: The sequence has been sub-sampled in time and we have selected N_{seq} key frames containing interest points. Our aim is to estimate the cameras extrinsic parameters and the points 3D coordinates. It is impossible to compute all the camera positions parameters first and then to optimize, because the initial estimate would be too far away from the real solution and would make bundle adjustment diverge. For the reconstruction, we should proceed in a sequential way.

3) Hierarchical method: The hierarchical method divides a long sequence into several subsequences of three images. For each subsequence, the last two frames correspond to the first two images of the following subsequence. For the first three key frames, we calculate the camera's movement by calculating the essential matrix [9]. Then, for each triplet, the camera motion is obtained by a pose calculation. These computations produce an initial solution which is optimized by a bundle adjustment. Then, the sub-sequences are merged. The process is iterated until the whole sequence is obtained (Figure 1).

In our case, reconstruction is carried out in an incremental way, in the order of video frames. When adding new elements, we are confident in the structure obtained previously.

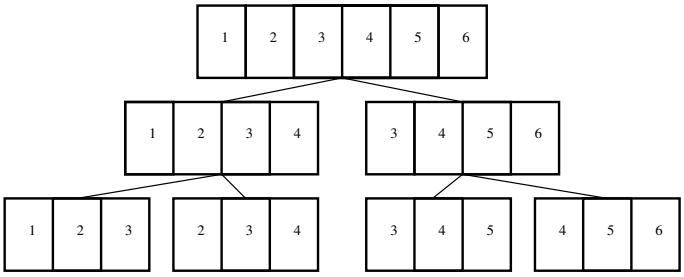

Fig. 1. Hierarchical reconstruction. Subsequences are optimized and merged.

4) Proposed incremental method: For the first image triplet, calculation is carried out by the method proposed by Nistér [9] for 3 images. It provides a solution for the first 3 camera poses noted C_1, C_2, and C_3, and for the position of points seen in these 3 images.
Then, the reconstruction extends to the following frames in a chronological order, and for each new key frame i:

1) We compute the new camera pose with visible points previously reconstructed.
2) New points are matched and reconstructed in 3D.
3) We apply a local bundle adjustment to refine 3D points and camera poses.

Stage i: adding camera i
Let us suppose that pose of cameras C_1 to C_{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_i and the new attached points. For that, we determine a set of points p^i whose projections on the cameras $(C_{i-2}\ C_{i-1}\ C_i)$ are known by the key frames selection and whose 3D coordinates have been computed. From C_{i-2} and C_{i-1}, we use Grunert's pose estimation algorithm as described in [5] to compute the location of camera C_i. New points that are matched only in images $i-2$, $i-1$, and i are then reconstructed in the reference frame.

Then comes the stage of 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 optimize extrinsic parameters of the n last cameras by taking account of the 2D reprojections in the N (with $N \geq n$) last frames (see Figure 2). Thus, $\mathcal{C}^i = \{C_{i-n+1}..C_i\}$ and $\mathcal{P}^i$ contains 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:

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

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 is directly correlated to the result quality and execution speed. An important task consists in determining what good values for n and N to provide a good accuracy.

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.

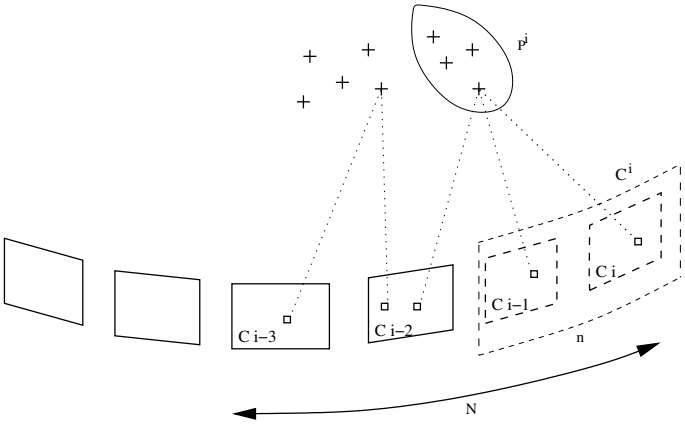

Fig. 2. The i^{th} stage of incremental reconstruction. Only surrounded points and cameras are optimized. Nevertheless, we take account of 3D points reprojections in the N last images.

III. Method Comparisons

A. Complexity of one bundle adjustment iteration

The goal of the proposed method is to accelerate the reconstruction process while preserving a given accuracy and reliable data. The Levenberg-Marquardt method is an iterative and well-known method for its convergence quality, and is very often used in this kind of problems.

Let $\mathbf{P}$ be the set of parameters to be estimated (cameras orientation + position of their optical center + 3D points coordinates), $\mathbf{X}$ the set of 3D points projections detected in images and $f(\mathbf{P})$ the projection of 3D points in images according to the parameters we are looking for. So, the problem is to minimize the function $\phi(\mathbf{P}) = ||f(\mathbf{P}) - \mathbf{X}||^2$. At stage k of the iterative algorithm, one calculates Δ_k such as $\mathbf{P}_{k+1} = \mathbf{P}_k + \Delta_k$. Δ_k is obtained by solving the equation: $J^T J.\Delta_k = J^T.\epsilon_k$ where J is the Jacobian matrix of f calculated in $\mathbf{P}_k$ and $\epsilon_k = \mathbf{X} - f(\mathbf{P}_k)$ (more precisely, the diagonal terms of matrix $J^T J$ are multiplied by a coefficient in the Levenberg-Marquardt method).

Since we want to deal with long sequences with a lot of parameters to be evaluated (6 parameters for each camera and 3 for each 3D point), it is quite naturally appropriate to exploit characteristics of bundle adjustment applied to the reconstruction of a set of points [7]: the block structure of matrix $J^T J$. This matrix is composed of three blocks U, V, and W such that U and V are block-diagonal:

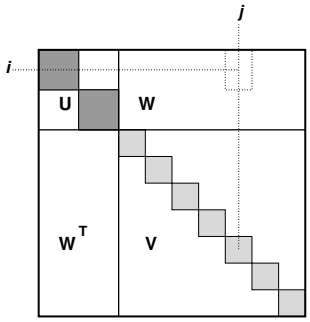

Fig. 3. Structure of $J^T J$ matrix

- U, matrix made of diagonal 6×6 blocks representative of the dependence between measurements of image i and associated camera's parameters.
- V, matrix made of diagonal 3×3 blocks representative of relations between point j parameters and measurements associated to it.
- W, matrix translating the intercorrelations between 3D points parameters and cameras parameters. The structure of W depends on the fact that many points are not projected through all the cameras. W has a number of not-null 6×3 blocks equal to the number of 2D reprojections.

So, the system is $\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta_{cameras} \\ \Delta_{points} \end{pmatrix} = \begin{pmatrix} Y_{cameras} \\ Y_{points} \end{pmatrix}$, and is solved in two steps [7]:

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 C 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 (Figure 3) (time complexity is proportional to the number N_r 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. $(p.C)$, number of reprojections in C 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 $\Theta(p.C^2)$. The time complexity of the traditional resolution of the linear system (1), is $\Theta(C^3)$ [13]. So, the time complexity of one bundle adjustment iteration is

$$\Theta(N_r + p.C^2 + C^3).$$

B. Calculation complexity for incremental reconstruction

Let us remind that the sequence to be reconstructed contains N_{seq} images. At each stage i (N_{seq} time), one adjusts poses of the n last cameras of a subsequence of N cameras, and one considers that the number of iterations k is the same one at each stage. Actually, the algorithm is stopped when the reprojection error in pixels does not decrease enough, i.e. when the ratio of error at stage i by the one at stage $(i-1)$ is higher than a coefficient α. In practice, we choose $\alpha = 0.9999$ and experiments have showed that iteration count k did not vary much. Thus, the resulting time complexity is

$$\Theta\left(k.N_{seq}(p.n^2 + n^3)\right).$$

It is also necessary to take into account the construction of matrix $J^T J$ whose number of operations is equal to the total number of reprojections in the N images, i.e. $(p.N)$ Finally, the complexity time is

$$\Theta\left(k.N_{seq}(p.N + p.n^2 + n^3)\right).$$

C. Calculation complexity for hierarchical reconstruction

Let us consider only the last stage where bundle adjustment is applied to the whole sequence on N_{seq} cameras. The number N_r of 2D reprojections taken into account, proportional to the total number $(p.N_{seq})$ of reprojections in the sequence, is negligible in front of $p.N_{seq}^2$. The number of iterations is noted k'. Thus, time complexity is

$$\Theta\left(k'(p.N_{seq}^2 + N_{seq}^3)\right).$$

D. Comparison

The calculation acceleration proposed by our method becomes interesting as soon as $k.N_{seq}(p.N + p.n^2 + n^3) = o\left(k'(p.N_{seq}^2 + N_{seq}^3)\right)$ when $N_{seq} \longrightarrow \infty$. For example, if we have $N \ll N_{seq}$ and $n^2 \ll N_{seq}$, then our method is acceptable. This is the case in practice since we have chosen constant n and N. It is also necessary that the numbers of iterations k and k' are equivalent. Actually, we could check that $k' > k$, what favors our method.

IV. Experiments

A. Intelligent vehicle localization from a 3D scene reconstructed by the incremental method

Within the framework of vision based localization of an autonomous mobile robot (Figure 4), one often uses a 3D map of the environment in which the robot evolves/moves, as well as the accurate trajectory it has to follow. This one is generated from an initial video recording carried out by an embedded camera and which is used as reference sequence for the localization. During robot's navigation, one is able to determine the current position in the reference frame by comparing images in memory (attached to the computed 3D structure) to the current images obtained with the same camera. The reference sequence calculated by a hierarchical bundle adjustment was replaced successfully in this study by the reconstruction obtained by our method. Images corresponding to this experiment are presented on Figure 5.

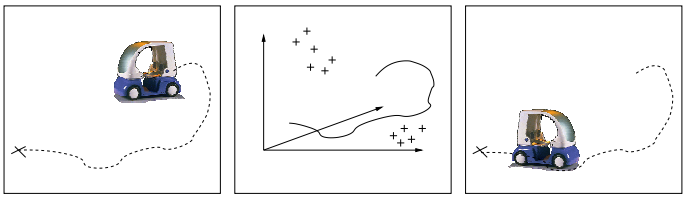

Fig. 4. Principle of vision based localization. 1) vehicle freely moves and records a video sequence. 2) trajectory and visual features of the video are reconstructed. 3) vehicle can localize itself and take the same path.

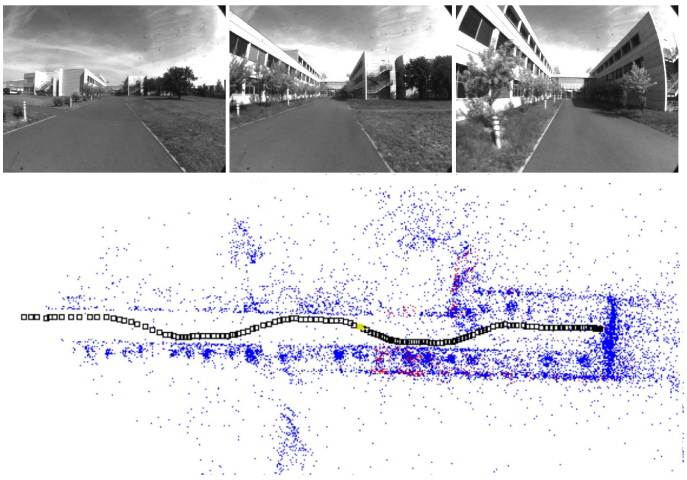

Fig. 5. 3 frames from "localization" sequence, with top view of reconstructed scene. Squares show camera position through the trajectory and points are 3D reconstructed points.

B. Comparison to hierarchical reconstruction

Figure 6 shows results obtained with a sequence of 93 key frames acquired with a camera mounted on a vehicle. Some sequence frames are visible in Figure 7. The real trajectory is closed. One can then see the "drift" of the reconstruction when parameters n (number of optimized cameras) and N (number of cameras taken into account with each stage) are not well selected.

Thereafter, we compare our method with the hierarchical reconstruction for a sequence of 112 key frames. The number of reconstructed 3D points is approximately 14900 and the trajectory is about 80 meters length (Figure 8). Many experiments are done for many values of n and N. We evaluate

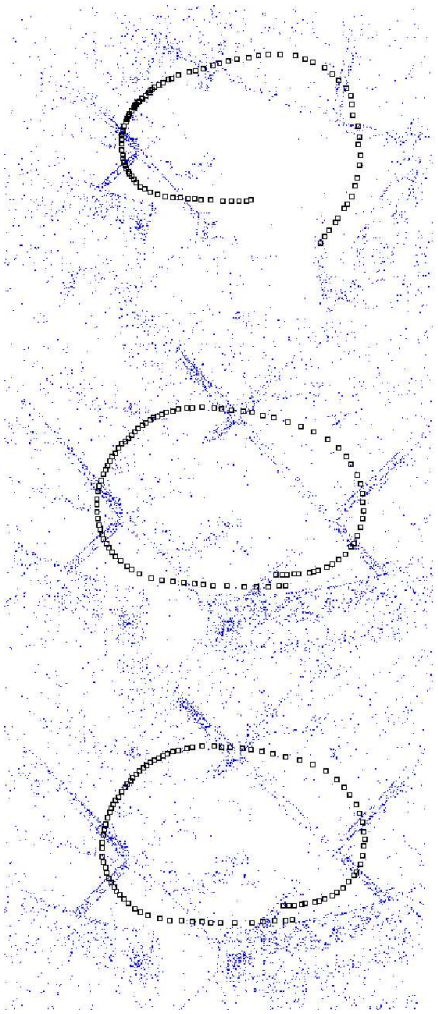

Fig. 6. Top: incremental reconstruction using n=3, N=3 (RMS=0.839 pixels). Middle: incremental reconstruction using n=3, N=10 (RMS=0.616 pixels). Bottom: hierarchical reconstruction (RMS=0.589 pixels).

Fig. 7. 3 frames from "loop" sequence.

and compare reprojection error, camera location and time computation.

1) Reprojection error comparison: Global reprojection error of the estimated 3D points in images is measured by coefficient $RMS = \sqrt{\frac{\sum_{i,j} \epsilon_{ij}^2}{N_r}}$ where N_r is the total number of reprojections. Results expressed in pixels are listed in table I. Values go from 0.978 pixels for the "worst" reconstruction with 0.602 pixels for the "best" one with our incremental approach. Note that RMS measured with hierarchical reconstruction is 0.589 pixels. Note also that, if we chose a value N not very higher than n, it is not interesting to select a high value for n. For example, if we take $N = n$, table I (first column) show that higher is n, higher is resulting reprojection error. It's due to the fact that we do not take enough account of structure's anteriority (considered as reliable) during optimization.

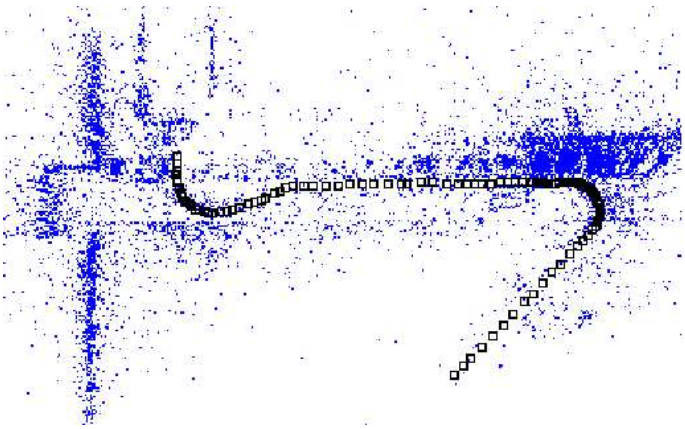

Fig. 8. Top view of reference reconstruction.

n \ N	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8
n=2	0.825	0.764	0.707	0.671	0.649	0.632	0.624	0.616	0.613
n=3	0.839	0.761	0.704	0.668	0.646	0.632	0.620	0.616	0.610
n=4	0.906	0.762	0.703	0.667	0.645	0.629	0.621	0.615	0.608
n=5	0.954	0.768	0.707	0.666	0.646	0.629	0.619	0.611	0.608
n=6	0.957	0.762	0.703	0.666	0.645	0.626	0.616	0.610	0.605
n=7	0.978	0.767	0.703	0.663	0.643	0.626	0.615	0.613	0.607
n=8	0.970	0.760	0.704	0.665	0.641	0.623	0.615	0.608	0.605
n=9	0.966	0.761	0.702	0.666	0.641	0.622	0.617	0.608	0.602
n=10	0.970	0.761	0.702	0.665	0.640	0.624	0.616	0.610	0.603

TABLE I

RMS FOR DIFFERENT n AND N

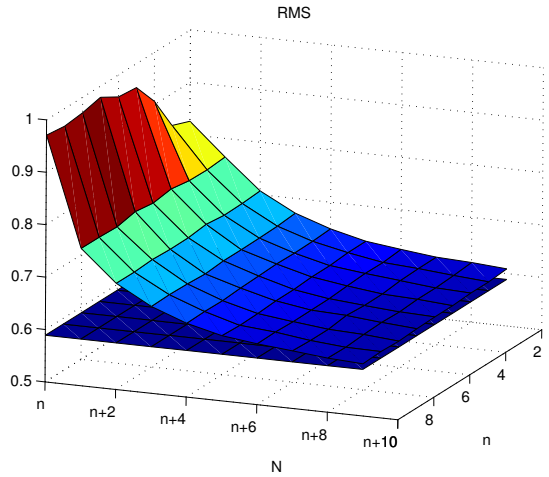

Fig. 9. 3D visualization of table I. Blue plane indicates RMS resulting from hierarchical reconstruction.

2) Camera location comparisons: In order to compare, computed trajectory is readjusted with reference sequence reconstructed with the hierarchical method. For that, a rigid transformation (rotation, translation, and scale factor) is ap-

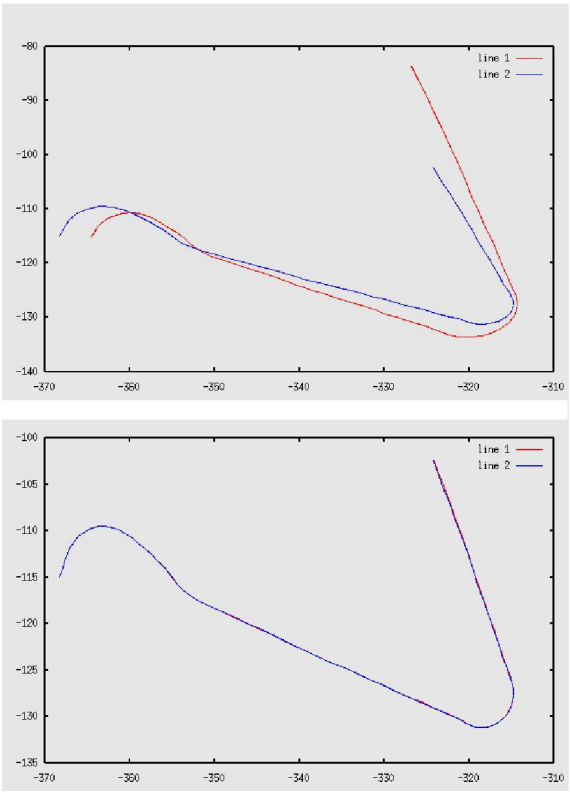

Fig. 10. 2 examples of readjustment with reference hierarchical reconstruction. a). top n=3, N=3 b). bottom n=3, N=10.

plied, so that they are in the same frame. This approach is described by Faugeras [3]. Because reference frame is graduated in meters, one can measure the average cameras position error (in meters) by $\frac{\sum_i \sqrt{E_{xi}^2 + E_{yi}^2 + E_{zi}^2}}{N_{seq}}$ where E_{xi}, E_{yi}, E_{zi} are position errors in directions x, y and z of camera i compared to the same camera in reference reconstruction. Results with different values of n and N are given in table II. We note that average error is lower than 13 cm if $N \geq n+3$ $\forall n$, and even lower than 11 cm with only 4 exceptions. This is a low relative error given displacements length. As for Figure 10, it shows first a very deformed reconstruction where readjustment suits not very well and then a good reconstruction.

n \ N	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8
n=2	7.966	4.581	0.112	0.085	0.114	0.105	0.075	0.050	0.045
n=3	2.783	0.957	0.200	0.054	0.073	0.093	0.104	0.050	0.035
n=4	1.600	0.637	0.124	0.051	0.055	0.109	0.039	0.044	0.029
n=5	1.157	0.271	0.099	0.064	0.046	0.041	0.069	0.081	0.071
n=6	0.942	0.170	0.112	0.082	0.032	0.074	0.102	0.055	0.073
n=7	0.803	0.088	0.131	0.055	0.036	0.088	0.066	0.071	0.070
n=8	0.754	0.189	0.177	0.127	0.027	0.076	0.063	0.117	0.114
n=9	0.659	0.148	0.057	0.097	0.051	0.026	0.097	0.065	0.093
n=10	0.563	0.085	0.124	0.093	0.028	0.058	0.068	0.037	0.027

TABLE II

MEAN ERROR (IN METERS) OF CAMERA POSITIONS COMPARED TO REFERENCE SEQUENCE.

3) Computation time comparison: Calculations have been done on a standard a Linux computer (Pentium 4 at 2.8 GHz, 1Go of RAM memory at 800 MHz). Our incremental method is actually 2-3 times faster than the hierarchical method. That responds to our waitings, more especially because bundle adjustment implementation can still be largely optimized. Results are reported in table III and are expressed as a percentage saving of time, compared to traditional reconstruction. During experiments, the computing times were measured in seconds and go from $325s$ to $508s$ against $919s$ for the hierarchical method.

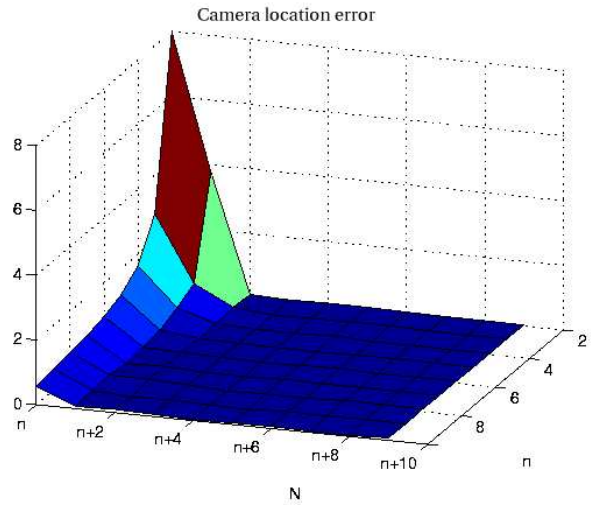

Fig. 11. 3D visualization of table II representing how varies camera location error with n and N.

n \ N	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8
n=2	64.5	63.1	63.6	63.3	62.7	62.5	63.0	62.7	62.8
n=3	64.5	62.0	60.8	61.0	60.4	61.2	61.5	60.8	60.5
n=4	61.0	58.9	58.8	58.9	60.3	58.4	57.7	58.2	59.0
n=5	58.9	57.6	57.0	57.1	56.1	56.4	56.7	57.1	57.4
n=6	56.7	55.4	55.9	54.3	53.9	55.2	54.5	54.7	53.6
n=7	52.6	53.0	52.1	52.1	52.4	52.0	53.2	51.7	51.6
n=8	51.4	50.6	49.4	50.7	49.5	50.0	50.3	49.0	50.4
n=9	49.0	47.5	47.8	47.3	48.9	48.0	46.8	46.6	47.2
n=10	45.4	44.9	44.2	46.1	45.5	44.6	45.6	44.7	47.3

TABLE III

PERCENTAGE OF SAVING TIME.

V. EVOLUTION OF THE METHOD: A STEP TO REAL TIME MOTION ESTIMATION AND 3D RECONSTRUCTION

In order to get closer to a real time reconstruction (i.e. the case where reconstruction proceeds in the same time as computer gets video frames) we have to accelerate the process. First, our most recent work permits an in-line feature detection and matching, what is crucial within our purpose.

As for the local bundle adjustment stage, iterations count is limited to 2×5 iterations. After a first series of iterations (maximum 5), computation includes a rejection of "outliers" i.e. 2D reprojections whose error is higher than 1.0 pixel. After that, a new series of iterations is carried out. Parameters n and N are fixed to 3 and 10. In this context, reconstruction associated to optimization does not exceed a time of $500ms/key\text{-}frame$ (and $40ms$ for the non-key-frames). Thanks to this recent improvement, we obtain a high quality 3D reconstruction of

very long urban scenes (see Figures 12 and 13) in a very low computing time. The presented example is a sequence acquired with a calibrated camera fixed on a car. The distance covered is more than one kilometer and the video is about 3min long. Calculating the reconstruction has taken 7min20 for 354 key frames and more than 16.000 3D points (image size is 512×384 pixels). Drift is very low compared to the covered distance.

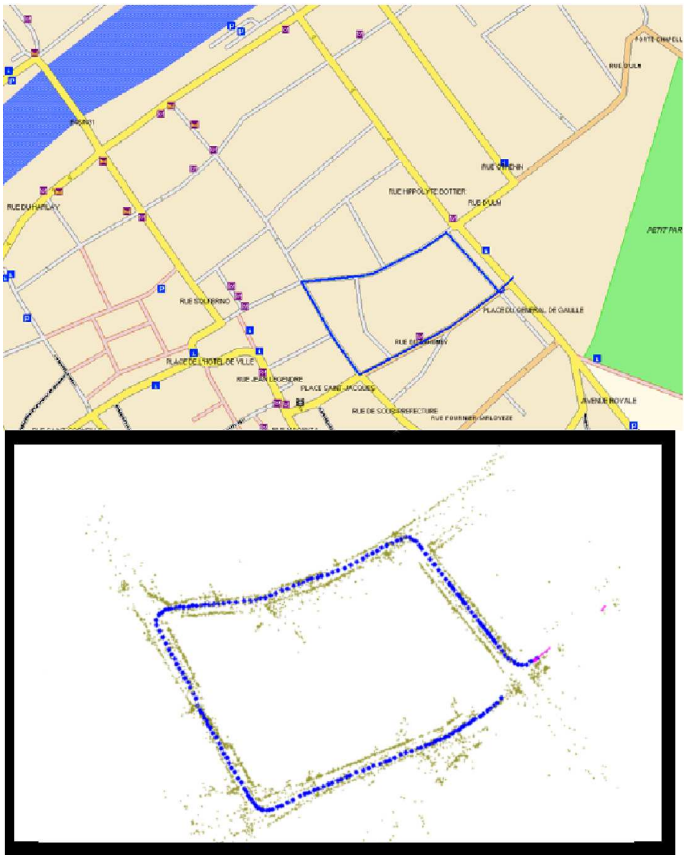

Fig. 12. "street" sequence a) top: a city map with the trajectory in blue b). bottom: reconstruction result (video: 3min, reconstruction: 7min20, 354 key frames, 16.135 3D points).

Fig. 13. 3 frames from "street" sequence.

VI. Conclusion

We presented a method that greatly accelerates long scenes 3D reconstruction process. Experiments showed that results obtained are very satisfactory and reliable. We have used our incremental method instead of the usual hierarchical method providing a 3D model in a robot's localization application. This application was successful, although asking for a real accuracy. This allowed us to validate the method. Now, we have two perspectives.

First of all, the results let us think that persevering in the optimization of the method implementation will make it possible to approach an accurate 3D reconstruction in real time. Indeed, for a video flow of 30 fps, a real time reconstruction based on a key frame every 10 frames takes a computation time lower than $10 \times 33ms$ (30 fps) = $330ms$. According to our recent experiments, very good results are obtained with a processing time of less than $500ms/key\text{-}frame$ (and $40ms$ for the non-key-frames) what is close to real time. The main expected improvement is a judicious choice of the number and distribution of points to reconstruct.

Second, with the aim of very long sequences reconstruction, it is possible to merge subsequences obtained with our reconstruction method. With a very similar approach to what is proposed in this paper, we think it would be possible to accelerate this fusion process, by reducing the number of optimized parameters to those which are close to the joint zone between the two subsequences.

References

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

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

[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.

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

[5] 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

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

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

[8] 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.

[9] D. Nister, "An efficient solution to the five-point relative pose problem," *CVPR03*, Vol. 2, pp. 195-202, 2003

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

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

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

[13] 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.

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

[15] 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.

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

[17] 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.

[18] Z. Zhang and Y. Shan. Incremental Motion Estimation through Modified Bundle Adjustment. *ICIP'03*.