Generic and Real Time Structure from Motion using Local Bundle Adjustment [1]

E. Mouragnon [a,b], M. Lhuillier [a,*], M. Dhome [a], F. Dekeyser [b], P. Sayd [b]

[a] *LASMEA UMR 6602, Université Blaise Pascal/CNRS, 63177 Aubière Cedex, France*

[b] *Image and embedded computer lab., CEA/LIST/DTSI/SARC, 91191 Gif-sur-Yvette Cedex, France*

Abstract

This paper describes a method for estimating the motion of a calibrated camera and the three dimensional geometry of the filmed environment. The only data used is video input. 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 to enable 3D reconstruction of the features. We introduce a local bundle adjustment allowing 3D points and camera poses to be refined simultaneously through the sequence. This significantly reduces computational complexity when compared with global bundle adjustment. This method is applied initially to a perspective camera model, then extended to a generic camera model to describe most existing kinds of cameras. Experiments performed using real world data provide evaluations of the speed and robustness of the method. Results are compared to the ground truth measured with a differential GPS. The generalized method is also evaluated experimentally, using three types of calibrated cameras: stereo rig, perspective and catadioptric.

Key words: Structure from Motion, Bundle Adjustment, Generic camera

* Corresponding author
Email address: Maxime.Lhuillier@univ-bpclermont.fr (M. Lhuillier).
URL: maxime.lhuillier.free.fr (M. Lhuillier).

[1] Expanded version of a paper published at the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition, New-York, 2006 (CVPR'06), and a paper published at the British Machine Vision Conference, Warwick, 2007 (BMVC'07).

1 Introduction

1.1 Previous Work

Automatic estimation of 3D scene structure and camera motion from an image sequence (known as "Structure from Motion" or SfM) has been the subject of much investigation. Different camera models - pinhole, fish-eye, stereo, catadioptric, multicamera systems, etc - have been used in such studies. This is still a very active field of research, and several successful SfM systems currently exist [20,1,18,11,23].

Calculation Time vs. Accuracy A large number of dedicated algorithms (i.e. for given camera models) have been successfully developed and are now commonly used for perspective or stereo rig models [18,20].
There are several noteworthy types of dedicated SfM algorithms. These include firstly approaches without global optimization of the full geometry which are fast but of questionable accuracy (due to errors that accumulate over time). Among the work proposed for Vision-Based SLAM (Simultaneous Localization and Mapping), Nistér *et al.*[18] have presented an approach known as "visual odometry". This method estimates the motion of a stereo head or a simple camera in real-time, using visual data only: its aim is to provide guidance for robots. Davison [3] proposes a real-time camera pose calculation method but assumes a small number of landmarks only (less than 100 landmarks). This approach is best suited to indoor environments and is not therefore appropriate for lengthy camera displacements due to the complexity of its algorithms and progressively increasing uncertainties.
There are also different off-line methods that use bundle adjustment to optimize global geometry and thus obtain highly accurate models (see [27] for a thorough survey of bundle adjustment algorithms). Such an optimization is very costly in terms of computing time and can not be implemented in a real-time application. Bundle adjustment entails iterative adjustment for camera poses and point positions in order to obtain the optimal least squares solution. Most of these articles call on the Levenberg-Marquardt (LM) algorithm, which combines the Gauss-Newton algorithm with the method of gradient descent to solve the non linear criterion involved in bundle adjustment. The main problem in such 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 the sparse structure of the systems involved is taken into account).
It is also important to have an initial estimate relatively close to the real solution. Applying a bundle adjustment in a hierarchical way is therefore an interesting idea [9,24,23] but does not solve the computing time problem. An alternative method is therefore necessary to decrease the number of parame-

ters to be optimized. Shum *et al.*[24] exploit information redundancy in images by using two virtual key frames to represent a sequence. Steedly and Essa [25] propose an incremental reconstruction with bundle adjustment. They then readjust only the parameters which have changed. While their method is faster than a global optimization, it is not efficient enough for long sequences that are highly data-dependent. Kalman filters or extended Kalman filters [3] are another possibility, but such filters are known to provide less accurate results than bundle adjustment.

Generic vs. Specific Structure from Motion Yet another widely explored avenue is that of omni-directional central (catadioptric, fish-eye) or non-central (e.g. multicamera) systems that offer a larger field of view [2,14,19]. It is highly challenging to develop generic SfM tools suitable for any camera model. This approach has recently been investigated using generic camera models [10,22]. In such models, pixels define image rays in the camera's coordinate system. Rays corresponding to pixels are given by the calibration function. They intersect at a single point usually called "projection center" in the case of a central camera [10] but this is not necessary in other cases. In recent work on generic SfM, camera motion is estimated by the generalization of the conventional essential matrix [22,19,16] derived from the Pless Equation [19], and minimal relative pose estimation algorithms [26].
In the same way as for the specific models, a method is also required for refinement of 3D points and camera poses. In the generic case which implies different cameras (pinhole, stereo, fish-eye), bundle adjustment is different from the conventional approach used for perspective cameras. The minimized error may be a 3D or a 2D error. As the projection function is not explicit for some camera models, a 3D error can be used [22,14]. A 3D error is not however optimal since it favors the contribution of far points from the cameras and can produce biased results [13]. An other solution is to minimize a 2D reprojection error (in pixels) by clustering all camera rays such that each cluster of rays is approximated by a perspective camera [22].

Summary and comparison with previous work This paper first proposes an accurate and fast incremental reconstruction and localization algorithm. Our idea [15] is to take advantage of both offline methods with bundle adjustment and faster incremental methods. In our algorithm, a local bundle adjustment is carried out each time a new camera pose is added to the system. A similar approach [4] was also published a few months after ours [15], but it is not generic and it does not include a key-frame selection to stabilize 3D calculation as ours. A related approach is proposed by Zhang and Shan [28], but their work calls for local optimization of an image triplet only, and eliminates structure parameters from the proposed reduced local bundle adjustment. By taking into account 2D reprojections of 3D estimated points in more than three images without eliminating 3D point parameters, it is possible to greatly improve reconstruction accuracy.

Secondly, this paper shows how our fast reconstruction method is extended to generic cameras. Within this generalization framework, our method [16] replaces image projections of a specific camera model with use of back-projected rays and minimization of angular error between rays. Its first advantage is, of course, a high degree of interchangeability between camera models. The second advantage is that it is also effective where the image projection function is not explicit (as in the case of non-central catadioptric cameras) and precludes clustering.

In summary, previously proposed approaches related to ours include the following:
- generic but non-real-time methods [22,10,12] ([10] deals only with central cameras).
- real-time but non-generic methods [3,18], not using bundle adjustment.
- generic methods using the Pless Equation [19,22] (generalization of the epipolar constraint), but without details on how to solve the equation in common situations.
- dedicated or "specific" (non-generic) methods using local bundle adjustment [15,4].

1.2 Our Contribution

The first new feature of our approach is use of local bundle adjustment, for both specific model (perspective) and generic model cameras. The second is inclusion of a detailed method for solving the Pless equation (in most cases, this is not a "simple" linear problem as suggested in [19,22]). Our original study of artificial solutions of the linearized version of Pless equation is a by-product. Finally, the system as a whole is new (in that it is the first to be both real-time and generic).

For purposes of clarity, this paper first describes the complete method (including local bundle adjustment) for a standard (perspective) camera in Section 2. This Section also compares the time complexity of our local bundle adjustment in the incremental scheme to that of the standard (global) bundle adjustment in the hierarchical scheme. Sections 3 and 4 deal with the generalization of the method, by describing the generic camera model and explaining modifications to geometry refinement. This is followed by a more detailed look at the generic initialization step and various solutions to the Pless equation. Finally, experiments performed for perspective and generic camera models are presented in Sections 5 and 6 respectively.

2 Incremental Method for the Perspective Camera Model

Let us consider a video sequence acquired with a camera mounted on a vehicle moving through an unknown environment. The purpose of our work is to enable determination of camera position and orientation in a global reference frame at several points in time t, along with the 3D position of a set of points (viewed along the scene). To do so, we use a monocular camera whose intrinsic parameters (including radial distortion) are known and assumed to be unchanged throughout the sequence. The algorithm initially determines a first image triplet for use in setting up the global frame and system geometry (Section 2.2). A robust pose calculation is then carried out for each frame of the video flow (Section 2.3) using feature detection and matching (Section 2.1). Some of the frames are selected as key frames for 3D point triangulation (Section 2.4). The system operates "incrementally", and when a new key frame and 3D points are added, it performs local bundle adjustment (Section 2.5). The result (see Figure 4) is a set of camera poses corresponding to key frames and 3D coordinates of the points seen in the images.

2.1 Interest Point Detection and Matching

Our entire method is based on the detection and matching of feature points (see Figure 1). In each frame, Harris corners [8] are detected. A pair of frames is matched as follows:

- For each interest point in *image* 1, we select possible corresponding points in a region of interest defined in *image* 2.
- Then a Zero Normalized Cross Correlation score is computed for these interest points neighborhoods.
- The pairs with the highest scores are selected to provide a list of corresponding point pairs between the two images. A unicity constraint (winner takes all) is applied such that a point can only be matched with one other point.

For adaptation to a real-time application, the "detection and matching" step has been implemented using SIMD extensions of modern processors. This is both a fast and highly efficient solution.

2.2 Sequence Initialization With 3 Views

We know that the motion between two consecutive frames must be sufficiently large to compute the epipolar geometry and 3D points. For this reason, frames

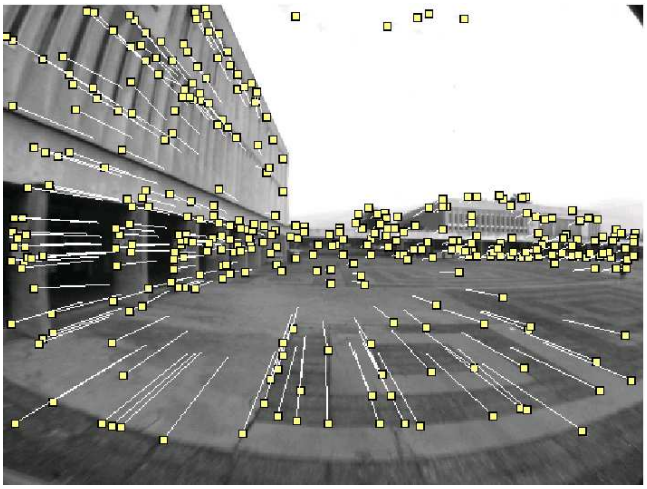

Fig. 1. Example of an image from video data. Small squares represent detected interest points, and white lines the distance covered by the matched points.

are selected relatively far from each other but with a suitable number of common points. To do so, the first image denoted 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 interest points matched in I^1. Then for I^3, the frame selected is the one farthest 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 (for our experiments, the values $M = 400$ and $M' = 300$ were selected). This process affords a sufficient number of corresponding points between frames to calculate the movement of the camera. The camera coordinate system associated with I^1 is taken as the global coordinate system, and the relative poses between the first three key frames are calculated using the 5-point algorithm [17] and a RANSAC [6] approach. Use of three views enhances the robustness of the method and eliminates ambiguities induced by coplanar points [17]. More details on the initialization process are given in [23]. Observed points are subsequently triangulated into 3D points using the first and the third observation. Finally estimated poses and 3D point coordinates are optimized through standard bundle adjustment

2.3 Robust Real-Time Pose Estimates

Let's assume that poses obtained with cameras C^1 to C^{i-1} and corresponding to selected key frames I^1 to I^{i-1} were previously calculated in the reconstruction reference frame. A set of points were computed whose projections were present in the corresponding images. The next step is to calculate camera pose C corresponding to the last acquired frame I. To do so, 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 co-

ordinates have already been computed. Taking 3D points reconstructed from C^{i-2} and C^{i-1}, we use Grunert's pose estimation algorithm as described in [7] to compute the location of camera C. A RANSAC process then gives an initial estimate of camera C pose which is subsequently 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 the camera pose C is calculated via the Hessian inverse and we can draw a 90% confidence ellipsoid (see Figure 2). If Cov is the covariance matrix of camera pose C, the ellipsoid of confidence is given by $\Delta x^T Cov^{-1} \Delta x \leq 6.25$ since $\Delta x^T Cov^{-1} \Delta x$ obeys the $\mathcal{X}_2$ distribution with 3 DOFs.

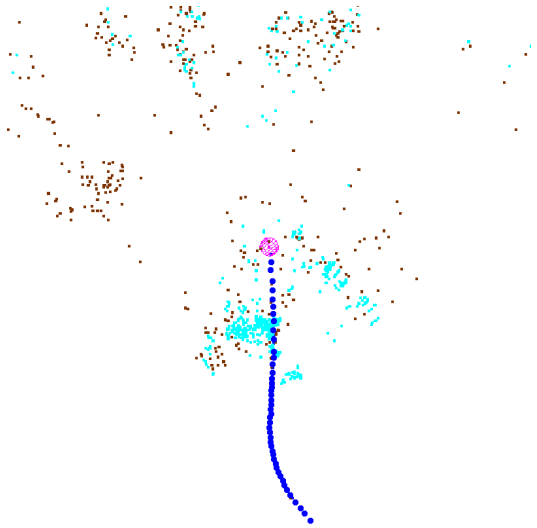

Fig. 2. Top view of a processing reconstruction. This shows the trajectory, reconstructed 3D points and a confidence ellipsoid for the most recently calculated camera.

2.4 Key Frame Selection and 3D Point Reconstruction

As already seen above, not all frames of the input are taken into account for 3D reconstruction: only a sub-sample of the video is used (Figure 3). For each frame, the normal approach is to compute the corresponding localization from the last two key frames. In our case, a criterion is incorporated to indicate whether or not a new frame must be added as a key frame. First, if the number of points matched with the last key frame I^{i-1} is not sufficient (typically, below a fixed level M, ($M = 400$ with in our experiments), a new key frame is required. This is also necessary if the calculated position uncertainty exceeds a certain level (for example, more than the mean distance between two consecutive, key positions). Obviously, the frame that did not meet the criterion cannot become a new key frame I^i and the immediately preceding frame is therefore taken. New points (i.e. those only observed in I^{i-2}, I^{i-1} and I^i) are reconstructed using a standard triangulation method.

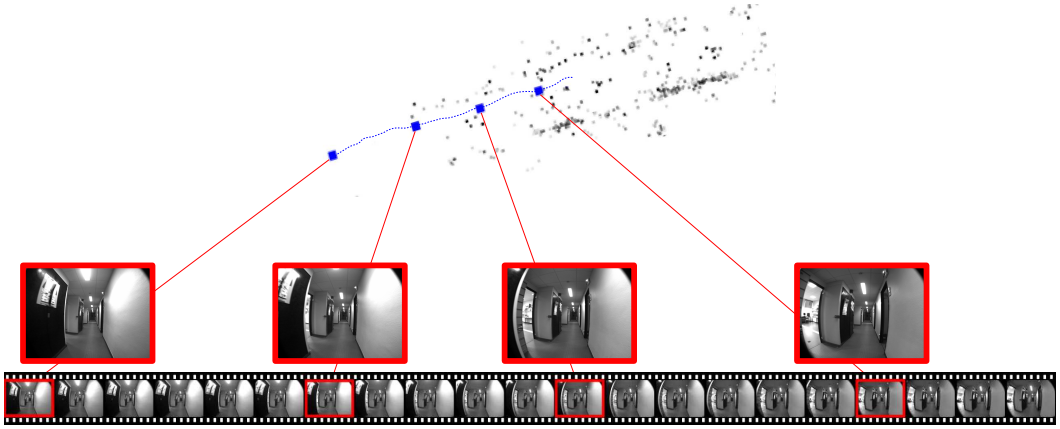

Fig. 3. Video sub-sampling: localization of all frames. 3D point reconstruction with key frames (squares).

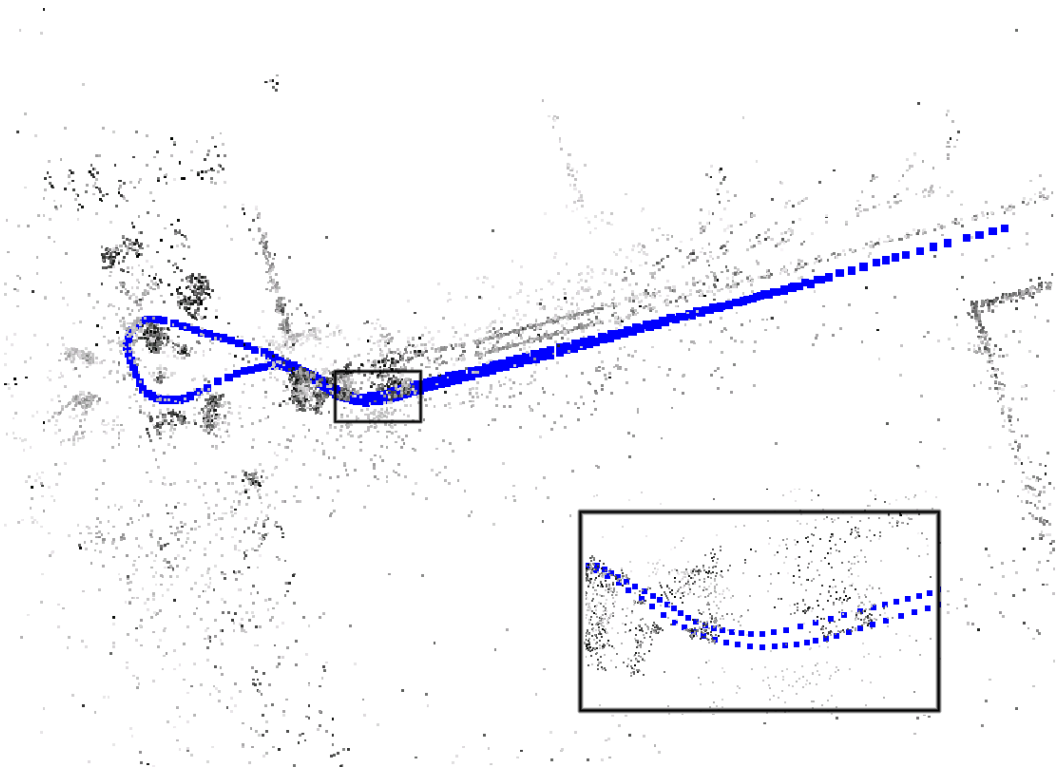

Fig. 4. Top view of a complete reconstruction in an urban environment. The distance covered is a length of about 200 meters including a half-turn. More than 8.000 3D points are reconstructed for 240 key frames.

2.5 Local Bundle Adjustment

Following selection of the last key frame I^i and its addition to the others, the reconstruction is optimized. The optimization process 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 parameters of the cameras (extrinsic parameters) and the 3D points chosen for this stage i. The idea is to reduce the number of calculated parameters by optimizing only the extrinsic parameters of the n last cameras and accounting for the 2D reprojections in the N (with $N \geq n$) last frames (see Figure 5). Thus, $\mathcal{C}^i = \{C^{i-n+1} \ \dots \ C^i\}$ and $\mathcal{P}^i$ contains all the 3D points projected on cameras $\mathcal{C}^i$. The 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:

$$f^i(\mathcal{C}^i, \mathcal{P}^i) = \sum_{C^k \in \{C^{i-N+1} \ \dots \ C^i\}} \sum_{\mathbf{P}_j \in \mathcal{P}^i} ||\varepsilon_j^k||^2$$

where $||\varepsilon_j^k||^2 = d^2(\mathbf{p}_j^k, \mathtt{K}^k\mathbf{P}_j)$ is the square of the Euclidean distance between $\mathtt{K}^k\mathbf{P}_j$, estimated projection of 3D point $\mathbf{P}_j$ through the camera C^k and the measured corresponding observation $\mathbf{p}_j^k$. $\mathtt{K}^k$ is the projection matrix 3×4 of camera k comprising C^k 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 two main parameters involved in the optimization process. Their given value can influence both quality of results and speed of execution. Our experiments enabled the determination of values for n and N (typically, we use $n = 3$ and $N = 10$) that provide an accurate reconstruction. Optimization takes place in two LM stages with an update of inliers/outliers between the two stages. A series of LM iterations is stopped if the error is not suitably decreased or a maximum number of iterations is reached (5 in our case). In practice, the number of necessary iterations for each local bundle adjustment is quite low; this is due to the fact that, excepting the last added camera, all the cameras poses have already been optimized at stage $i-1$, $i-2$, ...

It should be noted 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 opted for $N = n = i$. N_f is the maximum number of cameras required for stage i optimization to be global (in our experiments, we choose $N_f = 20$). In this way, reliable initial data is obtained, an important factor given the recursive nature of the algorithm, without any problems, since the number of parameters is still relatively restricted at this stage.

2.6 Time Complexities for Local and Global Bundle Adjustments

Detailed calculation complexity calculations are given in the Appendix A of this paper. Let p be the number (considered as constant) of points projected through each camera. Based on a video sequence of N_{seq} key frames, the

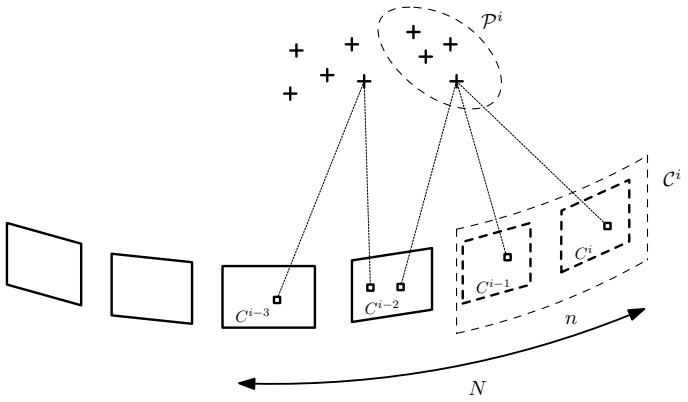

Fig. 5. Local bundle adjustment when camera C^i is added. Only the points and cameras surrounded by dotted lines are optimized. We nevertheless account for 3D point reprojections in the N last images.

complexity of one local bundle adjustment iteration applied to the n last cameras after allowance for 2D reprojections on N images is

$$\Theta(p.N + p.n^2 + n^3)$$

whereas the complexity of one global bundle adjustment iteration applied to the whole sequence is

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

It is clearly advantageous to reduce the number of parameters (n and N) involved in the optimization process. As an example, the complexity gains with respect to global bundle adjustment obtained for a sequence of 20 key frames and 150 2D reprojections per image are shown in Table 3.

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

Table 1
Complexity gain obtained through local bundle adjustment in comparison with global bundle adjustment for one iteration.

This study deals with only one LM iteration. In practice, parameters n and N are fixed with low values ($n = 3$ and $N \leq 10$) for our incremental method. Furthermore, the number of iterations is limited by 10 for each local bundle adjustment (although we note that the mean number of necessary iterations is less than that). As a consequence, the time complexity of local bundle adjustment is $O(1)$ for each key frame and is $O(N_{seq})$ in our incremental scheme for a complete sequence.

This should be compared to the complexity of global bundle adjustment in the hierarchical scheme [9]. At first glance, one iteration of the last bundle adjustment step over the whole sequence is greater than N_{seq}^3. So it is easy to say that the time complexity is the best for the incremental scheme. A more valuable comparison could be done if we assume that the track length is bounded by a constant l. In that case, the complexity N_{seq}^3 due to solve the reduced camera system of global bundle is reduced to $N_{seq}l^2$ [27]. So the complexity of one iteration of global bundle is at least proportional to the sequence length. Since the global bundle adjustment is applied in a hierarchical scheme, we apply it on 1 sequence of length N_{seq}, 2 sequences of length $N_{seq}/2$, 4 sequences of length $N_{seq}/4$ ... (in the reverse order). Thus the overall complexity is greater than $N_{seq}log(N_{seq})$. One time again, the time complexity is the best for the incremental scheme.

2.7 Method Summary

In summary, the proposed method consists of the following steps:

(1) Select an image triplet that provides the first three key frames (Section 2.2). Set up the global frame, estimate the relative pose, and triangulate 3D points.
(2) For each new frame, calculate matches with last key frame (Section 2.1) and estimate the camera pose and uncertainty (Section 2.3). Determine whether a new key frame is needed. If not, repeat 2.
(3) If a new key frame is necessary, select the preceding frame as new key frame, triangulate new points (Section 2.4) and make a local bundle adjustment (Section 2.5). Repeat the above starting from step 2.

3 Generic Camera Model and Geometry Refinement

The method presented in Section 2 is designed for a perspective camera model. Our approach, however, is geared to also using other kinds of cameras for 3D reconstruction. Dedicated methods are possible for each (e.g. catadioptric camera, stereo rig) but a method suitable for any type of camera involved would be very valuable. Our approach thus entails extending the incremental method to a generic camera model [16]. In the following section (Section 3), the pixel error applied to refine geometry is replaced by a generic error. Robust initialization with three views is then described in detail in Section 4.

3.1 Camera Model

For any pixel $\mathbf{p}$ of a generic image, the (known) calibration function f of the camera defines an optical ray $r = f(\mathbf{p})$. This projection ray is an oriented line $r = (\mathbf{s}, \mathbf{d})$ for which $\mathbf{s}$ is the starting point or origin and $\mathbf{d}$ is the direction of the ray in the camera frame ($||\mathbf{d}|| = 1$). For a central camera, $\mathbf{s}$ is a single point (camera center) whatever the pixel $\mathbf{p}$. In the general case, $\mathbf{s}$ could be any point given by the calibration function.

3.2 Error Choice

Let $\mathbf{P}_j = [x_j, y_j, z_j, t_j]^\top$ be the homogeneous coordinates of the j-th point in the world frame. Let $\mathtt{R}^i$ and $\mathbf{t}^i$ be the orientation (rotation matrix) and the origin of the i-th camera frame in the world frame.

If $(\mathbf{s}^i_j, \mathbf{d}^i_j)$ is the optical ray corresponding to the observation of $\mathbf{P}_j$ through the i-th camera, the direction of the line defined by $\mathbf{s}^i_j$ and $\mathbf{P}_j$ is $\mathbf{D}^i_j = {\mathtt{R}^i}^\top[\mathtt{I}_3 \mid -\mathbf{t}^i]\mathbf{P}_j - t_j\mathbf{s}^i_j$ in the i-th camera frame. In the ideal case, directions $\mathbf{d}^i_j$ and $\mathbf{D}^i_j$ are parallel (which is equivalent to an image reprojection error of zero pixels).

The conventional approach [27,15] consists of minimizing a sum of squares $||\epsilon^i_j||^2$ where ϵ^i_j is a specific error depending on the camera model: the reprojection error in pixels. In our case, a generic error must be minimized. We thus define ϵ^i_j as the angle between the directions $\mathbf{d}^i_j$ and $\mathbf{D}^i_j$ defined above (see Figure 6).

Many experiments show that the convergence of bundle adjustment is very poor with $\epsilon^i_j = \arccos(\mathbf{d}^i_j.\frac{\mathbf{D}^i_j}{||\mathbf{D}^i_j||})$ and satisfactory with ϵ^i_j defined as follows [12]. We have thus chosen $\epsilon^i_j = \pi(\mathtt{R}^\mathtt{i}_\mathtt{j}\mathbf{D}^i_j)$ with $\mathtt{R}^\mathtt{i}_\mathtt{j}$ a rotation matrix such that $\mathtt{R}^\mathtt{i}_\mathtt{j}\mathbf{d}^i_j = [0\ 0\ 1]^\top$ and π a function $\mathbb{R}^3 \to \mathbb{R}^2$ such that $\pi([x\ y\ z]^\top) = [\frac{x}{z}\ \frac{y}{z}]^\top$. Note that ϵ^i_j is a 2D vector whose Euclidean norm $||\epsilon^i_j||$ is equal to the tangent of the angle between $\mathbf{d}^i_j$ and $\mathbf{D}^i_j$. The tangent is a good approximation of the angle if it is small.

3.3 Intersection, Resection and Bundle Adjustment

Once ϵ^i_j redefined, it is a straightforward matter to redefine the common tools for (incremental) reconstruction. Calculation of a 3D point coordinates $\mathbf{P}_j$ knowing its observations in cameras $C^1 \ \ldots \ C^n$ (intersection) is given by min-

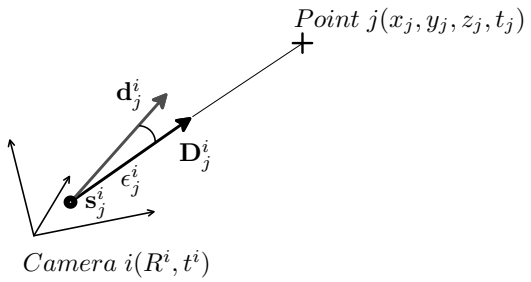

Fig. 6. Angular bundle adjustment: the angle between observation ray $(\mathbf{s}_j^i, \mathbf{d}_j^i)$ and 3D ray $\mathbf{D}_j^i$ which goes from $\mathbf{s}_j^i$ to 3D point is minimized.

imizing g_j:

$$g_j(\mathbf{P}_j) = \sum_{i=1...n} ||\epsilon_j^i||^2.$$

Calculation of camera pose C^i knowing 3D points $\mathbf{P}_1 \ \ldots \ \mathbf{P}_m$ (resection) is given by minimizing g^i:

$$g^i(C^i) = \sum_{j=1...m} ||\epsilon_j^i||^2.$$

In the same way as for the "specific" method, a local bundle adjustment is applied each time a new key frame I^i is added to the reconstruction. Parameters of estimated 3D points and cameras at the end of the sequence are refined by the minimization of the cost function g^i:

$$g^i(\mathcal{C}^i, \mathcal{P}^i) = \sum_{C^k \in \{C^{i-N+1} \ \ldots \ C^i\}} \sum_{\mathbf{P}_j \in \mathcal{P}^i} ||\epsilon_j^k||^2.$$

where $\mathcal{C}^i$ and $\mathcal{P}^i$ are respectively the generic camera parameters (extrinsic parameters of key frames) and the 3D points chosen for stage i (Figure 7). We account for points reprojections in the N (with $N \geq n$) last frames (typically $n = 3$ and $N = 10$).

4 Generic Initialization

The following paragraphs describe in detail the generic initialization of incremental 3D reconstruction. Section 4.1 briefly presents the Plücker coordinates used to describe 3D rays in space. The generic epipolar constraint (or Pless Equation) is also described in Section 4.2, and Section 4.3 shows how to solve it in different cases. Finally robust initialization with three views and robust pose estimation are explained in Sections 4.4 and 4.5 respectively.

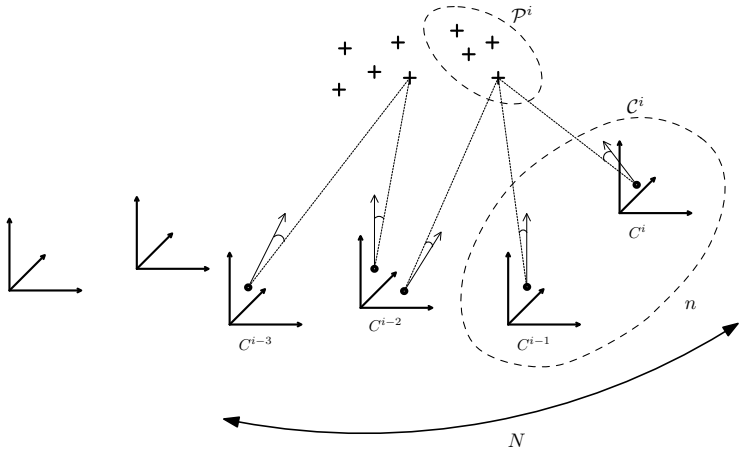

Fig. 7. Local angular bundle adjustment when camera C^i is added. Only points $\mathcal{P}^i$ and cameras $\mathcal{C}^i$ parameters surrounded by dotted lines are optimized. The minimized criterion nevertheless accounts for 3D points projections in the N last images.

4.1 Plücker Coordinates

Based on a set of pixel correspondences between two images, the relative pose $(\mathtt{R}, \mathbf{t})$ of two cameras can be estimated in a generic framework. For each 2D points correspondence (x_0, y_0) and (x_1, y_1) between images 0 and 1, we have a correspondence of optical rays $(\mathbf{s}_0, \mathbf{d}_0)$ and $(\mathbf{s}_1, \mathbf{d}_1)$. A ray $(\mathbf{s}, \mathbf{d})$ is defined by its Plücker coordinates, which are convenient for this calculation. The Plücker coordinates of a 3D line (or ray) are two 3×1 vectors $(\mathbf{q}, \mathbf{q}')$ respectively named direction vector and moment vector. $\mathbf{q}$ gives the direction of the line and $\mathbf{q}'$ is such that $\mathbf{q} \cdot \mathbf{q}' = 0$. Any point $\mathbf{P}$ on the line described by $(\mathbf{q}, \mathbf{q}')$ satisfies $\mathbf{q}' = \mathbf{q} \wedge \mathbf{P}$. Plücker coordinates are defined up to a scale and a parametrization of the line is given by $(\mathbf{q}' \wedge \mathbf{q}) + \alpha \mathbf{q},\ \forall \alpha \in \mathbb{R}$ if $||\mathbf{q}|| = 1$.

For an optical ray $(\mathbf{s}, \mathbf{d})$ as previously defined, where $\mathbf{s}$ is the origin of the ray and $\mathbf{d}$ its direction in the camera frame, Plücker coordinates in the same frame are: $\begin{cases} \mathbf{q} = \mathbf{d} \\ \mathbf{q}' = \mathbf{d} \wedge \mathbf{s} \end{cases}$

4.2 Generalized Epipolar Constraint (or Pless Equation)

Let camera 0 be the origin of the global coordinates system. Its pose is written as $\mathtt{R}_0 = \mathtt{I}_3$ and $\mathbf{t}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$. We want to determine $(\mathtt{R}, \mathbf{t})$ which is the pose of camera 1 in the same frame. For each pixel correspondence between image 0 and image 1, optical ray $(\mathbf{q}_0, \mathbf{q}'_0)$ and $(\mathbf{q}_1, \mathbf{q}'_1)$ must intersect in space at a single 3D point $\mathbf{M}$ (see Figure 8).

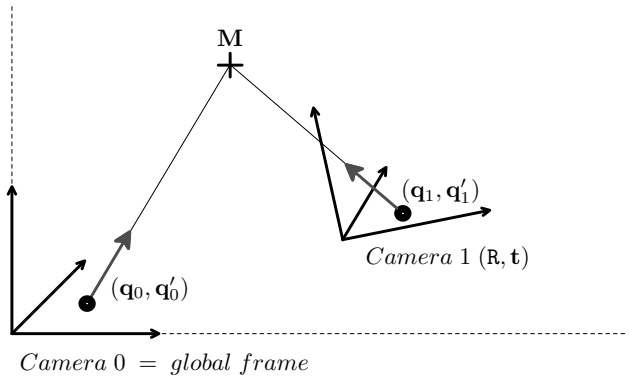

Fig. 8. Relative pose of 2 cameras

In the global frame, the two rays are:

$$\begin{cases} \mathbf{q}_{00} = \mathbf{q}_0 \\ \mathbf{q}'_{00} = \mathbf{q}'_0 \end{cases} \text{ and } \begin{cases} \mathbf{q}_{10} = \mathtt{R}\mathbf{q}_1 \\ \mathbf{q}'_{10} = -[\mathbf{t}]_\times \mathtt{R}\mathbf{q}_1 + \mathtt{R}\mathbf{q}'_1 \end{cases}$$

where $[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$ and $\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top$.

$[\mathbf{t}]_\times$ is the skew-symmetric cross-product matrix of $\mathbf{t}$ such that $[\mathbf{t}]_\times \mathbf{x} = \mathbf{t} \wedge \mathbf{x}$ for a 3×1 vector $\mathbf{x}$.

The two rays intersect if and only if $\mathbf{q}_{10} \cdot \mathbf{q}'_{00} + \mathbf{q}'_{10} \cdot \mathbf{q}_{00} = 0$, which leads to the generalized epipolar constraint or Pless Equation [19]:

$$\mathbf{q}_0'^{\top} \mathtt{R} \mathbf{q}_1 - \mathbf{q}_0^\top [\mathbf{t}]_\times \mathtt{R} \mathbf{q}_1 + \mathbf{q}_0^\top \mathtt{R} \mathbf{q}'_1 = 0 \tag{1}$$

The generalized essential matrix $\mathtt{G}$ is defined as follows [22]:

$$\mathtt{G} = \begin{pmatrix} -[\mathbf{t}]_\times \mathtt{R} & \mathtt{R} \\ \mathtt{R} & \mathtt{0}_{3\times 3} \end{pmatrix}$$

The matrix $\mathtt{G}$ is a 6×6 matrix verifying the constraint: $\begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}'_1 \end{pmatrix}^\top \mathtt{G} \begin{pmatrix} \mathbf{q}_0 \\ \mathbf{q}'_0 \end{pmatrix} = 0$ if and only if rays $(\mathbf{q}_0, \mathbf{q}'_0)$ and $(\mathbf{q}_1, \mathbf{q}'_1)$ intersect in space. $\mathtt{G}$ contains 9 zero coefficients and the 9 coefficients of $\mathtt{R}$ twice. There are thus 18 useful coefficients.

We identify two cases where this equation has an infinite number of solutions. Obviously, this number is infinite if the camera is central (the 3D is recovered

up to a scale). We note that Equation 1 is the usual epipolar constraint defined by the essential matrix $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$ if the camera center is at the origin of the camera frame.

The second case is less obvious but it occurs in practice. In our experiments, we assume there are only **simple matches**: all projection rays $(\mathbf{s}^i, \mathbf{d}^i)$ of a given 3D point go through a same camera center (in the local coordinate of the generic camera). In other words, we have $\mathbf{q}'_0 = \mathbf{q}_0 \wedge \mathbf{c}^0$ and $\mathbf{q}'_1 = \mathbf{q}_1 \wedge \mathbf{c}^1$ with $\mathbf{c}^0 = \mathbf{c}^1$. For a multicamera system comprising central cameras (e.g. the stereo rig), this means that 2D points correspondences are only made with points of the same sub-image. This is often the case in practice due to the small regions of interest used for reliable matching, or the empty intersections between the fields of views of compositing cameras. If the camera motion is a pure translation ($\mathtt{R} = \mathtt{I}_3$), Equation 1 becomes $\mathbf{q}_0^\top [\mathbf{t}]_\times \mathbf{q}_1 = {\mathbf{q}'_0}^\top \mathbf{q}_1 + \mathbf{q}_0^\top \mathbf{q}'_1 = 0$ where the unknown is $\mathbf{t}$. In this context, the scale of $\mathbf{t}$ can not be estimated. We assume for our purposes that the camera motion is not a pure translation at the initialization stage.

4.3 Solving the Pless Equation

Equation 1 is rewritten as

$$ {\mathbf{q}'_0}^\top \tilde{\mathtt{R}} \mathbf{q}_1 - \mathbf{q}_0^\top \tilde{\mathtt{E}} \mathbf{q}_1 + \mathbf{q}_0^\top \tilde{\mathtt{R}} \mathbf{q}'_1 = 0 \tag{2} $$

where the two 3×3 matrices $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ are the new unknowns. We store the coefficients of $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ in an 18×1 vector $\mathbf{x}$ and see that each value of the 4-tuple $(\mathbf{q}_0, \mathbf{q}'_0, \mathbf{q}_1, \mathbf{q}'_1)$ produces a linear equation $\mathbf{a}^\top \mathbf{x} = 0$. If we have 17 different values of this 4-tuple for each correspondence k, we have 17 equations $\mathbf{a}_k^\top \mathbf{x} = 0$. This is enough to determine $\mathbf{x}$ up to a scale factor [22]. We have built the matrix $\mathtt{A}_{17}$ containing the 17 correspondences such that $\|\mathtt{A}_{17}\mathbf{x}\| = 0$ with $\mathtt{A}_{17}^\top = [\mathbf{a}_1^\top | \mathbf{a}_2^\top | \cdots \mathbf{a}_{17}^\top]$. The resolution depends on the dimension of the $\mathtt{A}_{17}$ kernel which directly depends on the type of camera used. We determine $Ker(\mathtt{A}_{17})$ and its dimension by a Singular Value Decomposition of $\mathtt{A}_{17}$. In this paper, we have distinguished three cases: (1) central cameras with an unique optical center (2) axial cameras with collinear centers and (3) non-axial cameras.

It is not surprising that the kernel dimension of the linear system to solve is greater than one. Indeed, the linear Equation 2 has more unknowns (18 unknowns) than the non-linear Equation 1 (6 unknowns). Possible dimensions are reported in Table 2 and are justified below. Previous works [19,22] ignored these dimensions, although a (linear) method is heavily dependent on them.

Camera	Central	Axial	Non-Axial
$dim(Ker(\mathtt{A}_{17}))$	10	4	2

Table 2
$dim(Ker(\mathtt{A}_{17}))$ depends on the kind of camera.

Central Camera For central cameras (e.g. pinhole cameras), all optical rays converge at the optical center $\mathbf{c}$. Since $\mathbf{q}'_i = \mathbf{q}_i \wedge \mathbf{c} = [-\mathbf{c}]_\times \mathbf{q}_i$, Equation 2 becomes ${\mathbf{q}_0}^\top([\mathbf{c}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{E}} - \tilde{\mathtt{R}}[\mathbf{c}]_\times)\mathbf{q}_1 = 0$. We note that $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}}) = (\tilde{\mathtt{R}}, [\mathbf{c}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}]_\times)$ is a possible solution of equation 2 for any 3×3 matrix $\tilde{\mathtt{R}}$. Such solutions are "exact": Equation 2 is exactly equal to 0 whatever $(\mathbf{q}_0, \mathbf{q}_1)$. Our "real" solution is $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}}) = (\mathtt{0}, [\mathbf{t}]_\times \mathtt{R})$ if $\mathbf{c} = 0$, and it is not exact due to image noise. Thus the dimension of $Ker(\mathtt{A}_{17})$ is at least $9+1$. Experiments have confirmed that this dimension is 10 (up to noise). In this case, we simply solve the usual epipolar constraint constraint ${\mathbf{q}_0}^\top [\mathbf{t}]_\times \mathtt{R} \mathbf{q}_1 = 0$ as described in [9].

Axial Camera This case includes the common stereo rig of two perspective cameras. Let $\mathbf{c}_a$ and $\mathbf{c}_b$ be two different centers of the camera axis. Appendix B shows that "exact" solutions $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ are defined by

$$\tilde{\mathtt{E}} = [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times \text{ and } \tilde{\mathtt{R}} \in Vect\{\mathtt{I}_{3\times 3}, [\mathbf{c}_a - \mathbf{c}_b]_\times, (\mathbf{c}_a - \mathbf{c}_b)(\mathbf{c}_a - \mathbf{c}_b)^\top\}$$

based on our assumption of "simple" matches (Section 4.2). Our real solution is not exact due to image noise, and we note that the dimension of $Ker(A_{17})$ is at least 3+1. Experiments have confirmed that this dimension is 4.

We build a basis of 3 exact solutions $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and a non-exact solution $\mathbf{y}$ with the singular vectors corresponding to the four smallest singular values of $\mathtt{A}_{17}$. The singular values of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are 0 (up to computer accuracy) and that of $\mathbf{y}$ is 0 (up to image noise). We calculate the real solution $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ by linear combination of $\mathbf{y}$, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ such that the resulting matrix $\tilde{\mathtt{R}}$ verifies $\tilde{\mathtt{R}}^\top \tilde{\mathtt{R}} = \lambda \mathtt{I}_{3\times 3}$ or $\tilde{\mathtt{E}}$ is an essential matrix. Let $\mathbf{l}$ be the vector such that $\mathbf{l}^\top = [\lambda_1 \ \lambda_2 \ \lambda_3]^\top$, and thus we denote as $\tilde{\mathtt{R}}(\mathbf{l})$ and $\tilde{\mathtt{E}}(\mathbf{l})$ the matrix $\tilde{\mathtt{R}}$ and $\tilde{\mathtt{E}}$ extracted from solution $\mathbf{y} - [\mathbf{x}_1|\mathbf{x}_2|\mathbf{x}_3]\mathbf{l}$. Using these notations, we have $\tilde{\mathtt{R}}(\mathbf{l}) = \mathtt{R}_0 - \sum_{i=1}^3 \lambda_i \mathtt{R}_i$ and $\tilde{\mathtt{E}}(\mathbf{l}) = \mathtt{E}_0 - \sum_{i=1}^3 \lambda_i \mathtt{E}_i$ with $(\mathtt{R}_i, \mathtt{E}_i)$ extracted from $\mathbf{x}_i$.

Once the basis $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ is calculated, we compute the coordinates of the solution by non-linear minimization of the function $(\lambda, \mathbf{l}) \rightarrow \|\lambda \mathtt{I}_{3\times 3} - \mathtt{R}(\mathbf{l})^\top . \mathtt{R}(\mathbf{l})\|^2$ to obtain $\mathbf{l}$ and thus $\tilde{\mathtt{E}}$. SVD decomposition is applied to $\tilde{\mathtt{E}}$, and we obtain 4 solutions [9] for $([\mathbf{t}]_\times, \mathtt{R})$. The solution with the minimal epipolar constraint $\|\mathtt{A}_{17}\mathbf{x}\|$ is then selected. Lastly, we refine the 3D scale k by minimizing $k \rightarrow \sum_i ({\mathbf{q}'_{0i}}^\top \mathtt{R} \mathbf{q}_{1i} - \mathbf{q}_{0i}^\top k . [\mathbf{t}]_\times \mathtt{R} \mathbf{q}_{1i} + \mathbf{q}_{0i}^\top \mathtt{R} \mathbf{q}'_{1i})^2$ and perform $\mathbf{t} \leftarrow k\mathbf{t}$.

Non-Axial Camera For a non-axial camera (e.g. a multicamera system with perspective cameras such that centers are not collinear), the problem is also different. Appendix B shows that the "exact" solutions are $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}}) \in Vect\{(\mathtt{I}_{3\times3}, \mathtt{0}_{3\times3})\}$ based on our assumption of "simple" matches (Section 4.2). Our real solution is not exact due to image noise, and we see that the dimension of $Ker(A_{17})$ is at least 1+1. Experiments have confirmed that this dimension is 2. We have not yet experimented this case.

4.4 Initialization with Three Views (RANSAC process)

The first step of the incremental algorithm is the 3D reconstruction of a subsequence containing the first key frames triplet $\{0, 1, 2\}$. A number of random samples are taken, each containing 17 points. For each sample, the relative pose between views 0 and 2 is computed using the above described method and matched points are triangulated. The pose of camera 1 is estimated with 3D/2D correspondences by iterative refinement minimizing the angular error (see details in Section 3.3). The same error is minimized to triangulate points. Finally, the solution producing the highest number of inliers in views 0, 1, and 2 is selected from among all samples. The j-th 3D point is considered as an inlier in view i if the angular error $||\epsilon_j^i||$ is less than ϵ ($\epsilon = 0.01\ rad$ in our experiments).

4.5 Pose Estimates (RANSAC)

The generic pose calculation is useful for both steps of our approach (initialization and incremental process). We assume that the i-th pose $P^i = (\mathtt{R}^i, \mathbf{t}^i)$ of the camera is close to that of the i-1-th pose $P^{i-1} = (\mathtt{R}^{i-1}, \mathbf{t}^{i-1})$. P^i is estimated by iterative non-linear optimization initialized at P^{i-1} with a reduced sample of five 3D/2D correspondences, in conjunction with RANSAC. For each sample, the pose is estimated by minimizing an angular error (Section 3.3) and we count the number of inliers (points) for this pose. The pose with the maximum number of inliers is then selected and another optimization is applied with all inliers.

5 Experiments using a Perspective Camera

We have applied our incremental localization and mapping algorithm to a semi-urban scene. The goal of these experiments was to evaluate robustness (resistance to perturbations) in a complex environment and compare accuracy

to the ground truth provided by a Real-Time Kinematics Differential GPS. The camera was therefore mounted on an experimental vehicle whose velocity was about 1.1 m/s. The distance covered was about 70 m and the video sequence was 1 min long (Image size was 512×384 $pixels$ at 7.5 fps). More than 4.000 3D points were reconstructed and 94 images selected as key frames from a series of 445 (figure 11). This sequence was particularly interesting because of image content (Figure 9 shows people walking in front of the camera, sunshine, etc...) which was not conducive to the reconstruction process. Moreover, the environment was more appropriate to a GPS localization (which provides ground truth) since the satellites involved were not occluded by high buildings.

Fig. 9. Two frames used for the real data experiments.

5.1 Processing Time

Calculations were performed on a standard Linux PC (Pentium 4 at 2.8 GHz, 1Go of RAM memory at 800 MHZ). Image processing times through the sequence were as shown in Figure 10. Times measured included feature detection (#1500 Harris points per frame), matching, and pose calculation for all frames. For key frames, longer processing times were necessary (see Figure 10) due to 3D points reconstruction and local bundle adjustment. We took $n = 3$ (number of optimized camera poses) and used $N = 10$ (number of cameras for minimization of reprojection criterion). Note that computing speeds were interesting with an average of 0.09 s for normal frames and 0.28 s for key frames. These data are shown in Table 3.

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

Table 3
Results. Computing times are given in *seconds*.

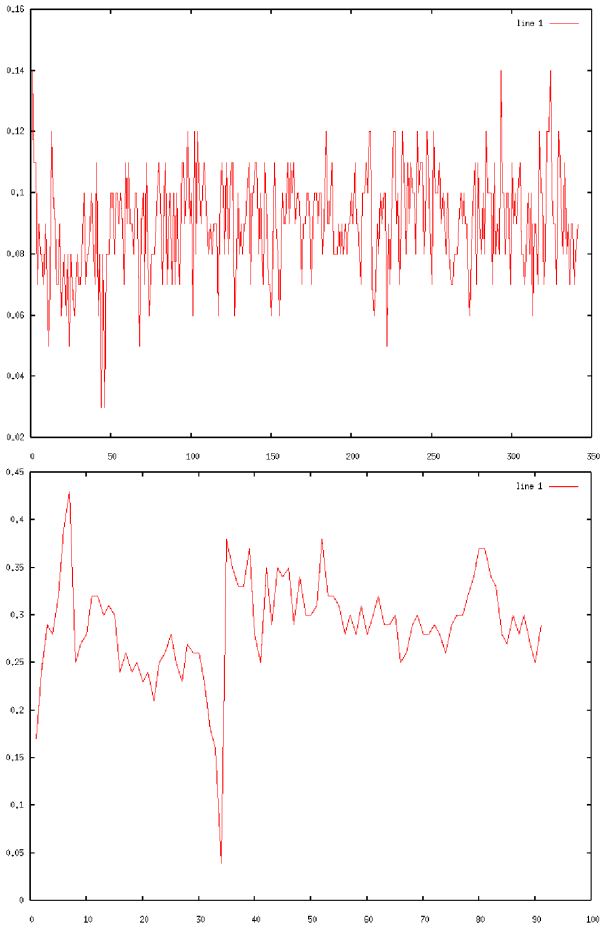

Fig. 10. Processing time (in *seconds*) for non-key frames and key frames.

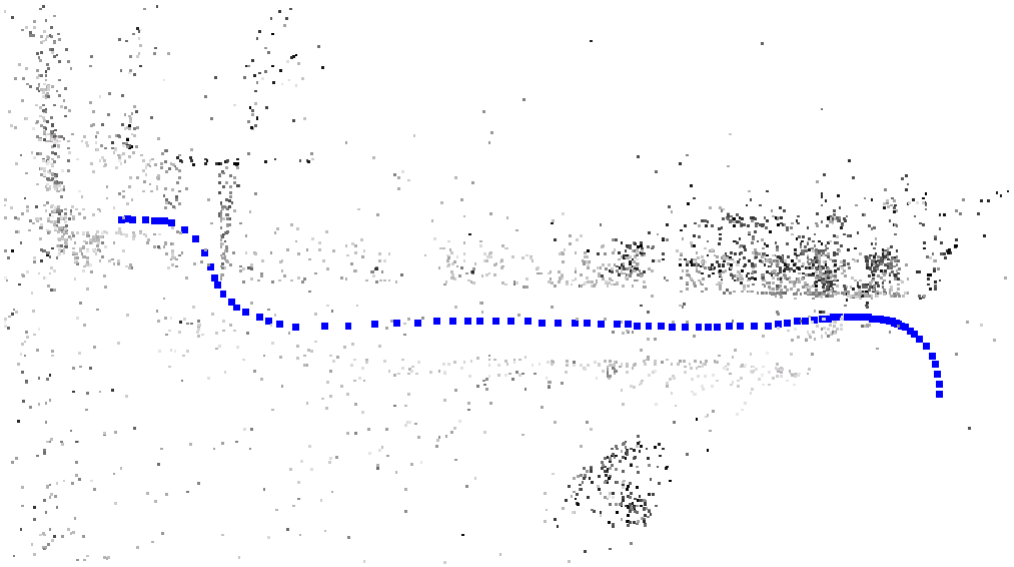

Fig. 11. Top view of the reconstructed scene and trajectory (# 4.000 points and 94 key positions).

The calculated trajectory obtained with our algorithm was compared against data given by a GPS sensor. For purposes of comparison, we applied a rigid transformation (rotation, translation and scale factor) to the trajectory as described in [5] to fit with GPS reference data. Figure 12 shows trajectory registration with the GPS reference. As GPS positions are given in a metric frame we could 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 the horizontal plane is:

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

where x_i, y_i, z_i are the estimated coordinates for camera pose i and x_{GPS}, y_{GPS}, z_{GPS} are the corresponding GPS coordinates. Figure 13 shows 2D/3D error variations for the 94 frames. The maximum measured error was 2.0 m with a 3D mean error of 41 cm and a 2D mean error of less than 35 cm.

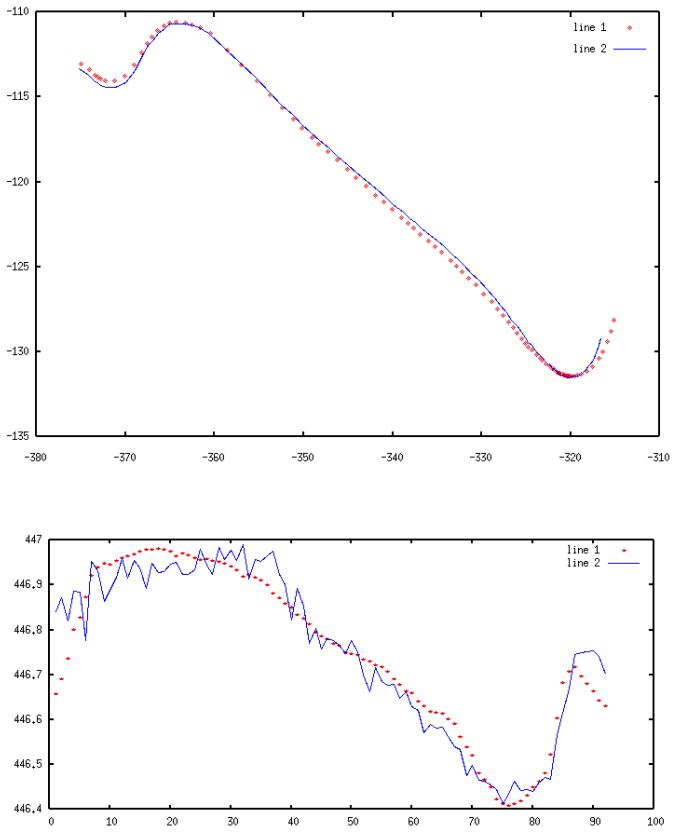

Fig. 12. Registration with the GPS reference, top: in horizontal plane, bottom: along altitude axis. The continuous line represents GPS trajectory and points the estimated key positions. Coordinates are expressed in *meters*.

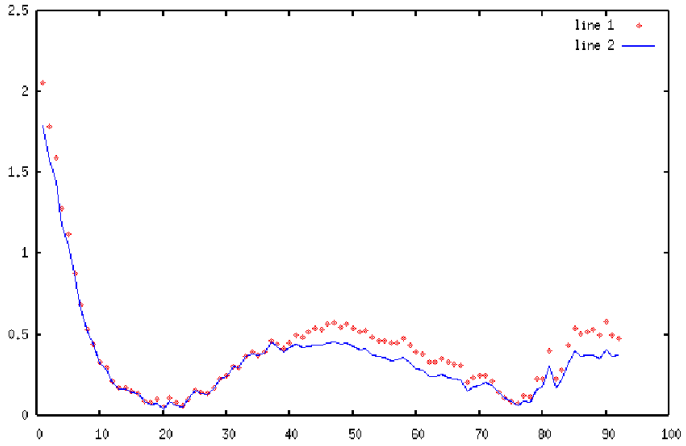

Fig. 13. Error in meters. Continuous line = 2D error, dotted line = 3D error.

5.3 Parametrizing Local Bundle Adjustment

In our incremental method, local bundle adjustment consists of optimizing the end of the reconstruction only, to avoid unnecessary calculations and excessive

computing times. As mentioned before, optimization is applied to the n last estimated camera poses, with allowance for points reprojections in a larger number N of cameras. We therefore tested several values for n and N as reported in Tables 4, 5 and 6.

First, note that we must have $N \geq n+2$ to define the reconstruction frame and the scale factor at the end of the sequence. For $N = n$ or $n + 1$, it happened that the reconstruction was not completed due to process failure before the end of the sequence (Tables 4 and 5). This confirms that $N \geq n + 2$ is an important condition. We also measured mean time processing for local bundle adjustment as a function of n (Table 6). In practice, it does not vary much with N since the mean track length of points in key-frames (Figure 14) is limited.

Then, we compare results with the GPS trajectory (Table 4) and a trajectory computed using global bundle adjustment (Table 5). As expected, the quality increases when n increase. We also observed that higher values of n provides minor improvements of quality in our context. In all the following experiments, we fix $n = 3$ and $N = 10$ since this provides a good trade-off between accuracy and time performance.

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 4
Mean 3D position error (in *meters*) compared to GPS for the incremental method with different n and N values, and for 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 5
Mean 3D position error (in *meters*) compared to global bundle adjustment for different n and N values.

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

Table 6
Mean local bundle adjustment computing times as a function of n for many N values (in *seconds*).

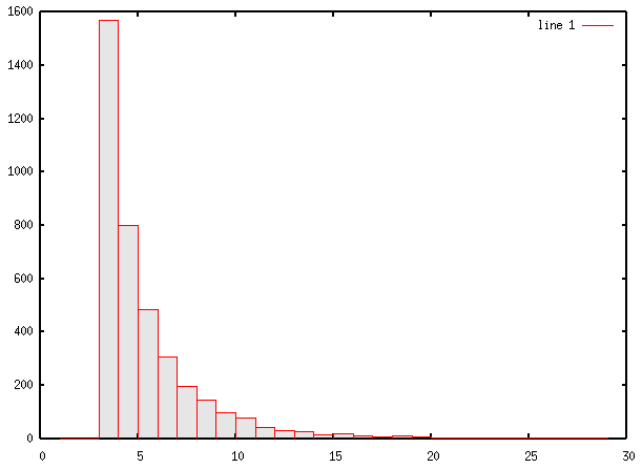

Fig. 14. Point distribution with track length.

5.4 Comparison with Global Bundle Adjustment for a Long Sequence

An other experiment has been carried out in a town-center (Figure 15) where differential GPS is not available for ground truth. One can visually ensure that reconstruction is not much deformed and drift is low compared to the covered distance. The incremental method selects 354 key-frames among 2900 frames and reconstructs 16135 3D points with $n = 3$ and $N = 10$. The estimated mean 3D position error compared to global bundle adjustment is 29cm (the trajectory length is about 500m). That shows that our algorithm (very appropriate to long scene reconstruction in term of computing time) is also robust and gives result similar as that of global bundle adjustment.

6 Experiments using a Generic Camera

The incremental generic 3D reconstruction method was tested on real data with three different cameras: a perspective camera, a catadioptric camera and a stereo rig. Examples of frames are shown in Figure 16 and sequences characteristics in Table 7. Computing performances are reported in Table 8. Following these experiments, the trajectory obtained with our generic method was compared to GPS ground truth or global specific bundle adjustment results. As already seen above, a rigid transformation (rotation, translation and scale

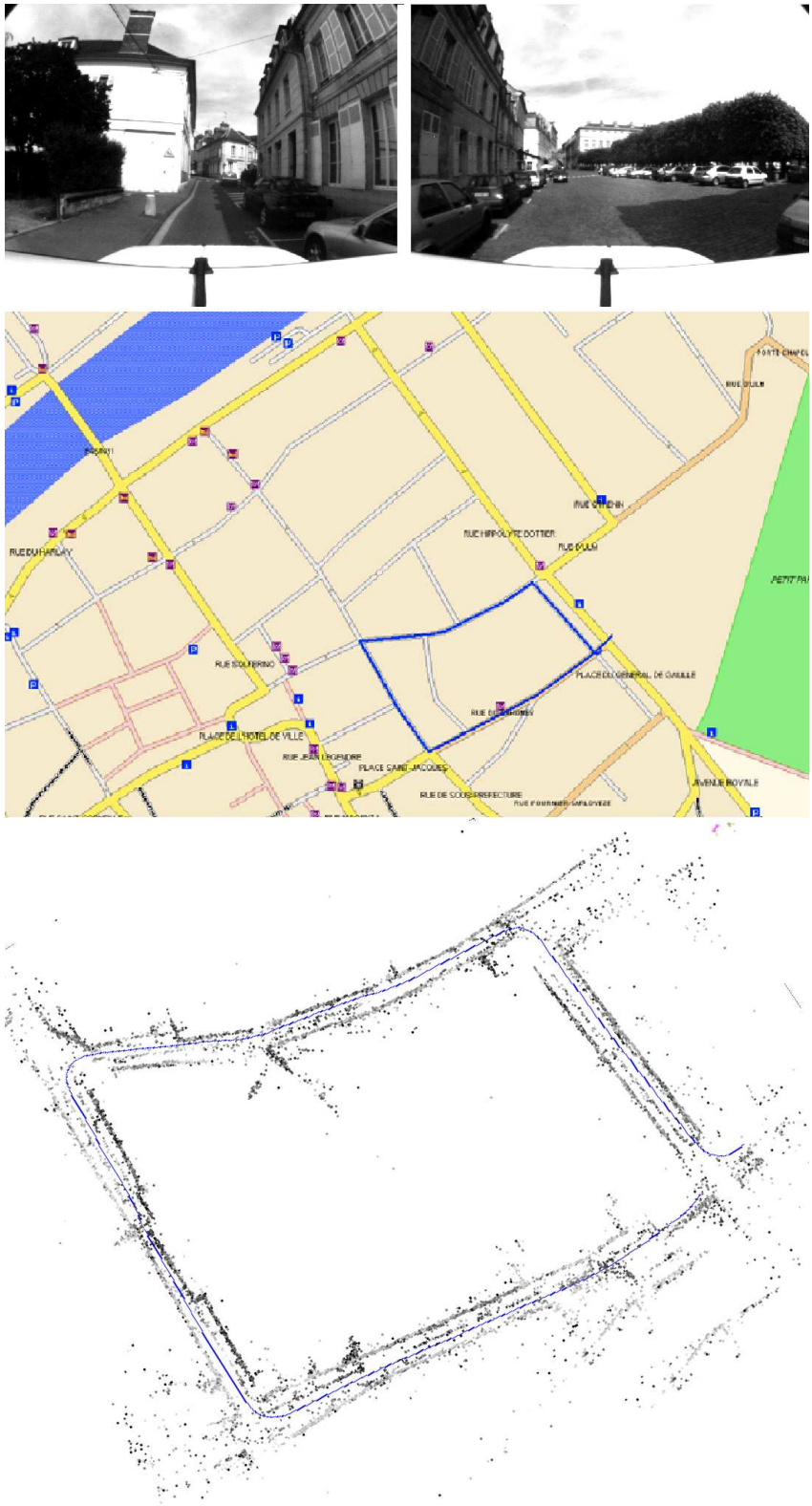

Fig. 15. Two images of the urban video, map and top view of the reconstructed scene and trajectory (16135 points and 354 key positions).

factor) must be applied to the trajectory to fit with reference data [5]. A mean 3D error or 2D error in the horizontal plane can then be measured between the generic and the reference trajectory.

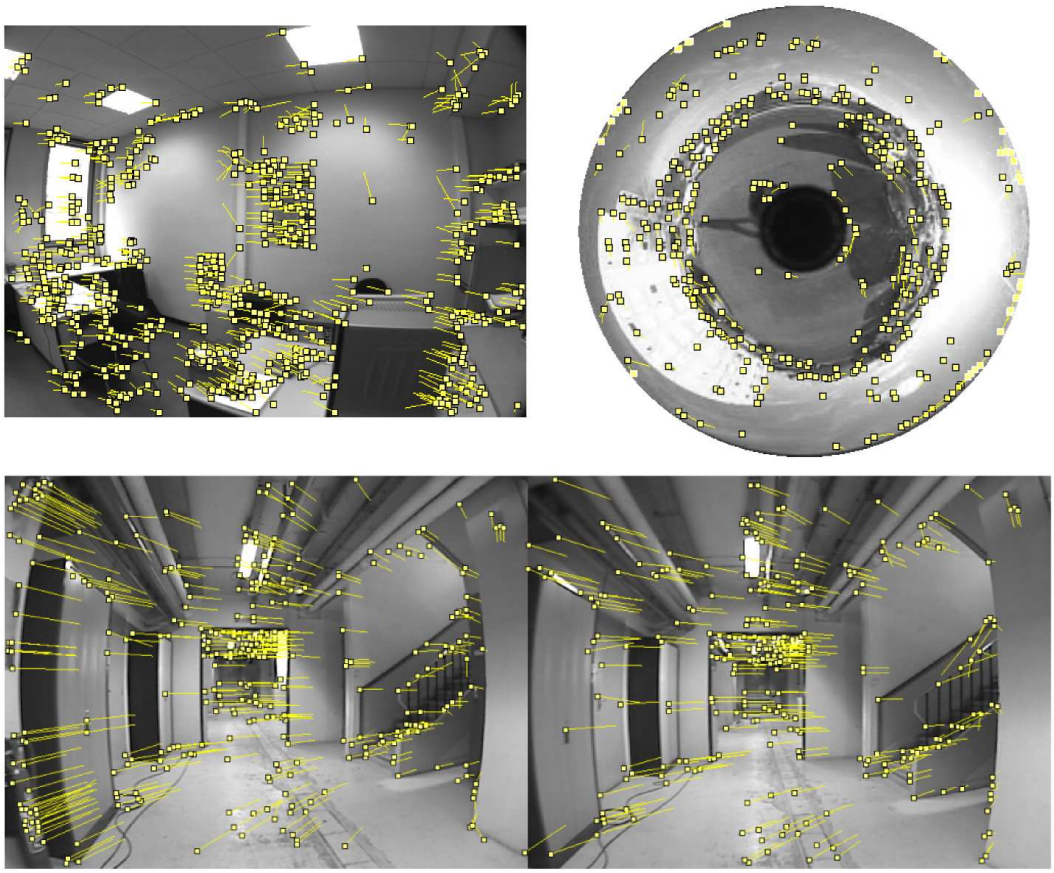

Fig. 16. Feature tracks for a generic camera image, using three types of cameras: perspective (top left), catadioptric (top right), and stereo rig (bottom) cameras. The same matching method was used for all three.

6.1 Comparison with Ground Truth (Differential GPS)

The first results correspond to the same video sequence as in Section 5.2, with a pinhole camera mounted on an experimental vehicle equipped with a differential GPS receiver (inch precision). The calculated motion obtained with our generic algorithm was compared against data given by the GPS sensor, and Figure 17 shows the two recorded trajectories. As GPS positions are given in a metric frame, we could compare camera locations and measure positioning error in meters: mean 3D error was 0.48 m and 2D error in the horizontal plane was 0.47 m. Results were slightly less accurate than those obtained with the specific perspective method.

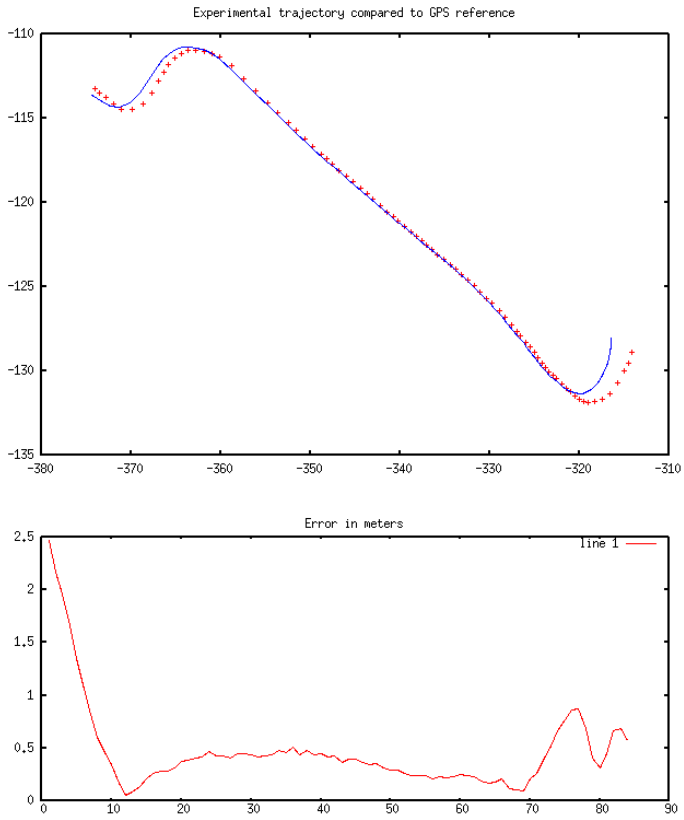

Fig. 17. Top: Registration of generic vision trajectory with GPS ground truth. The continuous line represents GPS and points represent vision estimated positions. Bottom: 3D error along the trajectory.

6.2 Comparison with Specific and Global Bundle Adjustment

In the following examples, ground truth is not available. We therefore compare our results against those of the best method available: a global and specific bundle adjustment (all 3D parameters have been refined so as to obtain an optimal solution with a minimum reprojection error). Sequences characteristics and results are reported in Table 7.

Sequence 1 is taken in an indoor environment using an handheld pinhole camera. A very small difference is obtained: the mean 3D error is less than 6.5 cm for a trajectory length of (about) 15 m. The relative error is 0.45%.

Sequence 2 is taken in an outdoor environment, using an handheld catadioptric camera (the 0-360 mirror with the Sony HDR-HC1E camera shown on Figure 18, DV format). The useful part of the rectified image is contained in a circle whose diameter is 458 pixels. The difference is also small: the mean 3D error is less than 9.2 cm for a trajectory length of (about) 40 m. Relative error is 0.23%.

Sequence 3 is taken with a stereo rig (baseline: 40 cm) in a corridor (Figure 18). The image is composed of two sub-images of 640×480 pix. The trajectory (20 m long) is compared to results obtained using the left/right camera and global bundle adjustment. Mean 3D error is 2.7/8.4 cm compared with the left/right camera and relative error is 0.13/0.42%.

Sequence	Camera	#Frames	#Key frames	#3D Pts	#2D Pts	Traj. length
Sequence 1	pinhole	511	48	3162	11966	15 m
Sequence 2	catadioptric	1493	132	4752	18198	40 m
Sequence 3	stereo rig	303	28	3642	14189	20 m

Table 7
Characteristics of video sequences.

Camera	Image size	Detection+Matching	Frame	Key frame	Mean rate
Pinhole	512×384	0.10	0.14	0.37	**6.3 fps**
Catadioptric	464×464	0.12	0.15	0.37	**5.9 fps**
Stereo rig	1280×480	0.18	0.25	0.91	**3.3 fps**

Table 8
Computing times in *seconds* for the three cameras.

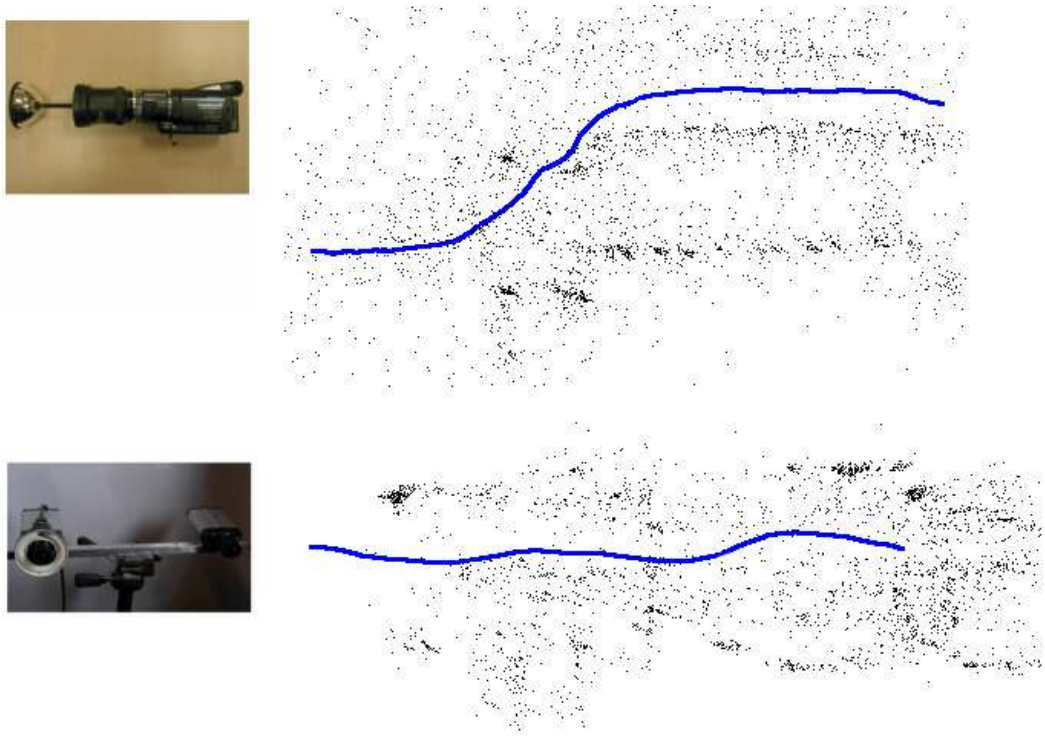

Fig. 18. Cameras and top views of 3D reconstructions. Top: catadioptric camera (Sequence 2). Bottom: stereo rig (Sequence 3). Trajectory in blue and 3D points in black.

7 Conclusion

We have developed and tested a method for solving the real-time Structure from Motion problem. This paper presents a complete process for doing so, from feature detection and matching to estimating geometry and refining it by local bundle adjustment. 3D reconstruction is an incremental process that begins with initialization using three key frames. For subsequent frames, camera localization is then determined and key frames are selected for 3D point reconstruction. This method is extended to a generic camera model facilitating the change from one kind of camera to another. Promising results have been obtained on real data with three different types of cameras (even if there is room for further optimizing implementation). Results and time complexities are compared favorably with those of global bundle adjustment (and ground truth when available). Future work includes experiments on more complex multicamera systems and automatic 3D modeling methods using the generic camera model.

Appendix A: Complexity of One Bundle Adjustment Iteration

Let $\mathbf{P}$ be the vector of parameters to be estimated (orientation of the cameras + position of their optical centers + 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. The problem here is to minimize the function $\phi(\mathbf{P}) = ||f(\mathbf{P}) - \mathbf{X}||^2$.
At stage k of the iterative algorithm, we calculate $\mathbf{\Delta}_k$ such that $\mathbf{P}_{k+1} = \mathbf{P}_k + \mathbf{\Delta}_k$. $\mathbf{\Delta}_k$ is obtained by solving the equation: $\mathtt{J}^\top\mathtt{J}.\mathbf{\Delta}_k = \mathtt{J}^\top.\epsilon_k$ where $\mathtt{J}$ 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 $\mathtt{J}^\top\mathtt{J}$ are multiplied by a coefficient using the Levenberg-Marquardt method [21]).

Since we want to process long sequences with multiple parameters to be evaluated (six for each camera and three for each 3D point), it seems natural to make use of bundle adjustment characteristics, i.e. the block structure of matrix $\mathtt{J}^\top\mathtt{J}$ for reconstruction of a set of points [9]. This matrix is composed of three blocks $\mathtt{U}$, $\mathtt{V}$, and $\mathtt{W}$ such that $\mathtt{U}$ and $\mathtt{V}$ are block-diagonal:

- $\mathtt{U}$, matrix made of diagonal 6×6 blocks representative of the dependency of image i measurements on the associated camera parameters.
- $\mathtt{V}$, matrix made of diagonal 3×3 blocks representative of relations between point j parameters and the measurements associated with this point.
- $\mathtt{W}$, matrix expressing the inter-correlations between 3D points parameters and cameras parameters. The structure of $\mathtt{W}$ is linked to the fact that many

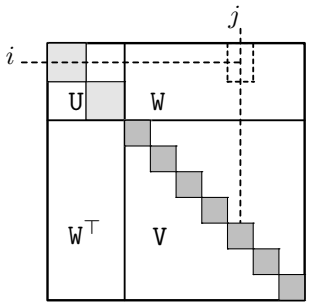

Fig. 19. Structure of the $\mathtt{J}^\top\mathtt{J}$ matrix

points are not projected through all cameras. $\mathtt{W}$ has a number of non-null 6×3 blocks equal to the number of 2D reprojections (half of the dimension of $\mathbf{X}$ or $f(\mathbf{P})$).

The system is therefore $\begin{pmatrix} \mathtt{U} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{V} \end{pmatrix} \begin{pmatrix} \mathbf{\Delta}_{cameras} \\ \mathbf{\Delta}_{points} \end{pmatrix} = \begin{pmatrix} \mathbf{Y}_{cameras} \\ \mathbf{Y}_{points} \end{pmatrix}$, and it is solved in two steps [9]:

(1) Calculation of the increment $\mathbf{\Delta}_{cameras}$ to be applied to cameras by resolution of the following system:

$$(\mathtt{U} - \mathtt{W}\mathtt{V}^{-1}\mathtt{W}^\top)\mathbf{\Delta}_{cameras} = \mathbf{Y}_{cameras} - \mathtt{W}\mathtt{V}^{-1}\mathbf{Y}_{points} \tag{3}$$

(2) Direct calculation of the increment $\mathbf{\Delta}_{points}$ applicable to 3D points:

$$\mathbf{\Delta}_{points} = \mathtt{V}^{-1}(\mathbf{Y}_{points} - \mathtt{W}^\top\mathbf{\Delta}_{cameras})$$

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

Once $\mathtt{J}^\top\mathtt{J}$ is calculated (Figure 19) (time complexity is proportional to the number N_r of 2D reprojections taken into account), the two most time-consuming expensive stages of this resolution are:

- calculation of matrix product $\mathtt{W}\mathtt{V}^{-1}\mathtt{W}^\top$
- resolution of camera linear system (3).

For matrix product $\mathtt{W}\mathtt{V}^{-1}\mathtt{W}^\top$, the number of necessary operations can be determined by first considering the number of non-null blocks of $\mathtt{W}\mathtt{V}^{-1}$. This is the same number as $\mathtt{W}$, i.e. $(p.C)$, number of reprojections in C images, because $\mathtt{V}^{-1}$ is block-diagonal. Then, in the product $(\mathtt{W}\mathtt{V}^{-1})\mathtt{W}^\top$, each non-null 6×3 block of $\mathtt{W}\mathtt{V}^{-1}$ is used once in the calculation of each block column of $\mathtt{W}\mathtt{V}^{-1}\mathtt{W}^\top$. Thus the time complexity of the product $\mathtt{W}\mathtt{V}^{-1}\mathtt{W}^\top$ is $\Theta(p.C^2)$. The time complexity

of the traditional resolution of the linear system (1) is $\Theta(C^3)$ [21]. Therefore the time complexity of one bundle adjustment iteration is

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

Appendix B: Exact Solutions of the (linearized) Pless Equation

The i-th pair of rays is defined by Plucker Coordinates $(\mathbf{q}_{0i}, \mathbf{q}'_{0i})$ and $(\mathbf{q}_{1i}, \mathbf{q}'_{1i})$ such that $\mathbf{q}'_{0i} = \mathbf{q}_{0i} \wedge \mathbf{c}_{0i} = -[\mathbf{c}_{0i}]_\times \mathbf{q}_{0i}$ and $\mathbf{q}'_{1i} = \mathbf{q}_{1i} \wedge \mathbf{c}_{1i} = -[\mathbf{c}_{1i}]_\times \mathbf{q}_{1i}$. The rays of the i-th pair intersect if they satisfy the Pless Equation

$$0 = {\mathbf{q}'_{0i}}^\top \tilde{\mathtt{R}} \mathbf{q}_{1i} - \mathbf{q}_{0i}^\top \tilde{\mathtt{E}} \mathbf{q}_{1i} + \mathbf{q}_{0i}^\top \tilde{\mathtt{R}} \mathbf{q}'_{1i} = {\mathbf{q}_{0i}}^\top ([\mathbf{c}_{0i}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{E}} - \tilde{\mathtt{R}}[\mathbf{c}_{1i}]_\times) \mathbf{q}_{1i} \qquad (4)$$

where the two 3×3 matrices $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ are the unknowns. In this section, we seek an $(\tilde{\mathtt{R}}, \tilde{\mathtt{E}})$ such that $\forall i, \tilde{\mathtt{E}} = [\mathbf{c}_{0i}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_{1i}]_\times$. In other words, we seek an $\tilde{\mathtt{R}} \in \Re^{3\times 3}$ such that this $\tilde{\mathtt{E}}$ is independent of any available camera center pair $(\mathbf{c}_{0i}, \mathbf{c}_{1i})$. As a consequence, Equation 4 is exactly equal to 0 (up to computer accuracy) whatever $(\mathbf{q}_{0i}, \mathbf{q}_{1i})$. We consider many cases.

Simple Matches Only: $\forall i, \mathbf{c}_{0i} = \mathbf{c}_{1i}$

As mentioned in Section 4.2, this particular case is important in practice.

Central Camera Let $\mathbf{c}$ be the center. This case is straightforward: we have $\mathbf{c}_{0i} = \mathbf{c}_{1i} = \mathbf{c}$ and $\tilde{\mathtt{E}} = [\mathbf{c}]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}]_\times$. Any $\tilde{\mathtt{R}} \in \Re^{3\times 3}$ is possible.

Stereo Camera Let $\mathbf{c}_a$ and $\mathbf{c}_b$ be the centers. We have $(\mathbf{c}_{0i}, \mathbf{c}_{1i}) \in \{(\mathbf{c}_a, \mathbf{c}_a), (\mathbf{c}_b, \mathbf{c}_b)\}$ and $\tilde{\mathtt{E}} = [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times = [\mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_b]_\times$. The constraint on $\tilde{\mathtt{R}}$ is $[\mathbf{c}_a - \mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a - \mathbf{c}_b]_\times = 0$. Any $\tilde{\mathtt{R}}$ in the linear space of polynomials of $[\mathbf{c}_a - \mathbf{c}_b]_\times$ is possible. Furthermore, it is easy to show that this constraint does not allow another $\tilde{\mathtt{R}}$ by changing the coordinate basis such that $\mathbf{c}_a - \mathbf{c}_b = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$. Thus,

$$\tilde{\mathtt{E}} = [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times \text{ and } \tilde{\mathtt{R}} \in Vect\{\mathtt{I}_{3\times 3}, [\mathbf{c}_a - \mathbf{c}_b]_\times, (\mathbf{c}_a - \mathbf{c}_b)(\mathbf{c}_a - \mathbf{c}_b)^\top\}.$$

Axial Camera All camera centers of an axial camera are collinear: there are $\mathbf{c}_a$ and $\mathbf{c}_b$ such that $\mathbf{c}_{0i} = \mathbf{c}_{1i} = (1 - \lambda_i)\mathbf{c}_a + \lambda_i \mathbf{c}_b$ with $\lambda_i \in \Re$. Thus,

$$\begin{aligned} \tilde{\mathtt{E}}(\lambda) &= [(1-\lambda)\mathbf{c}_a + \lambda \mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[(1-\lambda)\mathbf{c}_a + \lambda \mathbf{c}_b]_\times \\ &= [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times + \lambda([\mathbf{c}_b - \mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_b - \mathbf{c}_a]_\times) \end{aligned}$$

should not depend on λ. We see that $[\mathbf{c}_b - \mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_b - \mathbf{c}_a]_\times = 0$. This case is the same as the previous one.

Non-Axial Camera There are three non-collinear centers $\mathbf{c}_a, \mathbf{c}_b, \mathbf{c}_c$. The constraint on $\tilde{\mathtt{R}}$ is

$$0 = [\mathbf{c}_a - \mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a - \mathbf{c}_b]_\times = [\mathbf{c}_b - \mathbf{c}_c]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_b - \mathbf{c}_c]_\times = [\mathbf{c}_c - \mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_c - \mathbf{c}_a]_\times.$$

This constraint is three times that obtained for the stereo camera. Let's change the coordinate basis such that $\{\mathbf{c}_a - \mathbf{c}_b, \mathbf{c}_b - \mathbf{c}_c, \mathbf{c}_c - \mathbf{c}_a\}$ is the canonical basis and write the solutions for the three stereo cases: any $\tilde{\mathtt{R}} \in Vect\{\mathtt{I}_{3\times 3}\}$ is possible.

Whether Matches are Simple or Not

In this case, the "simple match" constraint is not enforced.

Central Camera It does not occur here.

Stereo Camera We have $(\mathbf{c}_{0i}, \mathbf{c}_{1i}) \in \{(\mathbf{c}_a, \mathbf{c}_a), (\mathbf{c}_b, \mathbf{c}_b), (\mathbf{c}_a, \mathbf{c}_b), (\mathbf{c}_b, \mathbf{c}_a)\}$ and $\tilde{\mathtt{E}} = [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times = [\mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_b]_\times = [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_b]_\times = [\mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times$. Thus, the constraint on $\tilde{\mathtt{R}}$ is $0 = [\mathbf{c}_a - \mathbf{c}_b]_\times \tilde{\mathtt{R}} = \tilde{\mathtt{R}}[\mathbf{c}_a - \mathbf{c}_b]_\times$. This constraint is stronger than that for the stereo camera with simple matches: we see that any $\tilde{\mathtt{R}} \in Vect\{(\mathbf{c}_a - \mathbf{c}_b)(\mathbf{c}_a - \mathbf{c}_b)^\top\}$ is possible.

Axial Camera There are $\mathbf{c}_a$ and $\mathbf{c}_b$ such that $\mathbf{c}_{0i} = (1 - \lambda_{0i})\mathbf{c}_a + \lambda_{0i}\mathbf{c}_b$ and $\mathbf{c}_{1i} = (1 - \lambda_{1i})\mathbf{c}_a + \lambda_{1i}\mathbf{c}_b$ with $\lambda_{0i} \in \Re$ and $\lambda_{1i} \in \Re$. Thus,

$$\begin{aligned}\tilde{\mathtt{E}}(\lambda_0, \lambda_1) &= [(1 - \lambda_0)\mathbf{c}_a + \lambda_0 \mathbf{c}_b]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[(1 - \lambda_1)\mathbf{c}_a + \lambda_1 \mathbf{c}_b]_\times \\ &= [\mathbf{c}_a]_\times \tilde{\mathtt{R}} - \tilde{\mathtt{R}}[\mathbf{c}_a]_\times + \lambda_0 [\mathbf{c}_b - \mathbf{c}_a]_\times \tilde{\mathtt{R}} - \lambda_1 \tilde{\mathtt{R}}[\mathbf{c}_b - \mathbf{c}_a]_\times\end{aligned}$$

should not depend on λ_0 and λ_1: we have $0 = [\mathbf{c}_b - \mathbf{c}_a]_\times \tilde{\mathtt{R}} = \tilde{\mathtt{R}}[\mathbf{c}_b - \mathbf{c}_a]_\times$. This case is the same as the previous one.

Non-Axial Camera There are three non-collinear centers $\mathbf{c}_a, \mathbf{c}_b, \mathbf{c}_c$ and the constraint on $\tilde{\mathtt{R}}$ is three times the one obtained for the stereo camera:

$$0 = [\mathbf{c}_a - \mathbf{c}_b]_\times \tilde{\mathtt{R}} = \tilde{\mathtt{R}}[\mathbf{c}_a - \mathbf{c}_b]_\times = [\mathbf{c}_b - \mathbf{c}_c]_\times \tilde{\mathtt{R}} = \tilde{\mathtt{R}}[\mathbf{c}_b - \mathbf{c}_c]_\times = [\mathbf{c}_c - \mathbf{c}_a]_\times \tilde{\mathtt{R}} = \tilde{\mathtt{R}}[\mathbf{c}_c - \mathbf{c}_a]_\times.$$

Let's change the coordinate basis such that $\{\mathbf{c}_a - \mathbf{c}_b, \mathbf{c}_b - \mathbf{c}_c, \mathbf{c}_c - \mathbf{c}_a\}$ is the canonical basis and write the solutions for the three stereo cases: we obtain $\tilde{\mathtt{R}} = 0$. This method does not provide exact solutions.

References

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

[2] P. Chang and M. Hebert. Omni-directional structure from motion. In *Proc. of the 2000 IEEE Workshop on Omnidirectional Vision*, pages 127–133, 2000.

[3] A. Davison. Real-time simultaneous localization and mapping with a single camera. In *Proc. of International Conference on Computer Vision*, 2003.

[4] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV)*, August 2006.

[5] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotic Research*, 5:27–52, 1986.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus, a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381 – 395, 1981.

[7] R. M. Haralick, C-N Lee, K. Ottenberg, and M. Noelle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal on Computer Vision*, 13(3):331–356, 1994.

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *4th ALVEY Vision Conference*, pages 147–151, 1988.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[10] J. Kannala and S.S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *Transactions on Pattern Analysis and Machine Intelligence*, 28:1335–1340, 2006.

[11] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005.

[12] M. Lhuillier. Effective and Generic Structure from Motion. *International Conference on Pattern Recognition*, 2006.

[13] C.P. Lu, G.D. Hager and E. Mjolsness, Fast and Globally Convergent Pose Estimation from Video Images. *Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[14] B. Micusik and T. Pajdla. Autocalibration & 3D reconstruction with non-central catadioptric cameras. In *Proc. of Conference on Computer Vision and Pattern Recognition*, 2004.

[15] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *Proc. of Conference on Computer Vision and Pattern Recognition*, June 2006.

[16] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion. *Proc. of the British Machine Vision Conference*, 2007.

[17] D. Nister. An efficient solution to the five-point relative pose problem. *Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004.

[18] D. Nister, O. Naroditsky, and J Bergen. Visual odometry. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 652–659, 2004.

[19] R. Pless. Using many cameras as one. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages II: 587–593, 2003.

[20] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Automated reconstruction of 3D scenes from sequences of images. *Isprs Journal Of Photogrammetry And Remote Sensing*, 55(4):251–267, 2000.

[21] W. H. Press, Saul A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C*. Cambridge University Press, second edition, 1992. The art of scientific computing.

[22] S. Ramalingam, S. Lodha, and P. Sturm. A generic structure-from-motion framework. *Computer Vision and Image Understanding*, 103(3):218–228, 2006.

[23] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau. Localization in urban environments: monocular vision compared to a differential gps sensor. In *Proc. of Conference on Computer Vision and Pattern Recognition*, 2005.

[24] H. Y. Shum, Q. Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In *Proc. of Conference on Computer Vision and Pattern Recognition*, 1999.

[25] D. Steedly and I. Essa. Propagation of innovative information in Non-Linear Least-Squares structure from motion. In *Proc. of International Conference on Computer Vision*, pages 223–229, 2001.

[26] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, 2005.

[27] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[28] Z. Zhang and Y. Shan. Incremental motion estimation through modified bundle adjustment. In *Proc. of International Conference on Image Processing*, pages II: 343–346, 2003.