Camera Uncertainty Computation in Large 3D Reconstruction

Michal Polic and Tomas Pajdla
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague
Jugoslávských partyzánů 3, 160 00 Prague 6, Czech Republic
policmic@fel.cvut.cz

***Abstract*—In computer vision, large scale Structure from Motion pipelines do not often evaluate the quality of the reconstruction by error propagation from measurements to the estimated parameters. It is a numerically sensitive and computationally challenging process, which is not easy to implement in practice for large scenes. We present a new algorithm that increases the numerical precision of the uncertainty propagation. It works with millions of feature points, thousands of cameras and millions of 3D points on a single computer. We provide an experimental comparison of our approach, as well as of previous approaches, on accurate ground truth and demonstrate that our algorithm is practical.**

***Keywords*-uncertainty propagation; numerical precision, 3D reconstruction**

I. Introduction

Recent work in Structure from Motion (SfM) has demonstrated a possibility of reconstructing geometry from large photo collections [2], [15]. Efficient non-linear refinement [4], [32] of camera and point parameters was developed to produce optimal reconstructions. However, practical computation of uncertainty [11] is mostly missing in the state of the art pipelines [27], [29], [31] because current algorithms for uncertainty propagation have cubic time and quadratic memory complexity.

If the uncertainty evaluation was fast enough, and with reasonable memory requirements, it could be used for filtering the most unconstrained cameras and 3D points before Bundle Adjustment which usually runs in each step of the SfM. That would make reconstruction pipelines faster and more robust. In addition, it would allow checking the uncertainty of added cameras and prevent wrong extensions of existing partial reconstructions, more sophisticated smoothing of reconstructed surfaces in dense reconstructions [19], [28], and better selection of the first reconstruction pair in sequential SfM.

Most often, the backward propagation of uncertainty is used [14]. It transports the covariance matrices of measurements to the covariance matrices of the parameters. The transport requires evaluating the Jacobian of the projection function, which projects the reconstruction parameters (i.e. camera rotation, position, focal length, radial distortion and points in 3D) to the measurements (observations) in images.

The projection function is often simplified by replacing it by its first order approximation, and the covariance matrices of the estimated parameters are then computed by inverting its information matrix [24]. Camera rotation angles and radial distortion parameters are usually much smaller than the coordinates of 3D points and also have much larger impact on the objective function, which is usually the sum of squared differences between the projected parameters and the measurements (reprojection errors) [14]. Therefore, the Jacobian of the objective function contains a wide range of values which are squared into the information matrix. This makes the "raw" information matrix numerically rank deficient for medium and larger image collections.

The state of the art algorithms usually do not produce correct results even for small reconstructions (i.e. 60 cameras and 250 points in 3D). Singular Value Decomposition (SVD) in double precision is not accurate enough for this task, Figure 7. Recent work [20], [24] uses a decomposition of the information matrix into blocks which simplifies and speeds-up its inversion.

The block related to 3D point parameters is, in general, a full rank block diagonal matrix, which can be easily inverted. The block related to camera parameters appears as numerically over-parameterized and the information matrix has, in general, seven degrees of freedom. This is due to the fact that the decomposition into the blocks does not fulfill the rank additivity condition [14], [30] (regardless the numerical over-parametrization, the sub-blocks are full rank, while their composition is rank deficient) and cannot be used.

The normal forms of covariance matrices [18] have to be computed using Moore-Penrose (M-P) pseudoinverse [8] of the information matrix, which is usually done by SVD [21]. The state of the art approach works well for problems with a few images and tens of points in 3D. As the number of images grows up to several thousand, the existing uncertainty computation algorithms fail to provide accurate results.

II. Contribution

We present the first approach to large scale covariance matrix computation which is practical. Our approach builds on top of the Taylor expansion (TE) idea [23]. However, we extend [23] by replacing the computation of the M-P pseudoinverse by estimating the inverse of the Schur complement of the matrix of the point parameters.

Secondly, we present an important experimental comparison of recent methods [18], [20] against Ground Truth (GT) covariance matrices, which we construct using more accurate arithmetics in Maple [1].

To calculate useful inversions, we have to fix the ambiguity (gauge freedom [18]) of the reconstruction and approximate the normal form of covariance matrices [18]. The inversion allows us to scale the information matrix and its decomposition to smaller blocks, which would not be possible with the M-P pseudoinversion.

We investigate different regularisation ideas that fix the reconstruction by projecting the parameters to a set of essential parameters and find out which parameters minimize the differences between the GT and the computed covariances of the camera parameters. Our approach is faster, more precise and much more stable than any previous one. The output of our work is publicly available as source code which can be used as an external library in nonlinear optimization pipelines, like Ceres Solver [3]. The code, test problems, and detailed experiments are available at cmp.felk.cvut.cz/∼policmic.

III. Previous work

The general principle of covariance propagation is well known. The forward/backward propagation for the linear/nonlinear system which is, or isn't, over-parameterized was described in [12], [14]. We are computing the backward transport (from measurements to the parameters) of the uncertainty for nonlinear over-parameterized systems (represented by projection functions). Our systems of equations are over-parameterized because reconstructions can be shifted, scaled and rotated without changing the objective function we optimize.

Hartley has shown [14] the propagation of uncertainty in an over-parameterized system in three steps: 1) mapping the space of the parameters to the set of the essential parameters, 2) finding the inverse of the information matrix of the essential parameters, and 3) mapping the inverse of information matrix back to the space of the original parameters. Kanatani presented a theory for describing the uncertainties under changing regularisation conditions (gauge transformations) [18]. There is a large number of choices of the regularisation conditions (e.g. fixing one camera rotation and pose and fixing one of the coordinates of another camera pose). Kanatani has also presented the gauge-free approach and defined the normal form of the covariance matrix. The normal form of the covariance matrix is computed as the M-P pseudoinverse of the information matrix. Pseudoinverse A^+ of A equals the inverse of A on the range of A and sends the orthogonal complement of the range of A to the zero vector [7]. Note that the residual vector is perpendicular to the range of A [12] and thus the pseudoinverse minimizes the sum of the squared Mahalanobis distances [14] of the residuals to the zero vector.

Lhuillier and Perriollat [20] decomposed the information matrix and computed the M-P pseudoinverse of the Schur complement [33] of the submatrix of 3D point parameters. This submatrix has the same size as the block of camera parameters. It is much smaller than the information matrix since reconstructions usually contain much fewer cameras than points in 3D. The decomposition was extended in recent articles [23], [24]. However, the decomposition does not satisfy the rank additivity condition [30]. Lhuillier has published a proof [20] of the existence of a correction term allowing us to use the decomposition. However, there is no straightforward connection between the proof and the correction term actually used in [20].

There are many specific extensions for computation of the uncertainty of lines [5], edges [6], laser scans [16], [26], and stereo setups [22], [25]. The authors tried to approximate covariances of specific setups using heuristics instead of following the general uncertainty propagation method. No other work was compared against GT (i.e. the normal form of the uncertainty matrix [18]) computed with an accurate representation of numbers.

IV. Problem formulation

The problem formulation extends the previous paper [23]. We consider a setup with n cameras $C = \{C_1, C_2, \ldots, C_n\}$ where $C_i \in \mathbb{R}^p$ is p-dimensional vector of camera parameters, m points $X = \{X_1, X_2, \ldots, X_m\}$ in 3D and k image observations represented by vector $\boldsymbol{u} \in \mathbb{R}^{2k}$. Each observation $\boldsymbol{u}_{i,j} \in \boldsymbol{u}$, i.e. an image point, is a projection of 3D point X_j by camera C_i, using projection function $\boldsymbol{p}(C_i, X_j)$. Noise $\epsilon_{i,j}$ has zero mean, $E(\epsilon_{ij}) = 0$, and standard deviation $V(\epsilon_{i,j})$. All pairs of indices (i, j) are in an index set S that determines which point X_j is visible in which camera view C_i.

$$\boldsymbol{u}_{i,j} = \boldsymbol{p}(C_i, X_j) + \epsilon_{i,j} \qquad \forall (i,j) \in S \tag{1}$$

The vector θ equals $[C_1, \ldots, C_n, X_1, \ldots, X_m]$ and ϵ is the random noise vector composed from all $\epsilon_{i,j}$ where $(i, j) \in S$. Function f is composed of projection functions $\boldsymbol{p}$. It projects vector θ into the image observations

$$\boldsymbol{u} = f(\theta) + \epsilon \tag{2}$$

The function (2) leads to a non-linear least squares optimization

$$\theta^* = \arg\min_{\theta} \|\boldsymbol{u} - f(\theta)\|_2^2 \tag{3}$$

minimizing the objective (residual) function which is the sum of squares of the differences $r(\theta) = \boldsymbol{u} - f(\theta)$ between observations $\boldsymbol{u}$ and reprojections $f(\theta)$.

The standard way to solve the nonlinear differentiable system is to use its first order approximation

$$r(\theta + \delta\theta) \approx r(\theta) + J(\theta)\,\delta\theta \tag{4}$$

One step $\delta\theta$ towards the solution θ^* is obtained by solving

$$J^\top J\,\delta\theta = -J^\top r \tag{5}$$

with $J(\theta)$ and $r(\theta)$, which we replace by J and r for brevity. Using Gauss-Markoff theorem [12], the covariance $\hat{V}_a(\theta)$ is computed as

$$\hat{V}_a(\theta) = \sigma_a^2 (J^\top \hat{V}(\boldsymbol{u}) J)^{-1} \tag{6}$$

for affine transformations. $\hat{V}(\boldsymbol{u}) \in \mathbb{R}^{2k\times 2k}$ is a block diagonal matrix with k measurement covariances on the diagonal. The variance factor σ_a is estimated from the residuals as $\sigma_a^2 \approx \|r\|^2/(2k-pn)$. In case of redundant set of the parameters, Equation 6 does not hold because $J^\top \hat{V}(\boldsymbol{u}) J$ is rank deficient.

A reconstruction by SfM usually contains 7 degrees of freedom [18] because the scene can be moved, rotated and scaled without change of residual function $r(\theta)$. To invert the information matrix, the space of the parameters has to be projected to a manifold such that there is a one-to-one mapping from observations to parameters [14], i.e. there is no ambiguity in parameters after all observations have been taken into account.

Such a projection can be realized by a *regularisation matrix* $R \in \mathbb{R}^{pn\times pn-7}$ which can, e.g., be constructed as a composition $R = R_p\,R_s$ of a *projection matrix* R_p [14], [18] and a *scaling matrix* R_s, which we introduce here. Using R, we can rewrite Equation 6 as

$$\hat{V}(\theta) = \sigma^2 R (R^\top J^\top \hat{V}(\boldsymbol{u}) J R)^{-1} R^\top \tag{7}$$

We investigate which regularisation minimizes the differences in comparison with GT covariance matrices in Section VI. The $\hat{V}(\theta)$ is the covariance matrix of all reconstruction parameters and an unbiased variance factor σ^2 equals $\|r\|^2/(2k-pn-7)$. If the content of the Jacobian is permuted to have cameras followed by points, i.e. $J = [J_C\; J_X]$, the *information matrix*

$$Q = R^\top J^\top \hat{V}(\boldsymbol{u}) J R = \begin{bmatrix} U & W \\ W^\top & V \end{bmatrix} \tag{8}$$

will be sparse with block diagonal matrices U and V, see Figure 1. To compute the inverse of information matrix, we introduce $Y = -V^{-1}W^\top$. We note, first, that V is composed of 3×3 blocks on the diagonal and its inverse can be computed separately for each block, and then also that forming Y should be fast thanks to the sparsity of V and W. The Upper triangular–Diagonal–Lower triangular (UDL) decomposition of the block matrix Q leads to

$$\hat{V}(\theta) = \sigma^2 R \left(\begin{bmatrix} I & -Y^\top \\ 0 & I \end{bmatrix} \begin{bmatrix} Z & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I & 0 \\ -Y & I \end{bmatrix} \right)^{-1} R^\top \tag{9}$$

where matrix Z is the Schur complement [33] of the block V of the information matrix

$$Z = U + WY \tag{10}$$

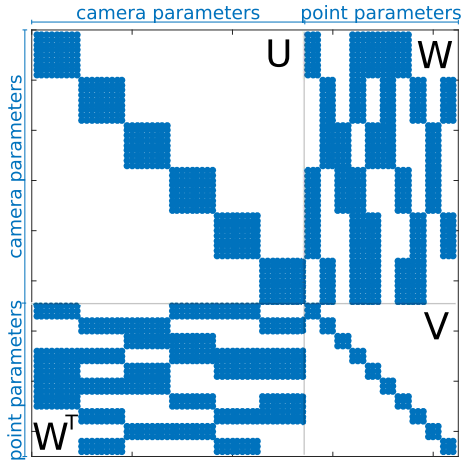

Figure 1: The structure of the information matrix for Cube dataset. Published in [23]

We are not interested in off-diagonal blocks. All covariances of reconstruction parameters are in the blocks on the diagonal of the dense matrix $\hat{V}(\theta)$. The interesting sub-matrices can be computed as

$$\hat{V}(\theta) = \sigma^2 R \begin{bmatrix} Z^{-1} & - \\ - & Y Z^{-1} Y^\top + V^{-1} \end{bmatrix} R^\top \tag{11}$$

The blocks of size $\mathbb{R}^{p\times p}$ on the diagonal Z^{-1} are covariances of camera parameters. The blocks of size $\mathbb{R}^{3\times 3}$ on the diagonal of sub-matrix $YZ^{-1}Y^\top + V^{-1}$ are covariances of point parameters.

V. The Taylor Expansion Algorithm

We estimate the inversion of Z instead of M-P inversion [23] of Z. First, we define $M = J^\top \hat{V}(\boldsymbol{u}) J$. Using the inversion instead of M-P inversion, allows us employ scaling

$$(R^\top M R)^+ \neq R^+ M^+ R^{+\top} \tag{12}$$

$$(R^\top M R)^{-1} = R^{-1} M^{-1} R^{-\top} \tag{13}$$

and LDU decomposition of Q to smaller blocks. To solve the TE inversion, we introduce function

$$g(\lambda) = (Z + \lambda I)^{-1} \tag{14}$$

where I is the identity matrix scaled by a scalar λ. Error produced by *damping term* λI is removed by TE of function $g(\lambda)$ in point 0. The general i-th derivative of function g with respect to λ is

$$\frac{d^i g}{d\lambda^i}(\lambda) = (-1)^i i (Z + \lambda I)^{-(i+1)} \tag{15}$$

We assigned the derivatives to the Taylor series estimated in zero point

$$\sum_{i=0}^{\infty} \left(\frac{(-\lambda)^i}{i!} \frac{d^i g}{d\lambda^i}(\lambda) \right) \tag{16}$$

and express the inversion of matrix Z

$$g(0) = (Z+\lambda I)^{-1} + \sum_{t=1}^{\infty} \left(\frac{\lambda^t}{(t-1)!} (Z+\lambda I)^{-(t+1)} \right) \quad (17)$$

The λI term allows us to compute the inversion of Z approximately for numerically rank deficient matrices and improves the numerical precision of inversion computation for large reconstructions.

VI. Regularization of the Jacobian

The regularisation matrix R combines projection R_p to the submanifold where the inversion can be computed and scaling R_s

$$R = R_p \, R_s \quad (18)$$

The matrix R_p fixes the ambiguity of the reconstruction. The inversion using the TE approach can be done with $R_p = I$ because we have infinitely differentiable function $r(\theta)$ and we can follow Taylor series to approximate the inversion function. However, the numerical precision of doubles, represented by 15 significant digits, causes that the results are less precise than in the case of appropriate projection to the submanifold of reconstruction parameters. There are different ways how to construct projections R_p in [14], [18], [20], which can be split into two groups: the trivial gauges (TG) and the nontrivial symmetric gauges (NSG). We will start shortly with NSG and then focus more to the TG.

To use NSG, we have to assume Gauss-Helmert model instead of Gauss-Markov model for redundant observations [12] and deal with measurements as parameters. Thus the objective (residual) function would be

$$r(\theta, \boldsymbol{u}) = \boldsymbol{u} - f(\theta) \quad (19)$$

with additional conditions represented by function $h(\theta)$ and the derivatives

$$A = \frac{\partial r(\theta, \boldsymbol{u})}{\partial \theta}, \quad B = \frac{\partial r(\theta, \boldsymbol{u})}{\partial \boldsymbol{u}}, \quad H = \frac{\partial h(\theta)}{\partial \theta} \quad (20)$$

and covariance matrix $\hat{V}(\theta)$ computed by

$$\begin{bmatrix} \hat{V}(\theta) & N \\ N^\top & P \end{bmatrix} = \begin{bmatrix} A^\top (B^\top V(\boldsymbol{u}) B)^{-1} A & H \\ H^\top & 0 \end{bmatrix}^{-1} \quad (21)$$

The covariance $\hat{V}(\theta)$ can be computed using sparse inversion, however the inverted matrix in 21 is much larger than inversion of Z and we can't use TE to improve numerical precision.

The NSG conditions usually fix some statistical properties of estimated centers and orientations of a subset of cameras or estimates of some 3D points to fix the global shift (3 parameters), orientation (3 parameters) and the scale (1 parameter) of the reconstruction.

The TG fix directly cameras and 3D points instead of their mean, covariance and scale. If we fix one camera rotation, pose and one of the coordinates of another camera pose, we lose the information about their uncertainties. Note, that we can not rely on the numerical precision of the uncertainty of points in 3D and when we fix some of them we don't loose any useful information. We empirically found out that most similar uncertainties w.r.t. GT are produced by the fixation of the three most distant points X_a, X_b, X_c. We seek for this triplet of points using RANSAC [10]. A triple of points fixes 9 instead of 7 parameters however we empirically found out that it produces more precise results than fixing two and one third of 3D point or any fixation of one or more cameras.

The matrix R_p is realized as the partial derivation of the function $h(\theta)$ w.r.t. points X_a, X_b, X_c. The function h projects parameters $\theta \backslash \{X_a, X_b, X_c\}$ to θ. Thus, the multiplication $J\,R_p$ removes the columns of the Jacobian J which correspond to the partial derivatives of function $r(\theta)$ w.r.t. points X_a, X_b, X_c.

The scale R_s of the Jacobian J is the diagonal matrix

$$R_{s(i,j)} = 1/\left\|J_j\right\| \quad for: \; i = j; \quad (22)$$

$$R_{s(i,j)} = 0 \quad for: \; i \neq j \quad (23)$$

where J_j represents j-th column of J. The scaled Jacobian has a similar range of values in each column and M has unit values at the diagonal.

VII. Experimental evaluation

The experiments are structured into four parts: the computation of the GT covariance matrices, the description of the datasets, the evaluation of the precision, and the comparison of the speed of the algorithms.

A. Computation of Ground Truth covariance matrices

We use the theory of gauge-free approach which leads to M-P pseudoinversion, described in [18]. We decompose the matrix Z using the SVD into

$$Z = \bar{U} \bar{S} \bar{V}^\top \quad (24)$$

and invert the diagonal values $\bar{S}'_{i,i} = 1/\bar{S}_{i,i}$ for $i \in 1, 2, \ldots, n\,p - 7$ because the reconstruction has 7 degrees of freedom. The remaining values on the diagonal of $\bar{S}'$ are set to zero. The inversion of Z is than obtained as

$$Z^+ = \bar{U} \bar{S}' \bar{V}^\top \quad (25)$$

The SVD algorithm is sensitive to rounding when the range of values in the matrix Z is large and different implementations may lead to different results (i.e. Maple, Matlab and Ceres which can be seen in Figure 7). All implementations, except for Maple, use the double precision, represented by 15 significant digits. To achieve more accurate results, we evaluated the GT covariance matrices in Maple using 100 significant digits. The accurate evaluation of the uncertainty matrix is computationally demanding (e.g. the SVD of Z for Daliborka dataset took approximately 22hours). Therefore,

Table I: This table summarize the number of cameras N_{Cams}, the number of points N_{Pts} and the number of observations N_{Obs} for the reconstructions which were created: 1,3 synthetically, 4-9 by Bundler [29] and 2, 10 by COLMAP [27]

#	Dataset	N_{Cams}	N_{Pts}	N_{Obs}
1.	Cube	6	15	60
2.	Toy	10	60	200
3.	Flat	30	100	1033
4.	Daliborka	64	200	5205
5	Marianska	118	80 873	248 511
6	Sagrada Familia	199	75 166	633 477
7	Dolnoslaskie	360	529 829	226 0026
8	Tower of London	530	65 768	508 579
9	Notre Dame	715	127 431	748 003
10	Seychelles	1400	407 193	2 098 201

Table II: Compared algorithms

#	Algorithm
1.	SVD of M using Maple (Kanatani [18]) (GT)
2.	TE inversion of scaled Z with three points fix
3.	SVD of M using Ceres (Kanatani [18])
4.	TE inversion of scaled Z with trivial camera fix
5.	SVD of Z with correction term (Lhuillier [20])
6.	SVD of M using Matlab (Kanatani [18])
7.	M-P inverse of Z using TE (Polic [23])

we computed GT covariance matrices only for the datasets 1-4, see Table I.

B. Datasets

We experiment with realistic synthetic reconstructions, as well as with middle to large scale Internet datasets. The parameters of the datasets are summarized in Table I. The datasets 2, 4 were reconstructed by publicly available pipelines (COLMAP [27], Bundler [29]) and after that, the number of the points in 3D was reduced to allow computing GT covariance matrices.

C. Precision

We compared the algorithms summarized in Table II. The algorithm 1 uses Maple computation with 100 significant digits and its result is considered the Ground Truth (GT). The algorithm 3 uses Ceres [3], algorithm 2 uses C++ libraries while other algorithms use Matlab [1].

Figure 7 shows the uncertainties. The order of rows corresponds to the order of algorithms in Table II (i.e. the first row corresponds to the SVD of M using Maple). You can see that the correct gauge-free approach [18] on rows 3,6 do not produce correct covariance matrices even for small reconstructions because of the numerical rank deficiency of the information matrix. These algorithms usually ignore the most unconstrained cameras or completely fail, see algorithm 6 for Daliborka dataset. This problem was not solved by any of previous approaches [14], [20], [23] which are on rows 4,5,7. The algorithm 4 is an improved version of [14]. It scales Jacobian by suitably choosing R_p, see Section VI. You can see that the Lhuillier algorithm [20] (the fifth row) also ignores the most unconstrained cameras even for small scenes. The covariances of the camera centers are not shown when containing complex, not a number or infinite values (e.g. for the algorithm 7, Figure 7). Our algorithm, TE inversion, has the opposite trend, i.e. the error decreases with the growing size of the reconstruction. Figure 6 shows the distribution of all errors (with the corresponding color coding) for the experiments in shown Figure 7.

The inversion of $Z+\lambda I$ usually stable for $\lambda = 0$ for small reconstructions, however for the large ones may be very unstable. The algorithms 2, 4, 7 use a damping term. The dependence of the mean error $\overline{err}(\hat{V}(\theta,\lambda))$ of the estimated covariance $\hat{V}(\theta,\lambda)$ for scene parameters θ on parameter λ is shown in Figure 2. Error function $\overline{err}(\hat{V})$ is computed as the mean of the Frobenius norm of elements of $\hat{V}_i - \hat{V}_{GT}$, which correspond to camera orientations and centers. It has been observed that the errors in covariances of camera extrinsic parameters are sufficient for finding suitable values of λ.

Figure 3 shows (red dashed line) the decreasing trend of the mean error $\overline{err}(\widetilde{V}(\theta,\hat{\lambda},np))$ (where $\widetilde{V}(\theta,\hat{\lambda},np)$ is estimated covariance for scene parameters θ and given $\hat{\lambda}$ depend on the number of camera parameters np) with increasing reconstruction size (i.e. the size of inverted matrix Z). The Figure 3 also shows (solid lines) the error

$$\widetilde{err}(ZZ^{-1}) = \sum_{1}^{np} \frac{1}{10^4 np} \left(\Sigma_{k=1}^{10^4} (Z\,Z^{-1} - I)\boldsymbol{x}_k \right) \quad (26)$$

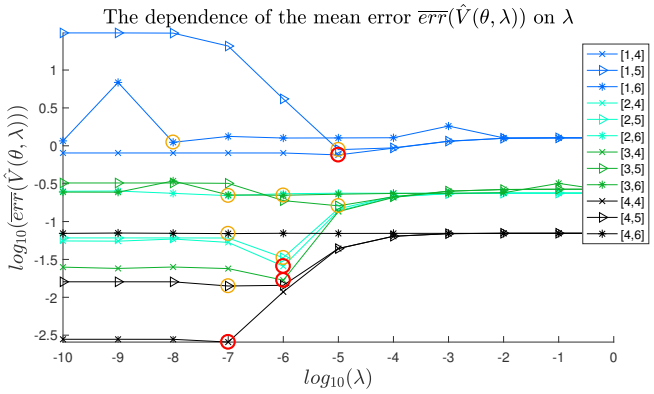

Figure 2: Each point represents the mean of errors (described precisely in Section VII-C) of the uncertainty matrices for one [dataset,algorithm] and given damping term λ. The lambdas chosen by our algorithm are shown as red circles and the lambdas chosen by other algorithms are shown as orange circles.

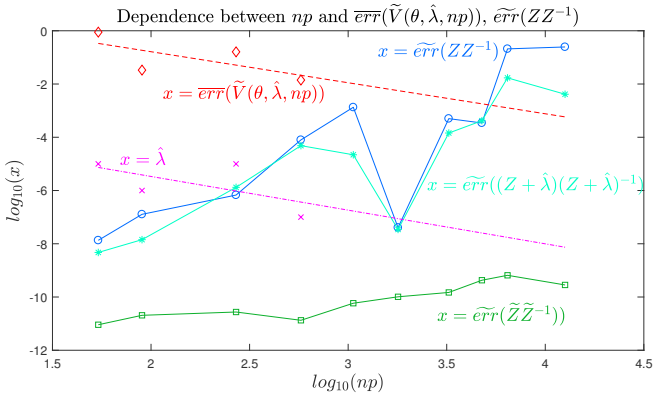

Figure 3: Each point (except $\hat{\lambda}$, the empirically selected value of λ) represents the mean of errors (described precisely in Section VII-C) of the uncertainty matrices for TE inversion algorithm, one dataset and $\hat{\lambda}$ based on number of camera parameters $n\,p$

of inversion ZZ^{-1} where $\boldsymbol{x} \in \mathbb{R}^{np}$ is a random vector with the zero mean and unit standard deviation. The matrix $Z \in \mathbb{R}^{np\times np}$ is either random matrix $\widetilde{Z}_{i,j} \in [0,1]$ or the Schur complement matrix Z (Equation 10) with and without using the damping term $\hat{\lambda}$ for computing Z^{-1}. It can be seen that the dumping term decreases the error for large datasets (i.e. datasets 9, 10). The best linear prediction $\hat{\lambda}$ of λ from the number of cameras (i.e. the size of inverted matrix Z) has been found as follows

$$\hat{\lambda} = 10^{-1.2653\ \log_{10}(n)-2.9415} \tag{27}$$

Finally, Figure 4 shows that error $\overline{err}(\overline{V}(\theta, X_a, X_b, X_c))$ decreases with increasing sum of distances between fixed points X_a, X_b, X_c. $\overline{V}(\theta, X_a, X_b, X_c)$ is the estimated covariance matrix for chosen triple of points. Moreover, the influence the choice of the fixed points on the covariance computation decreases with increasing size of the reconstructed scene. Thus, we can fix any three

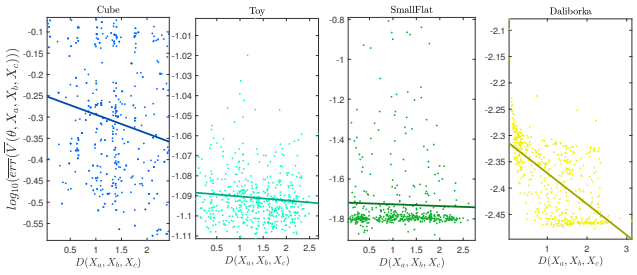

Figure 4: Each subplot represents one dataset (i.e. 1-4) and each point represents the mean of errors of uncertainty matrices for selected dataset and different sets of fixed points $\{X_a, X_b, X_c\}$. The function D is the sum of all points distances, i.e. $D(X_a, X_b, X_c) = \|X_a - X_b\| + \|X_a - X_c\| + \|X_b - X_c\|$.

mutually distant points for large datasets (e.g. datasets 9, 10).

D. Speed

The covariance matrix using M-P pseudoinversion of M for Daliborka dataset (i.e. for 1176 reconstruction parameters) was computed using Matlab in 0.45sec, using Ceres (via Eigen 3.3 [13]) in 25.9min. Our algorithm (TE inversion) was computed for Daliborka in Matlab in 0.67sec and using Intel MKL (C++ code) in 0.35sec. Further, the first middle sized reconstruction Marianska without reduction of 3D points has 243681 reconstruction parameters and requires about 470GB for dense representation of matrix M. Thus, we can't use current Ceres algorithm nor the algorithm 5. Secondly, the evaluation using SVD has cubic asymptotic complexity in the number of the parameters and the uncertainty evaluation for Marianska dataset would take approximately 9 million times the time of Daliborka evaluation. The TE inversion algorithm was computed for Marianska in 4.32sec from which the sparse matrix-matrix multiplication (SMMM) took 3.26sec. The SMMM, used for building the matrix M and Z, was performed by Eigen 3.3 which means that the speed can be further improved using the structure of the matrices or more enhanced algorithm [9], [17].

The state of the art methods are neither precise enough nor allow the computation for real middle sized datasets. We summarized the processing times of the three most important algorithms in Figure 5. The algorithm 1 was much slower (i.e. 22 hours for Daliborka) than all other algorithms. The algorithms 4 and 7 take the same time as algorithm 2 and the algorithms 3 and 5 can't be evaluated on datasets 5-10 due to the time and memory requirements. All experiments were performed on a single computer with one 2.6GHz Intel Core i7-6700HQ with 32GB RAM running a 64-bit Windows 10

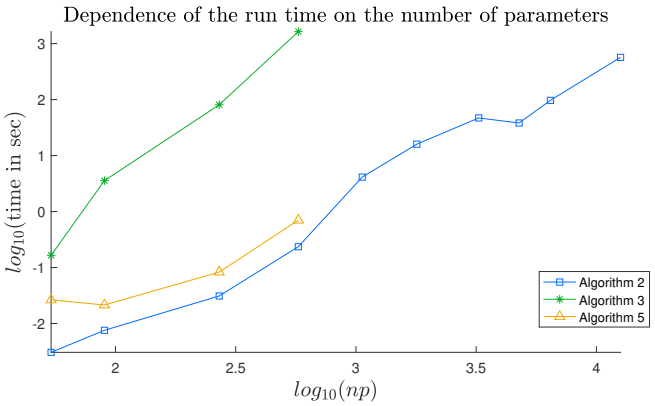

Figure 5: The dependency between the number of camera parameters of the reconstruction (equal the dimension of Z) and the run time of the algorithm. Each point represents evaluation of one dataset from Table I

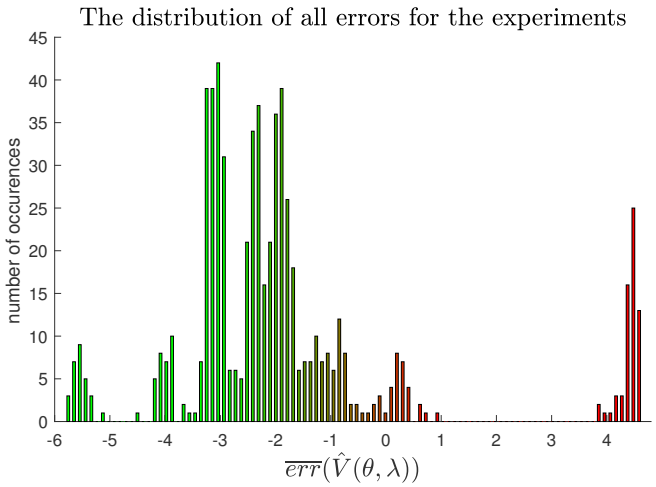

Figure 6: The distribution of all errors (with the corresponding color coding) for the experiments in Figure 7

operating system.

VIII. Conclusion

Current methods for evaluating the quality [14], [18], [20] of the reconstruction by error propagation from measurements to the estimated parameters are based on Moore-Penrose pseudoinversion (i.e. Singular Value Decomposition [21]) which is computationally challenging and mostly imprecise for real datasets because of a wide range of values in the information matrix. We propose a new method which computes yet another approximation of the inversion instead of the pseudoinversion [23] of the information matrix. That allows the scaling of the values of the information matrix and produces more precise results. We showed that other methods using the standard approaches to computing the covariance matrix work well for datasets with a few cameras and tens of points in 3D. Our method works for much larger reconstructions (e.g. a reconstruction with 1400 cameras, 407193 points in 3D and 2098201 observations in reasonable time 10min) on a single computer. We hope that the advanced sparse multiplication algorithms, sparse inverse algorithms and the usage of the GPU can speed-up our method about an order of magnitude and allow us filtering of the most unconstrained cameras and 3D points in SfM pipelines. The additional analysis would also lead to more precise evaluation of the uncertainty of the points in 3D and a sophisticated smoothing of the reconstructed surface.

Acknowledgment

This work was supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468), the EU-H2020 project LADIO (number 731970) and Grant Agency of the CTU Prague project SGS16/230/OHK3/3T/13.

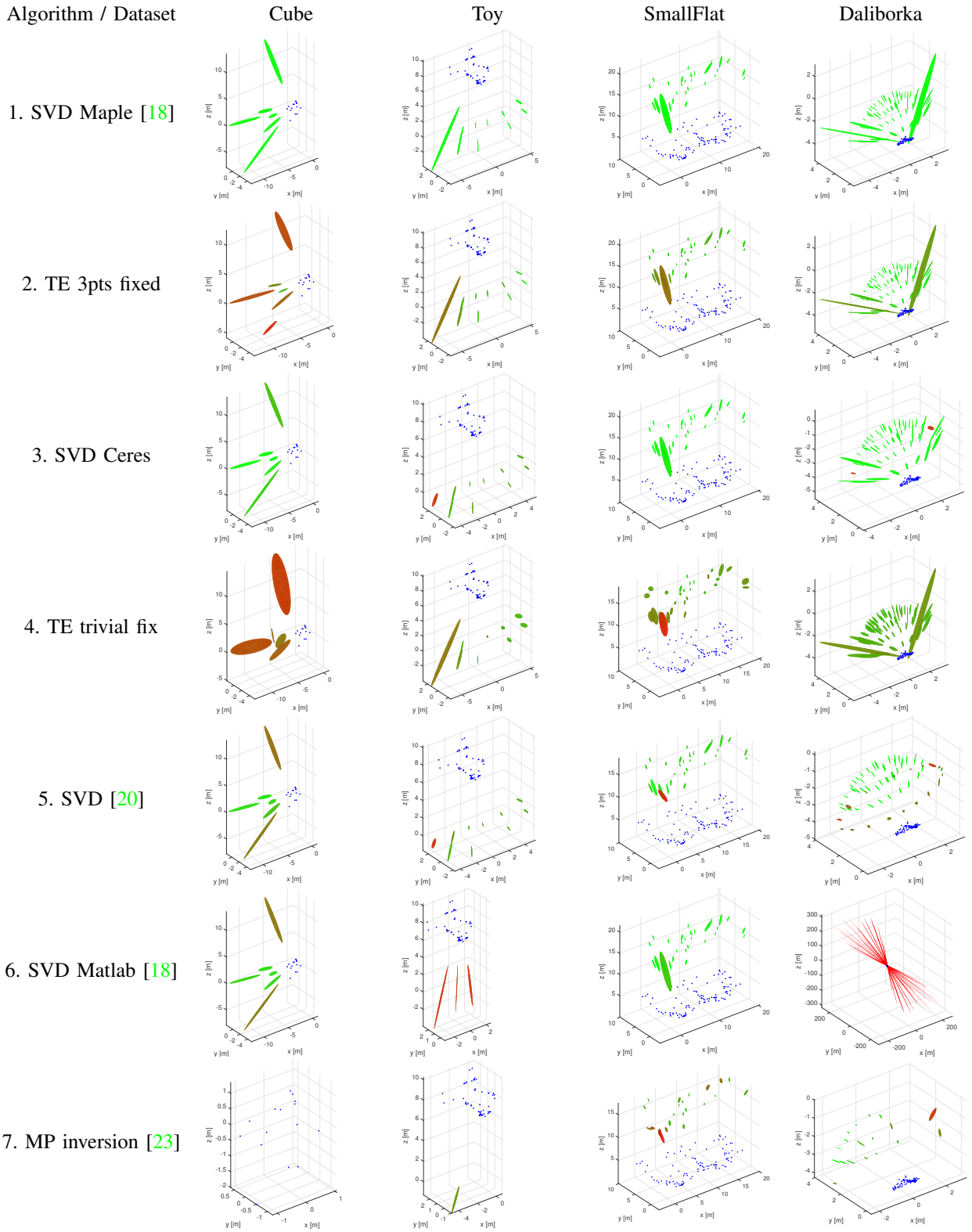

Figure 7: The comparison of the algorithms from Table II for the datasets 1-4 from Table I

References

[1] Maple 2016. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. http://matlab.com. 2, 5

[2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1

[3] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 2, 5

[4] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pages 29–42. Springer, 2010. 1

[5] R. Balasubramanian, S. Das, and K. Swaminathan. Error analysis in reconstruction of a line in 3-d from two arbitrary perspective views. *International journal of computer mathematics*, 78(2):191–212, 2001. 2

[6] A. Belhaoua, S. Kohler, and E. Hirsch. Error evaluation in a stereovision-based 3d reconstruction system. *EURASIP Journal on Image and Video Processing*, 2010(1):1, 2010. 2

[7] A. Ben-Israel and T. N. Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003. 2

[8] A. Björck. *Numerical methods for least squares problems*. Siam, 1996. 1

[9] A. Buluç and J. R. Gilbert. Parallel sparse matrix-matrix multiplication and indexing: Implementation and experiments. *SIAM Journal on Scientific Computing*, 34(4):C170–C191, 2012. 6

[10] 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. 4

[11] W. Förstner. Uncertainty and projective geometry. In *Handbook of Geometric Computing*, pages 493–534. Springer, 2005. 1

[12] W. Förstner and B. P. Wrobel. *Photogrammetric Computer Vision*. Springer, 2016. 2, 3, 4

[13] G. Guennebaud, B. Jacob, et al. Eigen v3.3. http://eigen.tuxfamily.org, 2010. 6

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2, 3, 4, 5, 7

[15] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[16] J. Höhle and M. Höhle. Accuracy assessment of digital elevation models by means of robust statistical methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(4):398–406, 2009. 2

[17] S. Jain-Mendon and R. Sass. Performance evaluation of sparse matrix-matrix multiplication. In *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, pages 1–4. IEEE, 2013. 6

[18] K.-i. Kanatani and D. D. Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5):2017–2028, 2001. 1, 2, 3, 4, 5, 7, 8

[19] F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. Shading-aware multi-view stereo. In *European Conference on Computer Vision*, pages 469–485. Springer, 2016. 1

[20] M. Lhuillier and M. Perriollat. Uncertainty ellipsoids calculations for complex 3d reconstructions. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3062–3069. IEEE, 2006. 1, 2, 4, 5, 7, 8

[21] N. Muller, L. Magaia, and B. M. Herbst. Singular value decomposition, eigenfaces, and 3d reconstructions. *SIAM review*, 46(3):518–545, 2004. 1, 7

[22] S.-Y. Park and M. Subbarao. A multiview 3d modeling system based on stereo vision techniques. *Machine Vision and Applications*, 16(3):148–156, 2005. 2

[23] M. Polic and T. Pajdla. Uncertainty computation in large 3d reconstruction. In *Scandinavian Conference on Image Analysis*, pages 110–121. Springer, 2017. 1, 2, 3, 5, 7, 8

[24] L. Polok, V. Ila, and P. Smrz. 3d reconstruction quality analysis and its acceleration on gpu clusters. 2016. 1, 2

[25] A. H. Rivera-Rios, F.-L. Shih, and M. Marefat. Stereo camera pose determination with error reduction and tolerance satisfaction for dimensional measurements. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 423–428. IEEE, 2005. 2

[26] P. Schaer, J. Skaloud, S. Landtwing, and K. Legat. Accuracy estimation for laser point cloud including scanning geometry. In *Mobile Mapping Symposium 2007, Padova*, number TOPO-CONF-2008-015, 2007. 2

[27] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5

[28] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016. 1

[29] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 1, 5

[30] Y. Tian. The moore-penrose inverses of m× n block matrices and their applications. *Linear algebra and its applications*, 283(1):35–60, 1998. 1, 2

[31] C. Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 1

[32] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3057–3064. IEEE, 2011. 1

[33] F. Zhang. *The Schur Complement and Its Applications*. Springer US, 2005. 2, 3