

Bundle Adjustment and Incidence of Linear Features on the Accuracy of External Calibration Parameters

Franck JUNG, Didier BOLDO
MATIS Laboratory, IGN, 2-4 Av. Pasteur, 94165 Saint Mande Cedex, FRANCE
first.lastname@ign.fr

KEY WORDS: Bundle Adjustment, Linear Features, Accuracy, Photogrammetry

ABSTRACT:

In this paper we investigate the influence of linear features (segments) in a bundle adjustment. Bundle adjustment is the problem of refining a set of parameters (internal calibration, external calibration, 3D model). The refinement is performed by the minimization of a cost function. In usual photogrammetric applications, this cost function is based on image tie points and 3D control points. The cost function measures the distance between the observed data and the model. Linear features are especially important when dealing with architectural or terrestrial images, since they do not require segment extremity to match. So, they are easier to detect automatically within particular scenes (buildings, landscapes...). First, we present the basic concepts of bundle adjustment and the integration of linear features. Using the same concept, the linear feature model is based on the distance between the 3D line re projection in the image and the detected image segments. We describe an algorithm for the resolution of this non-linear least-square problem under constraints. Second, we study the influence of these features on two cases. The first case is a calibration polygon, with a large overlap between images. The second one is a facade of a building. We compare a statistical evaluation of the reliability of the estimated parameters to the theoretical bounds calculated with the eigenvalues and eigenvectors of the Hessian matrix associated to our problem. We stress that segments can be relevant features and can highly increase precision.

1. INTRODUCTION

Accurate parameter estimation is paramount for many image based applications. Feature matching and 3D feature reconstruction are intimately linked to camera parameter estimation. Hence, the knowledge of the viewing parameters permits to face the 3D structure reconstruction problem. Conversely, accurate feature matching can provide a refinement of the viewing parameter estimation. For instance, viewing parameter estimation is a problem that both the computer vision community and the photogrammetry community have tackled for the last decades. Moreover, both communities are currently able to provide a solution for a rough estimation of numerous parameters using different techniques. On the one hand, linearity of geometric relations using projective geometry provides simple expressions for many parameters (Heuel, 2001). Hence, robust algorithms providing viewing parameters (Faugeras et al., 2001), (Xu and Zhang, 1996), (Torr and Davidson, 2003) are widespread in the computer vision community. These robust estimations often lack accuracy or metric interpretation. On the other hand, many problems in the photogrammetric community can be faced thanks to a rough estimation of the viewing parameters (using GPS, INS). Thus, given a rough estimation of the parameters, bundle adjustment provides jointly a refinement of the 3D structure and of the viewing parameters (Triggs et al., 2000), (Hartley and Zisserman, 2002), (Kraus, 1993)). Usually, only points are used for the process. But, in man-made environments, linear features are often easier to detect. So, it is of importance to include them in the bundle adjustment process, as well as to quantify their contribution. The quantification can be done by estimating the accuracy improvement of each parameter. If the measurements used for bundle adjustment are Gaussian, we can use error propagation results in order to associate an uncertainty to each estimated parameter. Correlation between different parameters can also be estimated (Triggs et al., 2000), (Hartley and Zisserman, 2002), (Förstner, 2004).

2. BUNDLE ADJUSTMENT

2.1 Introduction

Bundle adjustment is the well known process used to refine a set of parameters such as camera positions and orientations, 3D point and line positions. It relies on the minimization of a cost function based on all available observations such as Ground Control Points (GCP), tie points and tie lines. In this section, we suc-

cessively describe the parameters, the cost function and, finally, the observations.

2.2 Parameters

Parameters are representative of the problem modeling. In this problem, three kinds of parameters are to be retrieved : camera's, points' and linear features'.

Camera Camera position is obviously represented by three coordinates (x, y, z) in a true 3D system. Multiple representations exist for the camera orientation. We have chosen to use a unitary quaternion to represent it. This representation is non-minimal but simple and never degenerate. Unitary quaternions are represented as a quadruplet of value (q_0, q_1, q_2, q_3) , with constraint :

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 = 0 \quad (1)$$

and there is a simple mapping taking unitary quaternions to rotation matrices (Chou and Kamel, 1991). The system has 7 unknowns and 1 constraint for each camera.

Point feature A point is represented by three coordinates (x, y, z) in a true 3D system. So the system has 3 unknowns for each point.

Linear feature A linear feature can be rather tricky to represent. Both computer vision and photogrammetry communities have tried to introduce these features for camera parameter estimations (Habib, 1999), (Xu and Zhang, 1996). We have chosen a simple model: linear features are considered infinite in 3D, and their image measurement is simply the position of their extremities (Kumar and Hanson, 1994).

3D linear features have only 4 degrees of freedom, but representations using a minimal set of parameters can be rather complex. So we have chosen to use a simple 6 parameter representation : a line is represented by a start point P and a direction \vec{u} . Any point M on the line can be expressed as $M = P + \kappa\vec{u}$, with $\kappa \in \mathbb{R}$. To get a unique representation, P and \vec{u} are subject to two constraints :

$$\|\vec{u}\|^2 - 1 = 0 \quad (2)$$

and

$$\overrightarrow{OP} \cdot \vec{u} = 0 \quad (3)$$

where O has coordinates $(0, 0, 0)$. The system has 6 unknowns and 2 constraints for each line.

2.3 Cost function

Bundle adjustment is based on the minimization of a cost function F , which depends on the parameters that have been defined. Let $\vec{X} \in \mathbb{R}^{N_u}$ be the vector of all parameters. N_u is the number of unknowns : 7 per image, 3 per point, and 6 per line. The cost function is generally the sum of the squared differences between computed and observed values :

$$F(\vec{X}) = \sum_{i=1}^{N_o} \left(r_i(\vec{X}) \right)^2 = \sum_{i=1}^{N_o} \left(\frac{v_i^{obs} - v_i^{comp}(\vec{X})}{\sigma_i} \right)^2 \quad (4)$$

where

- N_o is the number of observations,
- $r_i(\vec{X})$ is the i^{th} element of the normalized residual vector $\vec{r} \in \mathbb{R}^{N_o}$, $r_i(\vec{X}) = \frac{v_i^{obs} - v_i^{comp}(\vec{X})}{\sigma_i}$,
- v_i^{obs} is the i^{th} observation,
- $v_i^{comp}(\vec{X})$ is the computed value of the i^{th} observation,
- σ_i is the expected standard deviation of v_i^{obs} .

Constraints have also to be taken into account. Let N_c be the total number of constraints : 1 per quaternion (see equation 1) and 2 per line (see equations 2 and 3). Let $\vec{c} \in \mathbb{R}^{N_c}$ be the vector of all constraints. Satisfying constraints is equivalent to $\vec{c}(\vec{X}) = \vec{0}$.

The minimization problem consists of finding \vec{X} :

$$\vec{X} = \arg \min \left(F(\vec{X}) \right) \text{ subject to constraints } \vec{c}(\vec{X}) = \vec{0} \quad (5)$$

2.4 Observations

Camera observations In some cases, camera position as well as its orientation can be directly measured. They can be taken into account in the cost function by adding one observation which can be a camera position or an orientation (e.g. GPS/INS measurement can be directly included in the process). σ is then the expected error on GPS/INS measures. These camera observations can also be used in order to provide an initial solution for our minimization process (Section 3.).

Point observations For Ground Control Points (GCP), the 3D position (x, y, z) is an observation. The cost function is directly the distance between the observed position and the measured one. σ is the expected error on position measurement, depending on the measurement method (GPS, topography, ...).

For each point, the image positions (c, l) either in pixels or in millimeters are also observations. The cost function is the distance between the projection of the corresponding 3D point and the measured image position. σ is the expected error on image position measurement, depending on the method used to determine the point position (monocular, stereo, automatic, etc.).

Line observations In order to introduce linear features in the bundle adjustment process, we only need to introduce a distance between an observed and a computed value. Most algorithms represent image segments using their two extreme points l_1 and l_2 (see Figure 1).

The cost function is the distance between the projected 3D line and these extreme points, orthogonally to the projected 3D line. Using this definition of distance, segment extremities in different images do not have to match, they just have to be as close as possible from the 2D re projection of the same 3D line. It is therefore possible to create tie lines between images with no overlap. σ is the expected error on the position of segment extremities, orthogonally to the 3D line projection. Practically, this expected error is not easy to determine.

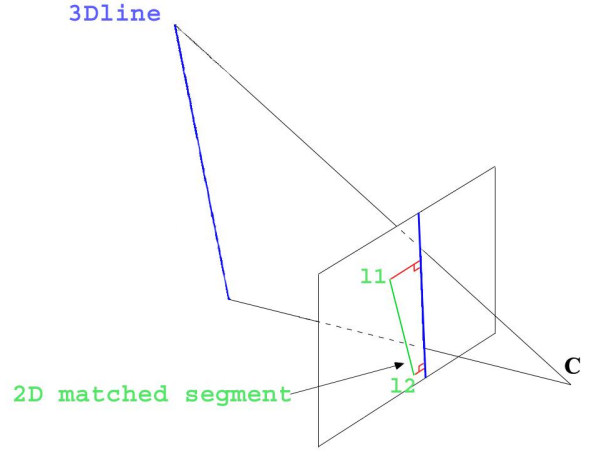


Figure 1: Linear features distance

3. MINIMIZATION

This section deals with the problem of minimizing the cost function F , which is obviously a non-linear function. It is achieved through the Gauss-Newton algorithm. Should robustness problems arise, our algorithm may be easily transformed in a Levenberg-Marquard iterative algorithm. Constraints will be addressed in subsection 3.3.

3.1 Gauss-Newton algorithm

The Gauss-Newton algorithm is an iterative algorithm requiring an initial estimate $\vec{X}^{(0)}$. In this paper, $\vec{X}^{(0)}$ is supposed to be available and will not be discussed (see Introduction). At each step k , vector $\vec{X}^{(k)}$ is updated : $\vec{X}^{(k+1)} = \vec{X}^{(k)} + \vec{d}\vec{X}$. Let $\mathbf{H}(\vec{X})$ be the Hessian matrix of function $F(\vec{X})$:

$$\mathbf{H}_{i,j}(\vec{X}) = \frac{\partial^2 F(\vec{X})}{\partial X_i \partial X_j} \text{ with } i, j \in [1, N_u] \quad (6)$$

and $\vec{g}(\vec{X})$ be the gradient of $F(\vec{X})$:

$$g_i(\vec{X}) = \frac{\partial F(\vec{X})}{\partial X_i} \text{ with } i \in [1, N_u] \quad (7)$$

At each step, we seek $\vec{X} + \vec{d}\vec{X}$ minimizing cost function F . Hence :

$$0 = \frac{dF(\vec{X} + \vec{d}\vec{X})}{dX} \approx \vec{g}(\vec{X}) + \mathbf{H}(\vec{X}) \vec{d}\vec{X} \quad (8)$$

which enables us to determine $\vec{d}\vec{X}$. The algorithm is :

- at each step k , compute $\vec{d}\vec{X}$ using:

$$\mathbf{H}(\vec{X}^{(k)}) \vec{d}\vec{X} = -\vec{g}(\vec{X}^{(k)}) \quad (9)$$

- $\vec{X}^{(k+1)} = \vec{X}^{(k)} + \vec{d}\vec{X}$
- $k = k + 1$
- go to step 1

3.2 Jacobian approximation

In most applications, $\mathbf{H}(\vec{X})$ and $\vec{g}(\vec{X})$ can be complex to compute, so a first order approximation is performed:

$$\mathbf{H}(\vec{X}) \approx \mathbf{J}^t(\vec{X})\mathbf{J}(\vec{X}) \quad (10)$$

where \mathbf{J} is the Jacobian matrix :

$$\mathbf{J}_{i,j}(\vec{X}) = \frac{\partial r_i(\vec{X})}{\partial X_j} = \frac{1}{\sigma_i} \frac{\partial v_i^{comp}(\vec{X})}{\partial X_j} \quad (11)$$

with $i \in [1, N_o], j \in [1, N_u]$

Using the same approximation, \vec{g} is computed using :

$$\vec{g}(\vec{X}) \approx \mathbf{J}^t(\vec{X})\vec{r}(\vec{X}) \quad (12)$$

This problem is equivalent to solving the linear least square problem :

$$\mathbf{J}(\vec{X}) \overrightarrow{dX} = \vec{r}(\vec{X}) \quad (13)$$

The system can be expressed using the usual matrix form :

$$[\mathbf{J}^t \mathbf{J}] \overrightarrow{dX} = [\mathbf{J}^t \vec{r}] \quad (14)$$

Seeing that the matrix is symmetric and positive, the system can be solved using the Cholevsky algorithm.

3.3 Constraints

At the beginning of this section, we did not take the constraints into account. So, even if at step k , $\vec{X}^{(k)}$ satisfies the constraints, $\vec{X}^{(k+1)} = \vec{X}^{(k)} + \overrightarrow{dX}$, with \overrightarrow{dX} solution of equation 14 has no reason to satisfy the constraints.

The first obvious solution to this problem is to modify each $\vec{X}^{(k+1)}$ in order to satisfy the constraints (by renormalization or projection on the constraint space). This is an easy and direct solution, but, the theory does not guarantee the convergence. In practice, the convergence is slow. An alternative solution is to introduce the constraints in the resolution process (see (Triggs et al., 2000)).

Equation 5 can be solved using Lagrange multipliers. Let $\vec{\lambda} \in \mathbb{R}^{N_c}$ be the vector of Lagrange multipliers. Let the first order expansion of the constraint vector be $\vec{c}(\vec{X} + \overrightarrow{dX}) \approx \vec{c}(\vec{X}) + \mathbf{C}\overrightarrow{dX}$, where \mathbf{C} is the matrix of the constraints gradients :

$$\mathbf{C}_{i,j} = \frac{\partial c_i}{\partial X_j} \quad \text{with } i \in [1, N_c], j \in [1, N_u] \quad (15)$$

Solving equation 5 is equivalent to finding $\vec{X} + \overrightarrow{dX}$ that optimizes $F + \vec{c} \cdot \vec{\lambda}$, subject to $\vec{c} = 0$. The first order expansion of both expressions are :

$$0 = \frac{d}{dX} \left(F(\vec{X} + \overrightarrow{dX}) + \vec{c}(\vec{X} + \overrightarrow{dX}) \cdot \vec{\lambda} \right) \approx \vec{g}(\vec{X}) + \mathbf{H}(\vec{X}) \overrightarrow{dX} + \mathbf{C}^t(\vec{X}) \vec{\lambda} \quad (16)$$

and

$$0 = \vec{c}(\vec{X} + \overrightarrow{dX}) \approx \vec{c}(\vec{X}) + \mathbf{C}\overrightarrow{dX} \quad (17)$$

Combining these two equations gives the expression for each step of the algorithm :

$$\begin{bmatrix} \mathbf{H} & \mathbf{C}^t \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \overrightarrow{dX} \\ \vec{\lambda} \end{bmatrix} = - \begin{bmatrix} \vec{g} \\ \vec{c} \end{bmatrix} \quad (18)$$

Using the same approximation as in section 3.2, the system becomes :

$$\begin{bmatrix} \mathbf{J}^t \mathbf{J} & \mathbf{C}^t \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \overrightarrow{dX} \\ \vec{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{J}^t \vec{r} \\ \vec{c} \end{bmatrix} \quad (19)$$

The main difference between this system and equation 14 is that the matrix is not positive anymore. So, it cannot be solved using the Cholevsky algorithm. Other algorithms such as LU, QR or Householder transform can be used. We generally use the LU transform.

4. ACCURACY

This section deals with error propagation in the bundle adjustment problem. It enables us to assess the accuracy of all reconstructed parameters.

4.1 Accuracy prevision

It is well known that, if observed values v^{obs} have only a Gaussian noise, vector \vec{X} which minimizes $F(\vec{X})$ is also the maximum likelihood vector. Let \mathbf{V} be the variance matrix :

$$\mathbf{V}_{i,j} = E \left[\left(\vec{X}_i - E(\vec{X}_i) \right) * \left(\vec{X}_j - E(\vec{X}_j) \right) \right] \quad (20)$$

with $i, j \in [1, N_u]$

where $E(x)$ denotes the expectation of variable x .

For \vec{X} minimum of F , the inverse of the Hessian matrix $\mathbf{H}^{-1}(\vec{X})$ is equal to the variance matrix \mathbf{V} . Some justifications can be found in (Hartley and Zisserman, 2002) or (Förstner, 2004).

So, it is possible to know the accuracy and the correlation between unknowns just by inverting the Hessian matrix. This results is confirmed by experimental results (see section 5).

A correlation matrix can also be computed :

$$\mathbf{Corr}_{i,j} = \frac{\mathbf{V}_{i,j}}{\sqrt{\mathbf{V}_{i,i} * \mathbf{V}_{j,j}}} \quad \text{with } i, j \in [1, N_u] \quad (21)$$

Each matrix element $\mathbf{Corr}_{i,j}$ gives the correlation (between -1 and +1) between two parameters X_i and X_j .

4.2 Camera position Accuracy

We denote by \hat{p} the estimation of the camera position after the bundle adjustment algorithm, and by \mathbf{V}_i , the 3×3 sub matrix of \mathbf{V} corresponding to variance-covariance matrix of the camera position is extracted. Error propagation follows:

$$P \sim \mathcal{N}(\hat{p}, \mathbf{V}_i) \quad (22)$$

Hence,

$$(P - \hat{p})^t \mathbf{V}_i^{-1} (P - \hat{p}) \sim \chi^2(3) \quad (23)$$

Now, given a significance number α , we can associate to the estimated camera position a covariance ellipsoid $\mathcal{E}(\hat{p}, \alpha)$:

$$(P - \hat{p})^t \mathbf{V}_i^{-1} (P - \hat{p}) \leq \chi^2(3, \alpha) \quad (24)$$

With $\chi^2(3, \alpha)$ representing the α^{th} quantile of a chi-square law with 3 degrees of freedom. To get the axes length of the ellipsoid, standard eigenvector decomposition of the matrix \mathbf{V}_i is performed.

4.3 Camera Rotation Accuracy

Rotation parameters are estimated using a normalized quaternion \hat{q} representation. As for the camera position, we can extract a variance covariance matrix from \mathbf{V} . This matrix \mathbf{V}_q will represent the variance covariance matrix of the quaternion. Because of the normalization constraint, \mathbf{V}_q does not provide any direct information on the uncertainty of the rotation. Hence, we will use a backward transport of the covariance (Hartley and Zisserman, 2002). We can define a rotation by an axis $u_{\phi, \psi}$ and an angle θ .

$$u_{\phi, \psi} = (\cos(\phi) \sin(\psi), \sin(\phi) \sin(\psi), \cos(\psi))^t \quad (25)$$

Let $Q : \mathbb{R}^3 \rightarrow \mathbb{R}^4$ be a differential mapping taking a parameter vector $\hat{r} = (\hat{\theta}, \hat{\phi}, \hat{\psi})$ to a measurement vector \hat{q} .

$$Q(\hat{\theta}, \hat{\phi}, \hat{\psi}) = (\cos \frac{\theta}{2}, \sin \frac{\theta}{2} u_{\phi, \psi}) \quad (26)$$

Hence, we can deduce an approximation of $\Sigma_{\hat{\theta}, \hat{\phi}, \hat{\psi}}$

$$\Sigma_{\hat{\theta}, \hat{\phi}, \hat{\psi}} = (\mathbf{J}_Q^t \mathbf{V}_q^{-1} \mathbf{J}_Q)^+ \quad (27)$$

With M^+ designing the pseudo-inverse of matrix M and \mathbf{J}_Q being the Jacobian of Q . Again, given a significance number α , we can associate a covariance ellipsoid to the estimated camera rotation $\mathcal{E}(\hat{\theta}, \hat{\phi}, \hat{\psi}, \alpha)$.

4.4 3D Points Accuracy

We use the same method as in the Camera position Accuracy section. Again, given a significance number α , we can associate to a tie point or a GCP a covariance ellipsoid $\mathcal{E}(\hat{p}, \alpha)$. One important application of this accuracy estimator is to predict roughly the accuracy of a 3D reconstruction.

4.5 3D Lines Accuracy

Line accuracy can be computed using our bundle adjustment technique. This part is still under development in our system. Some error propagation algorithms involving lines can be found in (Taillandier and Deriche, 2002) or (Heuel, 2001).

5. EXPERIMENTS

5.1 Test fields description

Calibration polygon The first test field contains 16 images of a calibration polygon with 75 materialized GCP with an accuracy of 0.1 mm (see Figure 2). The polygon was created in order to precisely determine camera internal parameters. Here, it is used to assess the quality of accuracy prevision.

Building facade The second test field is made of 16 images of a facade (see Figure 3) taken from a mobile vehicle using a stereo rig. It is representative of forthcoming image acquisitions from a vehicle. At a given instant t two cameras along a vertical baseline take pictures of building facades. A more accurate description of the acquisition process can be found in (Bentrah et al., 2004). Nine tie lines have been manually selected. Thirty GCP have been measured by a surveyor using classical topographic tools with an accuracy of 0.25mm.

5.2 Simulation

In order to verify accuracy previsions coming from the Hessian matrix inversion, two different experiments have been carried out on each test field. In each case, a perfect solution has been created. The first experiment (A) adds simply a Gaussian noise on all observed value (tie points, tie lines and GCP). The second one



Figure 2: Calibration polygon test field. 75 materialized GCP have been measured with an accuracy of 0.1 mm.

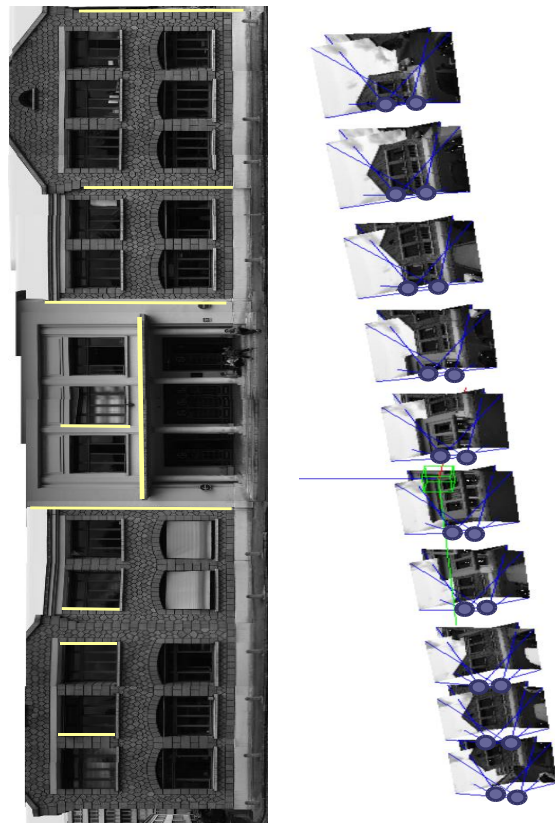


Figure 3: Building facade test field. A stereo rig mounted on a mobile vehicle has taken 16 images of a facade. 30 GCP are available. 8 vertical and one horizontal tie lines have been manually selected.

Experiment	GN I (px)	RE I (px)	GN G (m)	RE G (m)
A	1	0	0.001	0
B	1	1	0.001	0.001

Table 1: Experiments description.

Test field	V diagonal mean error	V mean error
Polygon A	0.026 %	0.51 %
Polygon B	0.027 %	0.71 %
Facade A	2.7 %	11.2 %
Facade B	2.7 %	11.3 %

Table 2: Stability of \mathbf{V} .

(B) introduces rounding errors, which are non Gaussian and are much more realistic. For each experiment, on each geometry, 1000 trials have been made.

Table 1 shows the different experiments. The legend for Table 1 is :

- GN I : Gaussian Noise on all Images observations, points and lines;
- RE I : Rounding Error on all Images observations, points and lines
- GN G : Gaussian Noise on Ground Control Points,
- RE G : Rounding Error on Ground Control Points.

Stability of accuracy prevision First, we have verified that the matrix \mathbf{V} is stable, i.e., that it does not change when data and solution change a little. Using all trials, we can compute a mean value and a standard deviation for each element of \mathbf{V} . Experimental results are shown in Table 2. \mathbf{V} diagonal mean error is the mean, for all elements of the diagonal of \mathbf{V} , of the standard deviation divided by the mean value of this element. \mathbf{V} mean error is the mean, for all elements of \mathbf{V} , of the standard deviation divided by the mean value of this element.

These results show that \mathbf{V} is very stable, and is not influenced much by usual noise. It seems that some simulations with lines are less stable than the one without lines. Anyway, for real cases, the matrix \mathbf{V} obtained after convergence can be used directly.

Accuracy previsions vs measured accuracy The goal of this part is to verify that the observed errors and the computed ones are coherent. Using all trials, it is possible to compute the variance and co-variance of all unknowns X_i , and to compare it with \mathbf{V} . The results are shown in Table 3. Legend for Table 3 is :

Test field	Mean SDE	Mean CE
Polygon A	0.016 %	0.025
Polygon B	0.037 %	0.025
Facade A	0.6 %	0.09
Facade B	0.5 %	0.17

Table 3: Accuracy simulation results.

- SDE : Standard Deviation Error is :

$$\frac{|\mathbf{V}_{i,i}^{obs} - \mathbf{V}_{i,i}|}{\mathbf{V}_{i,i}^{obs}} \text{ with } i \in [1, Nu] \quad (28)$$

where $\mathbf{V}_{i,i}^{obs}$ is the observed value of \mathbf{V} ;

- CE : Correlation Error is $|\mathbf{Corr}_{i,j}^{obs} - \mathbf{Corr}_{i,j}|$ for all unknowns ($i, j \in [1, Nu]$), where $\mathbf{Corr}_{i,j}^{obs}$ is the observed correlation between unknowns X_i and X_j

	Camera Position (mm)	Rotation (mrad)	3D Point (mm)
E_1 mean	39	3.8	2.7
sd	3	0.7	2.8e-02
E_2 mean	32	2.5	2.6
sd	2	0.6	2.8e-02

Table 4: Camera rotation and position accuracy with 30 GCP (line E_1) and with tie lines (line E_2).

	Camera Position (mm)	Rotation (mrad)	3D Point (mm)
E_1 mean	84	6	74
sd	12	2	45
E_2 mean	41	1	18
sd	9	0.2	8

Table 5: Camera rotation and position accuracy with 6 GCP and 24 tie points (line E_1) and with tie lines (line E_2).

Results are very promising : observed and computed variance are very similar. So the matrix \mathbf{H}^{-1} is a very good estimate of the variance matrix \mathbf{V} . The variance of any unknown can be predicted with an error inferior to 1%, even if noise is not perfectly gaussian. Linear features seem to degrade the quality of the estimation of correlation between parameters, especially when rounding error is applied (Facade B). Anyway, this method enables us to get the precision of each parameter of the system.

5.3 Mobile Vehicle

This real case study corresponds to the facade test field. In the first experiment (Table 4) all GCP have been used in the bundle adjustment algorithm. The second experiment (Table 5) analyzes the result of the bundle adjustment process with only 6 GCP (24 points are set as tie points in our process). For both experiments, significance number α is set to 0.95. Statistics are made on the length of the longest axis of each 3D ellipsoid.

The comparison of the angle between the longest ellipsoid axis and the Oz axis stresses the fact that using vertical tie lines provides a longest ellipsoid axis closer to the Oz axis by 2-5 degrees.

In the second experiment, one should note that all 24 control points, which are not used in the bundle, fall inside the estimated covariance ellipsoid.

5.4 Discussion

Simulation and real experiments have stressed the importance of lines in order to increase parameter estimation accuracy in a bundle adjustment process. We have observed that the gain in accuracy was depending on the orientation of the given lines. The error propagation modeling has also been validated by our experiments.

6. CONCLUSION AND FUTURE WORK

We have presented a technique to compute a bundle adjustment with points and lines. We provide an error propagation algorithm in order to assess the reliability of the estimated parameters (viewing position, viewing rotation, 3D points and lines). Experiments on simulated data and on real data are described. Many extensions and further research on this topic can be done:

- **Error propagation with non-Gaussian distributions.** Depending on the algorithm used to select points and lines the error distribution is non-Gaussian. Moreover, very often we must face a bundle adjustment problem with small data sets. In these situations, specific statistical tools have to be used in order to estimate parameter accuracy (large deviation, etc.)

- **Guidelines for a matching algorithm.** For some applications, we can suppose that the reliability of the viewing parameters and the 3D reconstruction is fixed. Hence, our bundle adjustment algorithm with error propagation can assess when the reliability goal is reached. If not, the matching algorithm still has to extract and match more features in order to increase the reliability in the given parameters.
- **Dealing with large scale surveys.** The experiments were performed with a reduced amount of data (20-30 images, several hundreds of measures). Dealing with large-scale surveys will necessitate an optimized design and we will have to handle sparse matrix. Some optimization processes are well known in the photogrammetric community and described in (Triggs et al., 2000).
- **Spatial reasoning.** A very promising area of research concerns the way to choose points and lines in images in order to improve the accuracy of the estimated parameters. As a matter of fact, the spatial distribution of the features (points and lines) inside the images, as well as the geometric arrangement of the camera, have a tremendous impact on the accuracy of the estimation of the different parameters of a bundle adjustment problem (Michaelsen and Stilla, 2003). Correlation analysis between different parameters will be of interest in order to estimate the crucial parameters.
- **Introducing more knowledge concerning the data.** More constraints or parameters can be introduced depending on the knowledge we have of the scene. For instance, vertical lines can be introduced in the bundle adjustment thanks to vanishing points or constraints on the 3D lines (Bentrah et al., 2004), (Van den Heuvel, 2001).

REFERENCES

- Bentrah, O., Paparoditis, N. and Pierrot-Deseilligny, M., 2004. Stereopolis: An image based urban environments modeling system. In: MMT 2004. The 4th International Symposium on Mobile Mapping Technology, Kunming, China.
- Chou, J. and Kamel, M., 1991. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *The International Journal of Robotics Research* 10(3), pp. 240–254.
- Faugeras, O., Luong, Q. and Papadopoulo, T., 2001. *The Geometry of Multiple Images*. The MIT Press.
- Förstner, W., 2004. Uncertainty and projective geometry. In: E. Bayro-Corrochano (ed.), *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics*, Springer.
- Habib, A., 1999. Aerial triangulation using point and linear features. In: *ISPRSGIS99*, pp. 137–142.
- Hartley, R. and Zisserman, A., 2002. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press. HAR r2 02:1 1.Ex.
- Heuel, S., 2001. Points, lines and planes and their optimal estimation. In: *Pattern Recognition, 23rd DAGM Symposium*, LNCS, Springer, pp. 92–99.
- Kraus, K., 1993. *Photogrammetry*. Dmmler/Bonn.
- Kumar, R. and Hanson, A., 1994. Robust methods for estimating pose and a sensitivity analysis. *CVGIP* 60(3), pp. 313–342.
- Michaelsen, E. and Stilla, U., 2003. Good sample consensus estimation of 2d-homographies for vehicle movement detection from thermal videos. In: R. S. International Archives of the Photogrammetry and S. I. Sciences (eds), *Proceedings of the ISPRS Conference on Photogrammetric Image Analysis PIA'03*, Munich, Vol. 34, Part 3/W8, pp. 125–130.
- Taillandier, F. and Deriche, R., 2002. Reconstruction of 3D linear primitives from multiple views for urban areas modelisation. In: *PCV'02*, Vol. 34:3B, pp. 267–272.
- Torr, P. H. S. and Davidson, C., 2003. Impsac: A synthesis of importance sampling and random sample census. *PAMI* 25(3), pp. 354–365.
- Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., 2000. Bundle adjustment – A modern synthesis. In: W. Triggs, A. Zisserman and R. Szeliski (eds), *Vision Algorithms: Theory and Practice*, LNCS, Springer Verlag, pp. 298–375.
- Van den Heuvel, F. A., 2001. Towards automatic relative orientation for architectural photogrammetry. In: *ISPRS Commission V, WG V/2*.
- Xu, G. and Zhang, Z., 1996. *Epipolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publisher.