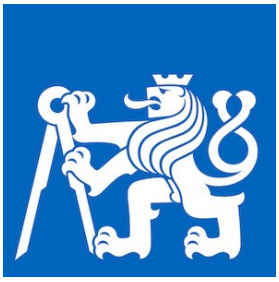
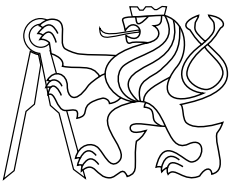
CENTER FOR
MACHINE PERCEPTION



CZECH INSTITUTE
OF INFORMATICS
ROBOTICS AND
CYBERNETICS



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

RESEARCH REPORT

ISSN 1213-2365

3D Scene Analysis

PhD Thesis Proposal

Michal Polic

policmic@fel.cvut.cz

CTU–CMP–2017–24

September 5, 2017

Available at
ftp://cmp.felk.cvut.cz/~policmic/articles/Polic-TP-2017-24.pdf

Supervisor: Ing. Tomáš Pajdla, Ph.D.

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) and Grant Agency of the CTU Prague project SGS16/230/OHK3/3T/13.

Research Reports of CMP, Czech Technical University in Prague, No. 24, 2017

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University
Jugoslávských partyzánů 3, 160 00 Prague 6, Czech Republic
phone: +420 2 2435 4139, www: http://cmp.felk.cvut.cz

Abstract

A three-dimensional scene is an output of many computer vision algorithms (e.g. Structure from Motion, Simultaneous localization and mapping, Multi-View Stereo) due to the wide range of applications in industry (e.g. robot navigation, self-driving cars) and entertainment (e.g. virtual reality). There are hundreds of papers reviewed the maximum likelihood estimate of the scene parameters. However, more enhanced statistics are rarely considered. In this work, we focus on describing the second moment of the image points detection error and its propagation to the scene parameters. First, we describe common steps of the reconstruction process and show its advantages and drawbacks. Next, we pinpoint the options of representing the standard input of Structure from Motion and apply the theory of the propagation of these statistics from measurements (image points) to parameters (three-dimensional scene) in practise. We show the open issues which can be investigated in the context of the second moment propagation. The directions in which are the investigations made are robustness, speed and precision. Further, we present our previous work and show how to increase the numerical precision and speed of the propagation process. 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. Finally, we emphasize main goals of the thesis which focuses on designing scalable, robust and more precise propagation and its application to overcome the current state of the art methods.

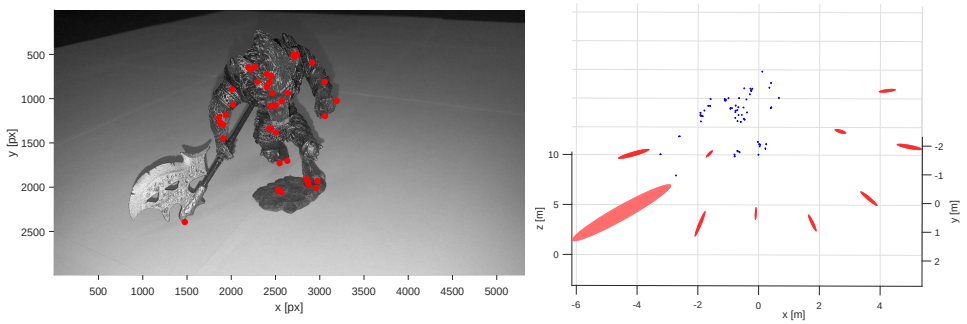
Contents

1 Introduction

Precise and robust three-dimensional (3D) scene reconstruction is an important goal of many computer vision algorithms such as Structure from Motion (SfM) and Multi-View Stereo (MVS). 3D reconstruction has received a lot of attention due to wide range of applications, e.g. quality verification of industry products [1], robot navigation [2], self-driving cars [3], virtual reality [4] and other [5, 6]. Recent work in SfM has demonstrated a possibility of reconstructing geometry from large photo collections [7,8]. We can reconstruct 3D models of entire cities from pictures taken by customer cameras using a single computer. These models can be computed from as many as hundreds of thousands of pictures and lead to reconstructions composed of millions of 3D points.

What to analyse in 3D scene?
A typical output of SfM is a set of parameters describing camera poses and coordinates of 3D points. These parameters are estimated from detected points in images. The goal of this thesis is to find out hidden relations between the parameters and how the detection error of image points coordinates (the precision of the input) influence the quality of estimated parameters (the precision of the output). These properties (relationships and precision) can be approximately described by first few moments of the parameters. The first moment (the mean) is computed by SfM and MVS in many reconstruction pipelines [9–11]. This thesis analyses rarely investigated second moment (the covariance matrix) of the parameters and its computation in practice. The directions in which the investigations are made are precision, robustness, and scalability.

Why to analyse the 3D scene?
Iterative Structure from Motion is an iterative algorithm where an error in early stages influences a lot the error of estimated 3D scene. If we knew the relationships between parameters of the scene and their precisions, we could use it for selecting the best possible model for camera representation, filtering the most unconstrained parameters and speed up the reconstruction. It would allow checking of the uncertainty of iteratively added cameras and prevent wrong extensions of existing partial reconstruction. The reconstruction pipelines would be faster and more robust. In addition, the precision of parameters may allow more sophisticated smoothing of reconstructed surfaces in dense reconstructions [12, 13], and better selection of the first reconstruction pair in sequential SfM.



(a) (b)

Figure 1: The propagation of the second moment of image points (a) to the second moment of the camera poses (b). The red ellipsoids show where are the image points and cameras likely to be. Blue points are reconstructed 3D points of Toy dataset, see Table 2.

2 State of the Art

A three-dimensional scene is an output of a reconstruction process. This section starts with a short description how the scene is build up and the summary of the advantages and drawbacks of this process. Next, we pinpoint the information which is available from the input of standard reconstruction pipelines. We collect the statistic properties of the image points and propagate them into the parameters of the 3D scene. At the end of this section, we focus on previously published enhancements which speed up the propagation process.

2.1 The reconstruction process

It is necessary to understand how the scene is built up to investigate its properties. The models of relationships (relative and absolute pose models, projection functions) between the parameters of the scene influence the relationships and the error distributions within the final 3D scene.

There are two main approaches how to reconstruct the scene. The global ones (e.g. [14, 15]) and the local ones (e.g. [10, 16]). The propagation of the statistics is performed on optimized parameters. Both approaches are usually optimized with respect to the same projection function at the end of the reconstruction. Therefore the propagation process is similar for local and global reconstructions.

We mostly focus on the Iterative SfM which can be categorized into the local ones. It starts with detection of the interesting image points called fea-

ture points. The feature points are usually detected by SIFT [17], SURF [18], MESR [19] and other [20–22] detectors. The areas around all feature points are described by descriptors (e.g. [17, 18]). The relationship between feature points and scene parameters is based on projective geometry (a summary is in Hartley - Multiple View Geometry in Computer Vision [23]). Standard implementation of Iterative SfM [10, 16], first, compute pairwise tentative matches of feature points between pairs of cameras using an Approximate Nearest Neighbors (ANN) search (e.g. [24–27]). Second, verify found tentative matches. The SfM robustly estimates parameters of relative pose model (e.g. [28, 29]) and filter all correspondences which do not fit the model. The parameters are robustly estimated by an extension of Random Sample Consensus (RANSAC), see an overview [30]. Further, the algorithm selects the first reconstruction pair of cameras and setup the global coordinate system.

The iterative part of the algorithm starts by triangulation [23] of 3D points from verified feature points which lie in images of the first reconstructed pair of cameras. A new camera is added to the partial reconstruction (e.g. the first pair of cameras with their 3D points) by solving the absolute pose problem. The absolute pose problem is the task of computing the external parameters (e.g. the orientation and the position) of the camera from the image points - 3D points correspondences. There are many absolute pose solvers (e.g. [31,32]). The algorithm iterate between adding new cameras and triangulation of new 3D points.

The parameters of the scene are often optimised after few iterations using an efficient nonlinear refinement [33, 34]. This optimization is usually implemented by Google nonlinear least squares solver Ceres [35] and use an another model of relationships based on reprojection error [23]. It minimises the distance between feature points and "bundles" of rays leaving the 3D points and creating projections into the images. This method, called Bundle Adjuster (BA) [36], usually runs at the end of reconstruction pipelines.

2.1.1 Properties of the reconstruction process

Most of the current reconstruction pipelines [9–11] are realized by Iterative SfM algorithm. It locally extends the partial scene which is fast. The another benefit is that the Iterative SfM has received an immense amount of partial improvements over few last decades. There are tens of improvements of the robust estimation (RANSAC) of the parameters, dozens of the relative and absolute pose models for different sets of parameters (e.g. radial distortion, tangential distortion, focal length, rolling shutter, etc.) and usually several implementations of each model based on different geometric relationships (e.g. angles between rays, distances between 3D points, ratios of distances

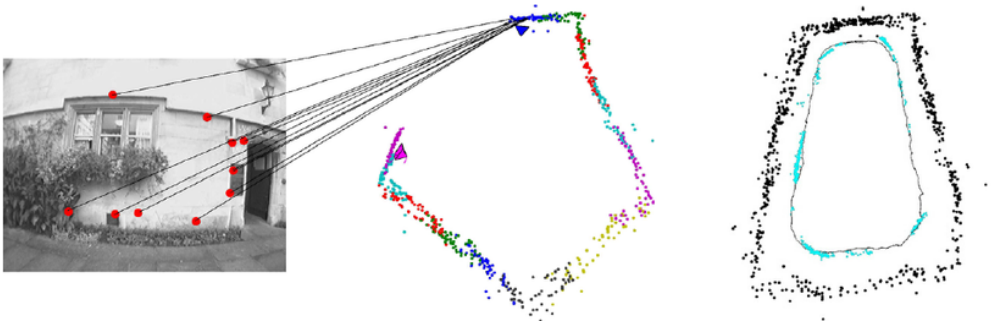


Figure 2: The 3D scene is created form images (e.g. left subfigure) captured in a loop. The result of SfM (center subfigure) have to be corrected (right subfigure). Published in [39].

between 3D points, etc.).

The drawback of Iterative SfM is that the local extension of partial reconstruction may lead to a local optimum which can be seen for example on "loop closing" problem [37, 38]. The loop composed of tens or more of cameras usually do not end in the same position as the start of the loop is. Current reconstruction pipelines [9–11] optimize the reconstruction after few iterations which slow down the reconstruction process [14, 15]. Further, the pipelines only estimate the mean of 3D scene parameters while the higher moments (relations between scene parameters) are usually too computationally expensive to be computed (current algorithms for uncertainty propagation have cubic time and quadratic memory complexity). Another disadvantage is that there is no comparison of all relative and absolute pose solvers in the sense of the second moment of the parameters.

There are also the Global SfM algorithms [14, 15]. Global SfM usually assume an approximation of the projection function and lead to another type of the errors (e.g. similar parts of the reconstruction may be reconstructed as one object, see Fig.5 in [14]).

2.2 The statistics of a 3D scene

The detection error of image points, chosen relative and absolute pose model and chosen projection function influence the precision and relationships inside the 3D scene. For example, the Iterative SfM may lead to local optimum, e.g., curves the straight geometry in 3D space because it does not consider the radial distortion parameters [40] or in opposite case, it can collapses the whole reconstruction into one flat because of considering the rolling shutter parameter [41]. Regardless the reconstruction process, the BA minimizes an

objective function and adjust the scene to an optimum. Thus, we have to perform the propagation of the uncertainty using the objective function.

2.2.1 The formulation of 3D scene

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 $\boldsymbol{p}(C_i, X_j)$. Parameter $\epsilon_{i,j}$ is the detection error of the observation $\boldsymbol{u}_{i,j}$. 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 vector composed of 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 nonlinear least squares optimization

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

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

2.2.2 The uncertainty of the measurements

In real applications, the image point $\boldsymbol{u}_{i,j}$ can be determined only to a finite accuracy due to the quantization process of the image and unmodeled errors caused by image compression and the imprecision of detection. We call the sum of these errors: the *detection error* $\epsilon_{i,j}$. The distribution of detection error may be different from point to point. These distributions may be approximated by the normal distributions and described by their first and second moments. The representation of the second moment for multiple variables is called the *covariance matrix*. The detection error has zero mean $E(\epsilon_{ij}) = 0$ and a nonzero covariance matrix $\Sigma_{\epsilon_{i,j}} \in \mathbb{R}^{2\times 2}$ for each $\epsilon_{i,j}$. The third and higher moments have not been used in practice because we cannot estimate them reliably unless we have a large number of samples of each

image point [42]. Note that, we usually have one image point for describing its distribution. The covariance of one point is always zero matrix. So, the previous work usually approximates these distributions based on the norm of the residuals.

Lhuillier and Perriollat [43] assumed that all points have the same standard deviation scaled by the squared Euclidean norm of the residuals. They defined the covariance matrix for all image points as

$$\Sigma_{\boldsymbol{u}} = \sigma^2 \mathtt{I} \tag{4}$$

where $\mathtt{I}$ is the identity matrix scaled by an approximation of the non-biased *variance factor*

$$\sigma^2 = \|r(\theta)\|^2 / (2k - pn - 7) \tag{5}$$

Bishop [44] defined a *precision matrix* $P_{\boldsymbol{u}}$ for statistically independent measurements with nonzero covariance matrix

$$P_{\boldsymbol{u}} = \mathrm{Diag}(|r(\theta)|)^{-1} \qquad \Sigma_\epsilon = \Sigma_{\boldsymbol{u}} = P_{\boldsymbol{u}}^{-1} \tag{6}$$

The precision matrix $P_{\boldsymbol{u}}$ is the inversion of the diagonal matrix composed of the absolute values of the residual vector. Förstner and Wrobel [45] used the precision matrix to weight the sum of squared residuals and to derive a formula for maximum likelihood estimate (MLE) $\hat{\theta}$ from observations $\boldsymbol{u}$, see Section 2.2.3. This approach replaced one scalar, the variance factor σ^2, by diagonal covariance matrix Σ_ϵ. The image points, e.i. their covariances $\in \mathbb{R}^{2\times 2}$ on the diagonal, has large values (large uncertainty) for the points with large reprojection error and vice verse.

Kanatani and Morris [46] assumed that the noise vector ϵ is the Gaussian random variable, which may not be independent for different images. The unknown variance was set up using the template matching. The patch of corresponding points was shifted around the detected point and normalized variation of the residual $R_{\boldsymbol{u}}$ was used instead of identity matrix

$$\Sigma_{\boldsymbol{u}} = \sigma_R^2 \frac{R_{\boldsymbol{u}}}{||R_{\boldsymbol{u}}||} \tag{7}$$

The variance factor σ_R^2 which they called noise level was computed following [47].

2.2.3 The propagation of the uncertainty

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 [23, 45]. We are computing the backward transport (from measurements to the parameters) of the uncertainty for the nonlinear over-parameterized system (represented by objective function $r(\theta)$). The system of equations is over-parameterized because any 3D scene can be shifted, scaled and rotated without any change of the objective function we optimize.

To describe the propagation process we are using following notation

Variable	Expectation	Covariance	Meaning
ϵ	$E(\epsilon) = 0$	Σ_ϵ	detection error
$\boldsymbol{u}$	$E(\boldsymbol{u})$	$\Sigma_u = \Sigma_\epsilon$	measured observations
$\hat{\boldsymbol{u}}$	$E(\hat{\boldsymbol{u}})$	$\Sigma_{\hat{\boldsymbol{u}}}$	projections of $f(\hat{\theta})$, $\hat{\boldsymbol{u}} = f(\hat{\theta})$
$\bar{\boldsymbol{u}}$	$E(\bar{\boldsymbol{u}})$	0	correct observations
θ	$E(\theta)$	Σ_θ	scene parameters
$\hat{\theta}$	$E(\hat{\theta})$	$\Sigma_{\hat{\theta}}$	MLE of scene parameters
$\bar{\theta}$	$E(\bar{\theta})$	0	correct scene parameters

Table 1: The notation for detection error, measured, estimated and correct observations and parameters of 3D scene. MLE is the abbreviation for maximum likelihood estimate.

2.2.4 Forward propagation of linear function

We show an example using the Equation 2, i.e. the function $f(\theta)$. Forward propagation propagate the input of the function (e.g. Σ_θ) to the output of the function (e.g. $\Sigma_{\boldsymbol{u}}$). The vector ϵ is a constant vector and therefore do not change the output covariance.

If we assume a linear function f, the Equation 2 can be rewritten into the matrix form

$$\boldsymbol{u} = f(\theta) + \epsilon = \bar{A}\,\theta + \epsilon \tag{8}$$

where $\bar{A}$ realize the mapping of function f. Let us assume that each camera and 3D point has its own distribution approximated by normal distribution. These distributions are diagonal blocks inside the covariance matrix Σ_θ. Due

to the linearity of the expectation operator holds

$$E(\boldsymbol{u}) = \bar{A}\, E(\theta) \qquad \Sigma_{\boldsymbol{u}} = \bar{A}\Sigma_{\theta}\bar{A}^{\top} \tag{9}$$

2.2.5 Forward propagation of nonlinear function

If f is a nonlinear differentiable function we can approximate it using Taylor expansion (TE). The linearization up to forth-order in point $\hat{\theta}$ equals

$$\boldsymbol{u} \approx f(\hat{\theta}) + f'(\hat{\theta})(\theta - \hat{\theta}) + \frac{1}{2}f''(\hat{\theta})(\theta - \hat{\theta})^2 + \frac{1}{6}f'''(\hat{\theta})(\theta - \hat{\theta})^3 + \frac{1}{24}f'''(\hat{\theta})(\theta - \hat{\theta})^4 \tag{10}$$

Our point $\hat{\theta}$ is the maximum likelihood estimate of 3D scene parameters, i.e. $\hat{\theta} = E(\theta)$. If θ is random variable with normal distribution with symmetric density function, the expectation and covariance matrix equal

$$E(\boldsymbol{u}) \approx f(\hat{\theta}) + \frac{1}{2}f''(\hat{\theta})(\theta - \hat{\theta})^2 + \frac{1}{24}f'''(\hat{\theta})(\theta - \hat{\theta})^4 \tag{11}$$

$$\Sigma_{\boldsymbol{u}} \approx f'^2(\hat{\theta})(\theta - \hat{\theta})^2 + \frac{1}{3}\left(f'(\hat{\theta})f'''(\hat{\theta}) + \frac{1}{2}f''^2(\hat{\theta})\right)(\theta - \hat{\theta})^4 \tag{12}$$

and the third moment $(\theta - \hat{\theta})^3$ equals zero. For non-symmetric density function becomes the propagation functions (i.e. Equations 11,12) more complicated.

However, we usually do not have the third and higher moments for the detection error. Thus, the previous work [23, 43, 45] use the first order Taylor expansion for the forward propagation

$$\boldsymbol{u} \approx f(\hat{\theta}) + f'(\hat{\theta})(\theta - \hat{\theta}) = f(\hat{\theta}) + J_f(\theta - \hat{\theta}) \tag{13}$$

where J_f is the partial derivation of function f in $\hat{\theta}$. It leads to expectation and covariance matrix

$$E(\boldsymbol{u}) \approx f(\hat{\theta}) \qquad \Sigma_{\boldsymbol{u}} \approx J_f\Sigma_{\theta}J_f^{\top} \tag{14}$$

The first order linearization is also used in backward propagation because we describe the observations by first two moments, the mean $E(\boldsymbol{u}) = \boldsymbol{u}$ and the covariance matrix $\Sigma_{\boldsymbol{u}}$.

2.2.6 Backward propagation of linear function

We derive how to construct the function q which maps the measurements $\boldsymbol{u}$ to the maximum likelihood estimate of 3D scene parameters $\hat{\theta}$. Then, we perform the forward propagation using q. First, we define *the weight sum of the squared residuals*

$$\Omega(\theta) = r^\top(\theta)\, P_{\boldsymbol{u}}\, r(\theta) \tag{15}$$

The MLE of 3D scene parameters minimises the function

$$\hat{\theta} = \arg\min_{\theta} \Omega(\theta) \tag{16}$$

If we assume that f is a linear function, f can be rewritten in a matrix form

$$f(\theta) = \bar{A}\theta \tag{17}$$

and the residual function will be represented by

$$r(\theta) = \bar{A}\theta - \boldsymbol{u} \tag{18}$$

The necessary condition for estimating of the minimum $\Omega(\theta)$ is that the partial derivation equals zero. So, it holds

$$\frac{1}{2}\left[\frac{\partial\Omega(\theta)}{\partial\theta}\right]_{\theta=\hat{\theta}} = \bar{A}^\top P_{\boldsymbol{u}}(\bar{A}\hat{\theta} - \boldsymbol{u}) = 0 \tag{19}$$

which can be adjusted into the form called *normal equation system*

$$\widetilde{N}\hat{\theta} = \widetilde{n} \tag{20}$$

where the unknown parameter $\hat{\theta}$ appears linear and the matrices equal

$$\widetilde{N} = \bar{A}^\top P_{\boldsymbol{u}} \bar{A} \qquad \widetilde{n} = \bar{A}^\top P_{\boldsymbol{u}} \boldsymbol{u} \tag{21}$$

The solution of this system is the function $q(\boldsymbol{u})$

$$\hat{\theta} = q(\boldsymbol{u}) = \widetilde{N}^{-1}\widetilde{n} = (\bar{A}^\top P_{\boldsymbol{u}} \bar{A})^{-1} \bar{A}^\top P_{\boldsymbol{u}}(\boldsymbol{u}) \tag{22}$$

If we apply the forward propagation of linear function (Equation 9) to the function $q(\boldsymbol{u})$ the covariance matrix of the estimated parameters will be

$$\Sigma_{\hat{\theta}} = \widetilde{N}^{-1} = (\bar{A}^\top P_{\boldsymbol{u}} \bar{A})^{-1} \tag{23}$$

This formula propagates the uncertainty ($\Sigma_{\boldsymbol{u}} = P_{\boldsymbol{u}}^{-1}$) to the uncertainty of MLE of scene parameters $\Sigma_{\hat{\theta}}$. The propagation is in the opposite direction than forward propagation (i.e. Equation 9) and therefore we call it the *backward propagation* [23].

2.2.7 Backward propagation of nonlinear function

Hartley [23] has shown the backward propagation for nonlinear differentiable function. The function f is approximated by its first-order approximation and the residual function in Equation 18 can be rewritten to

$$r(\theta) \approx J_f(\theta - \hat{\theta}) \tag{24}$$

where J_f is the Jacobian of function f in $\hat{\theta}$. Assume that we do not have an over-parametrized system, i.e. the Jacobian J_f has full rank which equals the number of scene parameters $n\,p + 3m$. While the objective function $r(\theta)$ do not depend on $\hat{\theta}$ we can substitute $J_f\hat{\theta}$ by a constant vector j_f and write the residual function

$$r(\theta) \approx J_f\theta - j_f \tag{25}$$

Applying the Equations 19-26 leads to the formula for approximation of covariance matrix. It is the same as in Hartley [23]. The backward propagation for nonlinear not over-parametrized function f is

$$\Sigma_{\hat{\theta}} \approx (J_f^\top P_{\boldsymbol{u}} J_f)^{-1} \tag{26}$$

2.2.8 Backward propagation of over-parameterized nonlinear function

In our case, the objective function $r(\theta)$ is over-parameterized. The parameters θ may vary without bound which means that their uncertainties are infinitely large and the Jacobian J_r does not have full rank. Note that J_f equals J_r because $\boldsymbol{u}$ is the vector of constant values. Therefore, the inversion of Fisher information matrix (Equation 26) does not exist. We can solve this problem two ways.

First, we can add some restrictions to make the Jacobian J_f full-rank. Kanatani [46] presented a theory for describing the uncertainties under changing regularisation conditions (gauge transformations). There is a large number of choices of the regularisation conditions (e.g. fixing some parameters of the scene: camera rotation, camera position, 3D point, etc. or fixing some statistical properties of the scene: mean of a subset of 3D points, the scale of a subset of camera poses, etc.). The additional restrictions change the optimization function, and therefore we obtain different covariance matrices for one 3D reconstruction based on various restrictions. The comparison of the covariance matrices computed with different restrictions against the ground truth was part of our research, and it is in Section 3.

Second, we can project the parameters θ to the subset which uniquely describe a 3D scene. The minimal subset of parameters which uniquely

describe a scene is called the set of *essential parameters* θ_e. The number of essential parameters $n_{\theta e}$ equals the number of scene parameters n_θ minus the rank of the null space of J_f, i.e. $n_{\theta e} = n_\theta - 7 = np + 3m - 7$. So, we can reduce 7 parameters, e.g. 3 for scene rotation, 3 for scene position and one for scale. The mapping to the set of essential parameters can be realized by a matrix $\hat{A} \in \mathbb{R}^{n_\theta \times n_{\theta e}}$. The column vectors of $\hat{A}$ span the tangent space S_θ at $\bar{\theta}$. Since we do not know the correct parameters $\bar{\theta}$ the previous work [43, 45, 46] used MLE $\hat{\theta}$ instead of $\bar{\theta}$. The space S_θ is a smooth sub-manifold of dimension $n_{\theta e}$ which is embedded to n_θ, pass through $\hat{\theta}$ and for which exist one-one mapping from neighbourhood of $\hat{\theta}$ to observations, i.e. $f(S_\theta) \in \mathbb{R}^{2k}$. We can write the function $s : \mathbb{R}^{n_{\theta e}} \to \mathbb{R}^{n_\theta}$ which derivation equals the matrix $\hat{A}$. So that, the composition $f \circ s$ of over-parametrized f and mapping s leads to one-one map from sub-manifold S_θ to observations. The derivation of $f \circ s$ equals $J_f \hat{A}$ and after applying the Equation 26, the formula

$$\Sigma_{\theta e} = (\hat{A}^\top J_f^\top P_{\boldsymbol{u}} J_f \hat{A})^{-1} \tag{27}$$

realize the mapping of the covariance matrix of the image points to the covariance matrix of the essential parameters $\Sigma_{\theta e} \in \mathbb{R}^{n_{\theta e} \times n_{\theta e}}$. If we apply forward propagation to the set of parameters, i.e the Equation 14, we get

$$\Sigma_{\hat{\theta}} = (J_f^\top P_{\boldsymbol{u}} J_f)^{+\hat{A}} = \hat{A}(\hat{A}^\top J_f^\top P_{\boldsymbol{u}} J_f \hat{A})^{-1} \hat{A}^\top \tag{28}$$

The expression depend on particular choice of column-space of $\hat{A}$. The matrix $\hat{A}$ can be also seen as the regularisation matrix. For example, fixing one of scene parameters cause that the corresponding column in $\hat{A}$ is a zero vector. The Jacobian J_f with this additional constrain is equals the Jacobian without this constrain multiplied by $\hat{A}$. The covariance matrix $\Sigma_{\hat{\theta}}$ has dimension n_θ, rank $n_{\theta e}$ and zero variance in directions orthogonal to S_θ. Kanatani 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 Moore-Penrose (M-P) pseudoinverse

$$\Sigma_{\hat{\theta}} = (J_f^\top P_{\boldsymbol{u}} J_f)^+ \tag{29}$$

In this case is the constrained surface S_θ orthogonal to the null space of J_f. For example, a homogeneous vector v_h has as the constraint surface the unit sphere $||v_h|| = 1$. The function which works with homogeneous vectors is usually invariant to changes of scale (the radial direction) and thus its Jacobian has null vector in radial direction. The tangent plane S_{v_h} is perpendicular to the parameter vector v_h, i.e. to the range of the Jacobian, in any point and the covariance matrix has zero variance in S_{v_h}. Thus, the normal form

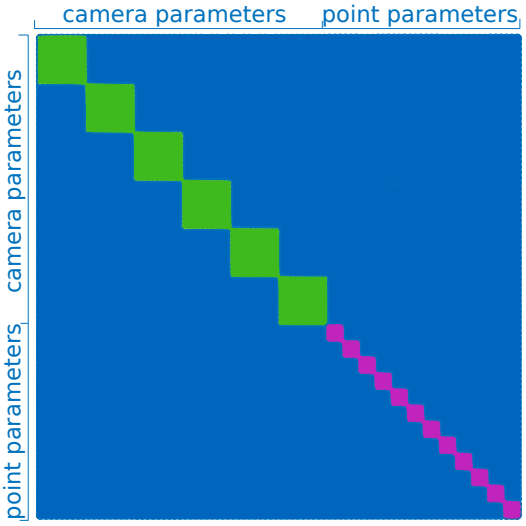


Figure 3: The structure of the covariance matrix for Cube dataset. The covariance matrix $\Sigma_\theta \in \mathbb{R}^{90\times90}$ is composed of blocks $\mathbb{R}^{9\times9}$ camera (green) and $\mathbb{R}^{3\times3}$ point (violet) covariance matrices

of the covariance matrix has zero variance in radial direction, the direction of the additional constrain.

2.2.9 Sparse backward propagation

In the case of Equation 26, i.e. not-overparametrized objective function, we can use the sparse algorithms [48] to compute interesting parts of the covariance matrix of scene parameters (i.e. cameras and points submatrices), see Figure 3. This computation can be done without determining the inverse of the normal equation matrix $\widetilde{N}$. The covariance matrix $\Sigma_{\boldsymbol{u}}$ and the precision matrix $P_{\boldsymbol{u}}$ are diagonal matrices for mutually independent image points. Therefore, for mutually independent image points hold

$$\widetilde{N} = \sum_{i=1}^{n_\theta} \boldsymbol{p}_i \bar{\boldsymbol{a}}_i \bar{\boldsymbol{a}}_i^\top \qquad \widetilde{n} = \sum_{i=1}^{n_\theta} \boldsymbol{p}_i \boldsymbol{u}_i \bar{\boldsymbol{a}}_i \tag{30}$$

where $\bar{\boldsymbol{a}}_i$ is i-th row of the matrix $\bar{A}$ and $\boldsymbol{p}_i$ is i-th column of $P_{\boldsymbol{u}}$. We can see that each item of the sum equals one vector of the matrix $\widetilde{N}$ and one number of $\widetilde{n}$. Thus, $\widetilde{N}$,$\widetilde{n}$ cannot be stored. The interesting parts of covariance matrix Σ_θ can be computed using Cholesky decomposition. This sparse approach was used in recent paper Polok [49], however the author didn't assume the over-parametrization and used Cholesky decomposition

on rank-deficient matrix $\widetilde{N}$. Note that, Cholesky decomposition works only for symmetric positive semi-definite full-rank matrices.

2.2.10 Speed up of general backward propagation

The M-P pseudoinverse is computationally demanding task (i.e. has the cubic time and quadratic memory complexity). Lhuillier and Perriollat [43] decomposed the normal equation matrix $\widetilde{N}$ also called the *Fisher information matrix* [50] to sub-blocks

$$\widetilde{N} = J_f^\top P_{\boldsymbol{u}} J_f = \begin{bmatrix} U_{\widetilde{N}} & W_{\widetilde{N}} \\ W_{\widetilde{N}}^\top & V_{\widetilde{N}} \end{bmatrix} \tag{31}$$

The Schur complement [51] of the submatrix of 3D point parameters

$$Z_{\widetilde{N}} = U_{\widetilde{N}} - W_{\widetilde{N}} V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top \tag{32}$$

has the same size as the block of camera parameters $Z_{\widetilde{N}} \in \mathbb{R}^{np \times np}$. It is much smaller than $\Sigma_\theta \in \mathbb{R}^{n_\theta \times n_\theta}$ since the reconstructions usually contain much fewer cameras than 3D points. The inversion of the decomposed information matrix may be written

$$\Sigma_{\hat{\theta}} = \sigma^2 \begin{bmatrix} \mathtt{I} & 0 \\ -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top & \mathtt{I} \end{bmatrix} \begin{bmatrix} Z_{\widetilde{N}}^{-1} & 0 \\ 0 & V_{\widetilde{N}}^{-1} \end{bmatrix} \begin{bmatrix} \mathtt{I} & -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top \\ 0 & \mathtt{I} \end{bmatrix} \tag{33}$$

if the following two conditions hold: the input covariance matrix equals $\Sigma_{\boldsymbol{u}} = \sigma^2 \mathtt{I}$ and the matrix $Z_{\widetilde{N}}$ has full-rank. Generally, the matrix $Z_{\widetilde{N}}$ has not full-rank. In that case, we can replace the inversion of $Z_{\widetilde{N}}^{-1}$ by pseudoinversion and write the decomposition

$$\Sigma_{\hat{\theta}} = \sigma^2 \begin{bmatrix} \mathtt{I} & 0 \\ -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top & \mathtt{I} \end{bmatrix} \begin{bmatrix} Z_{\widetilde{N}}^{+} & 0 \\ 0 & V_{\widetilde{N}}^{-1} \end{bmatrix} \begin{bmatrix} \mathtt{I} & -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top \\ 0 & \mathtt{I} \end{bmatrix} \tag{34}$$

if the rank additivity condition [23, 52] holds. The rank additivity condition meas that the sum of rank of the submatrices equals the rank of whole matrix

$$rank\, \widetilde{N} = rank \begin{bmatrix} U_{\widetilde{N}} \\ W_{\widetilde{N}}^\top \end{bmatrix} + rank \begin{bmatrix} W_{\widetilde{N}} \\ V_{\widetilde{N}} \end{bmatrix} = rank \begin{bmatrix} U_{\widetilde{N}} & W_{\widetilde{N}} \end{bmatrix} + rank \begin{bmatrix} W_{\widetilde{N}}^\top & V_{\widetilde{N}} \end{bmatrix} \tag{35}$$

The problem is that the submatrices $U_{\widetilde{N}}$,$V_{\widetilde{N}}$ has full-rank and the matrix $\widetilde{N}$ is rank deficient. Therefore, even this condition does not hold. Thus, Lhuillier extend the Equation 34 about the correction terms

$$P_f^\perp = \mathtt{I} - K_f (K_f^\top K_f)^{-1} K_f^\top \tag{36}$$

$$P_f^c = \mathtt{I} - K_f (J_c K_f)^{-1} J_c \tag{37}$$

where K_f equals the kernel of $J_f^\top J_f$ and J_c is the Jacobian of additional constrains (the matrix $\hat{A}$ in our notation) and the formulas

$$\Sigma_{\hat{\theta}} = \sigma^2 P_f^{\perp} \begin{bmatrix} \mathtt{I} & 0 \\ -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top & \mathtt{I} \end{bmatrix} \begin{bmatrix} Z_{\widetilde{N}}^{+} & 0 \\ 0 & V_{\widetilde{N}}^{-1} \end{bmatrix} \begin{bmatrix} \mathtt{I} & -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top \\ 0 & \mathtt{I} \end{bmatrix} (P_f^{\perp})^\top \tag{38}$$

$$\Sigma_{\hat{\theta}} = \sigma^2 P_f^{c} \begin{bmatrix} \mathtt{I} & 0 \\ -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top & \mathtt{I} \end{bmatrix} \begin{bmatrix} Z_{\widetilde{N}}^{+} & 0 \\ 0 & V_{\widetilde{N}}^{-1} \end{bmatrix} \begin{bmatrix} \mathtt{I} & -V_{\widetilde{N}}^{-1} W_{\widetilde{N}}^\top \\ 0 & \mathtt{I} \end{bmatrix} (P_f^{c})^\top \tag{39}$$

should approximate the normal form of the covariance matrix. Lhuillier has also published a proof number one in [43] that exist such correction term however, there is no straightforward connection between the proof and the correction term actually used (Equation 36-39). We empirically observed that neither $P_f^{\perp}$ nor P_f^{c} is composed from some matrices $\widetilde{Q}\widetilde{Q}^\top$ for which holds $\widetilde{Q}^\top\widetilde{Q} = \mathtt{I}$ and $\widetilde{N} = \widetilde{Q}\widetilde{D}^2\widetilde{Q}^\top$ where D is a diagonal and $\widetilde{Q}$ a rectangular matrix.

2.2.11 Other work

There are also many specific extensions for computation of the uncertainty of lines [53], edges [54], laser scans [55, 56], and stereo setups [57, 58] which are not general. The authors tried to approximate covariances of specific setups using heuristics instead of following the general uncertainty propagation method.

2.3 Properties of the propagation process

The previous work estimates very roughly the second moment of the detection error. Therefore, the output of the uncertainty propagation may be corrupted. This problem was not well investigated and may be solved by more detailed analysis of the second and higher moments of the detection error.

To invert the Fisher 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 [23], i.e. there is no ambiguity in parameters after all observations have been taken into account. We showed that many additional restrictions could realize such projection. Each additional restriction changes the objective function and no comparison against the normal form of covariance matrix was published. Further, the computation of the normal form is realized by M-P pseudoinversion which is too computationally

expensive to be used in current reconstruction pipelines, see comparison of the speed of the algorithms in Section 3.4.4.

The application of the theory, described above, on the problem of the uncertainty propagation leads to another issue. Current computers and most of the libraries work with double representation of the numbers (i.e. 15 significant digits). 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 typically the sum of squared differences between the projected parameters and the measurements (reprojection errors) [23]. Therefore, the Jacobian of the objective function contains a wide range of values. The values of the Jacobian are squared into the Fisher information matrix which makes the "raw" information matrix numerically rank deficient for medium and larger image collections, see comparison of the precision in Section 3.4.3.

3 Our previous work

We presented the first approach for large scale covariance matrix propagation which is practical. We derived the Taylor expansion (TE) idea (Section 3.2) for the approximation of the M-P pseudoinversion [59]. This approximation was used to extend the Lhuillier paper [43]. After we found the problem of the decomposition of rank deficient matrices (Equation 38) we extended TE approach by estimating the inverse instead of M-P pseudoinversion.

Secondly, we presented an important experimental comparison of recent methods [43, 46] against Ground Truth (GT) covariance matrices, which we constructed using more accurate arithmetics in Maple [60] (Section 3.4).

To calculate useful inversions, we had to fix the ambiguity (gauge freedom [46]) of the 3D scene and approximate the normal form of covariance matrices [46]. The inversion allowed us to scale the information matrix and its decomposition to smaller blocks (Section 3.1), which would not be possible with the M-P pseudoinversion.

We investigated 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 covariance matrices of the camera parameters (Section 3.3). Our approach is faster, more precise and much more stable than any previous one.

The output of our work was publicly available source code which can be used as an external library in nonlinear optimization pipelines, like Ceres Solver [35].

3.1 Constrained decomposition of Fisher information matrix

The standard way to solve the backward propagation of nonlinear over-parametrized system of equations is to use the M-P pseudoinverse. The objective function which we optimize in the last step of the reconstruction process is

$$r(\theta) = f(\theta) - \boldsymbol{u} \tag{40}$$

Therefore we are in its minimum and the propagation is realized by

$$\Sigma_{\hat{\theta}} = (J^{\top} \Sigma_{\boldsymbol{u}}^{-1} J)^{+} \tag{41}$$

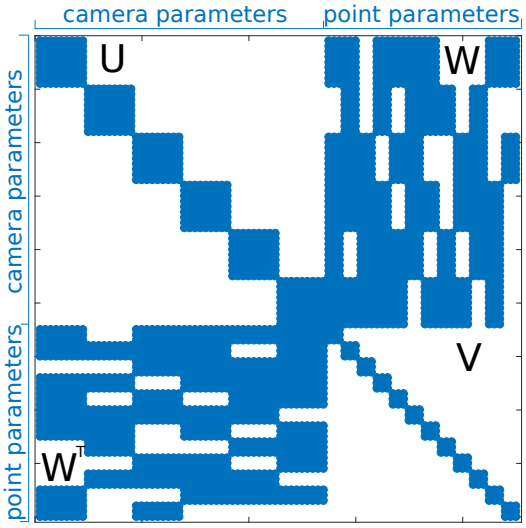


Figure 4: The structure of the information matrix for Cube dataset

with J_r replaced by J for brevity.

Pseudoinverse $\widetilde{A}^{+}$ of matrix $\widetilde{A}$ equals the inverse $\widetilde{A}$ on the range of $\widetilde{A}$ and sends the orthogonal complement of the range $\widetilde{A}$ to the zero vector [61]. We approximate the projection of orthogonal complement 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 [23, 46] and a *scaling matrix* R_s, which we introduce here. Using R, we can rewrite Equation 41 as

$$\Sigma_\theta = R(R^\top J^\top \Sigma_{\boldsymbol{u}}^{-1} J R)^{-1} R^\top \tag{42}$$

We investigate which regularisation minimizes the differences in comparison with Ground truth (GT) covariance matrices in Section 3.3. If the content of Jacobian is permuted to have cameras followed by points, i.e. $J = [J_C\, J_X]$, the information matrix

$$Q = R^\top J^\top \Sigma_{\boldsymbol{u}}^{-1} J R = \begin{bmatrix} U & W \\ W^\top & V \end{bmatrix} \tag{43}$$

will be sparse with block diagonal matrices U and V, see Figure 4.

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

$$\Sigma_\theta = \sigma^2 R \left(\begin{bmatrix} I & -Y^\top \\ 0 & I \end{bmatrix} \begin{bmatrix} Z & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I & 0 \\ -Y & I \end{bmatrix} \right)^{-1} R^\top \tag{44}$$

where matrix Z is the Schur complement [51] of the block V of the information matrix

$$Z = U + WY \tag{45}$$

We are not interested in off-diagonal blocks. All covariances of reconstruction parameters are in the blocks on the diagonal of the dense matrix Σ_θ. The interesting sub-matrices can be computed as

$$\Sigma_\theta = \sigma^2 R \begin{bmatrix} Z^{-1} & - \\ - & YZ^{-1}Y^\top + V^{-1} \end{bmatrix} R^\top \tag{46}$$

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.

3.2 The Taylor expansion algorithm

We derived the Taylor expansion (TE) algorithm for estimation of M-P pseudoinversion in [59]. After we had found the problem of the decomposition of the Fisher information matrix (Equation 38) because of using the pseudoinversion we focused on estimation of the inversion of Z. First, we defined $M = J^\top \Sigma_{\boldsymbol{u}}^{-1} J$. Using the inversion allowed us employ necessary scaling

$$(R^\top M R)^+ \neq R^+ M^+ R^{+\top} \tag{47}$$

$$(R^\top M R)^{-1} = R^{-1} M^{-1} R^{-\top} \tag{48}$$

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

$$g(\lambda) = (Z + \lambda \mathtt{I})^{-1} \tag{49}$$

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

$$\frac{d^i g}{d\lambda^i}(\lambda) = (-1)^i i (Z + \lambda \mathtt{I})^{-(i+1)} \tag{50}$$

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

$$\sum_{i=0}^{\infty} \left(\frac{(-\lambda)^i}{i!} \frac{d^i g}{d\lambda^i}(\lambda) \right) \tag{51}$$

and express the inversion of matrix Z

$$g(0) = (Z + \lambda \mathtt{I})^{-1} + \sum_{t=1}^{\infty} \left(\frac{\lambda^t}{(t-1)!} (Z + \lambda \mathtt{I})^{-(t+1)} \right) \tag{52}$$

The $\lambda \mathtt{I}$ 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.

3.3 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

$$R = R_p \, R_s \tag{53}$$

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 [23,43,46], 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 [45] and deal with measurements as parameters. Thus the objective (residual) function would be

$$r(\theta, \boldsymbol{u}) = \boldsymbol{u} - f(\theta) \tag{54}$$

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

$$A = \frac{\partial r(\theta, \boldsymbol{u})}{\partial \theta}, \quad B = \frac{\partial r(\theta, \boldsymbol{u})}{\partial \boldsymbol{u}}, \quad H = \frac{\partial h(\theta)}{\partial \theta} \tag{55}$$

and covariance matrix Σ_θ computed by

$$\begin{bmatrix} \Sigma_\theta & N \\ N^\top & P \end{bmatrix} = \begin{bmatrix} A^\top (B^\top V(\boldsymbol{u}) B)^{-1} A & H \\ H^\top & 0 \end{bmatrix}^{-1} \tag{56}$$

The covariance Σ_θ can be computed using sparse inversion, however the inverted matrix in 56 is much larger than inversion of Z and we cannot 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 cameras and 3D points directly instead of their mean, covariance and scale. If we fix one camera pose and one of the coordinates of another camera center, 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 do not 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 triple of points using RANSAC [62]. A triple of points fixes nine instead seven parameters however we empirically found out that it produces more precise results than fixing two and one-third of a 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

$$R_{s(i,j)} = 1/\|J_j\| \quad for: \; i = j; \tag{57}$$

$$R_{s(i,j)} = 0 \quad for: \; i \neq j \tag{58}$$

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

3.4 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.

#	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 2: 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 [9] and 2, 10 by COLMAP [10]

3.4.1 Computation of Ground Truth covariance matrices

We use the theory of gauge-free approach which leads to M-P pseudoinversion, described in [46]. We decompose the matrix Z using the SVD into

$$Z = \bar{U}\bar{S}\bar{V}^\top \tag{59}$$

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 3D scene 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

$$Z^+ = \bar{U}\bar{S}'\bar{V}^\top \tag{60}$$

The SVD algorithm is sensitive rounding when the range of values in matrix Z is large and different implementations may lead to different results (i.e. Maple, Matlab and Ceres which can be seen in Figure 6). 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 precise evaluation of the uncertainty matrix is computationally demanding (e.g. the SVD of Z for Daliborka dataset took approximately 22hours). Therefore, we computed GT covariance matrices only for the datasets 1-4, see Table 2.

3.4.2 Datasets

We experimented with realistic synthetic reconstructions, as well as with middle to large scale Internet datasets. The parameters of the datasets are

#	Algorithm
1.	SVD of M using Maple (Kanatani [46]) (GT)
2.	TE inversion of scaled Z with three points fix
3.	SVD of M using Ceres (Kanatani [46])
4.	TE inversion of scaled Z with trivial camera fix
5.	SVD of Z with correction term (Lhuillier [43])
6.	SVD of M using Matlab (Kanatani [46])
7.	M-P inverse of Z using TE (Polic [59])

Table 3: Compared algorithms

summarized in Table 2. The datasets 2, 4 were reconstructed by publicly available pipelines (COLMAP [10], Bundler [9]) and after that, the number of the points in 3D was reduced to allow computing GT covariance matrices.

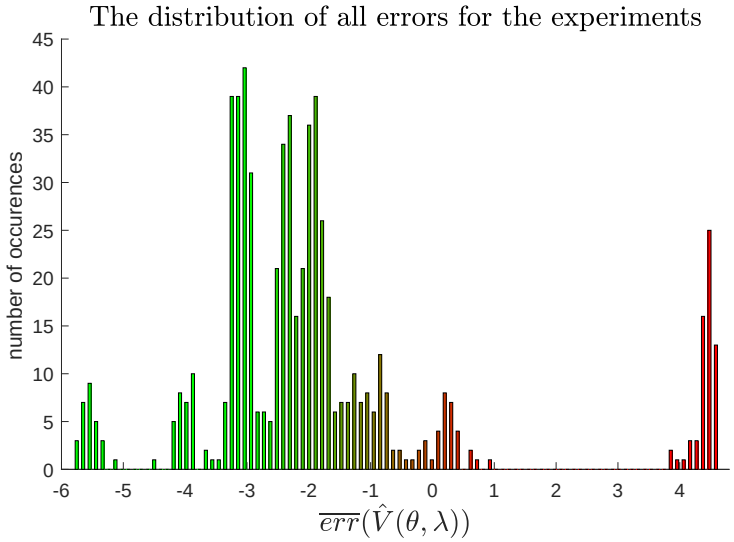


Figure 5: The distribution of all errors (with the corresponding color coding) for the experiments in Figure 6

3.4.3 Precision

We compared the algorithms summarized in Table 3. The algorithm 1 uses Maple computation with 100 significant digits and its result is considered the Ground Truth (GT). The algorithm 3 uses Ceres [35], algorithm 2 uses C++ libraries while other algorithms use Matlab [60].

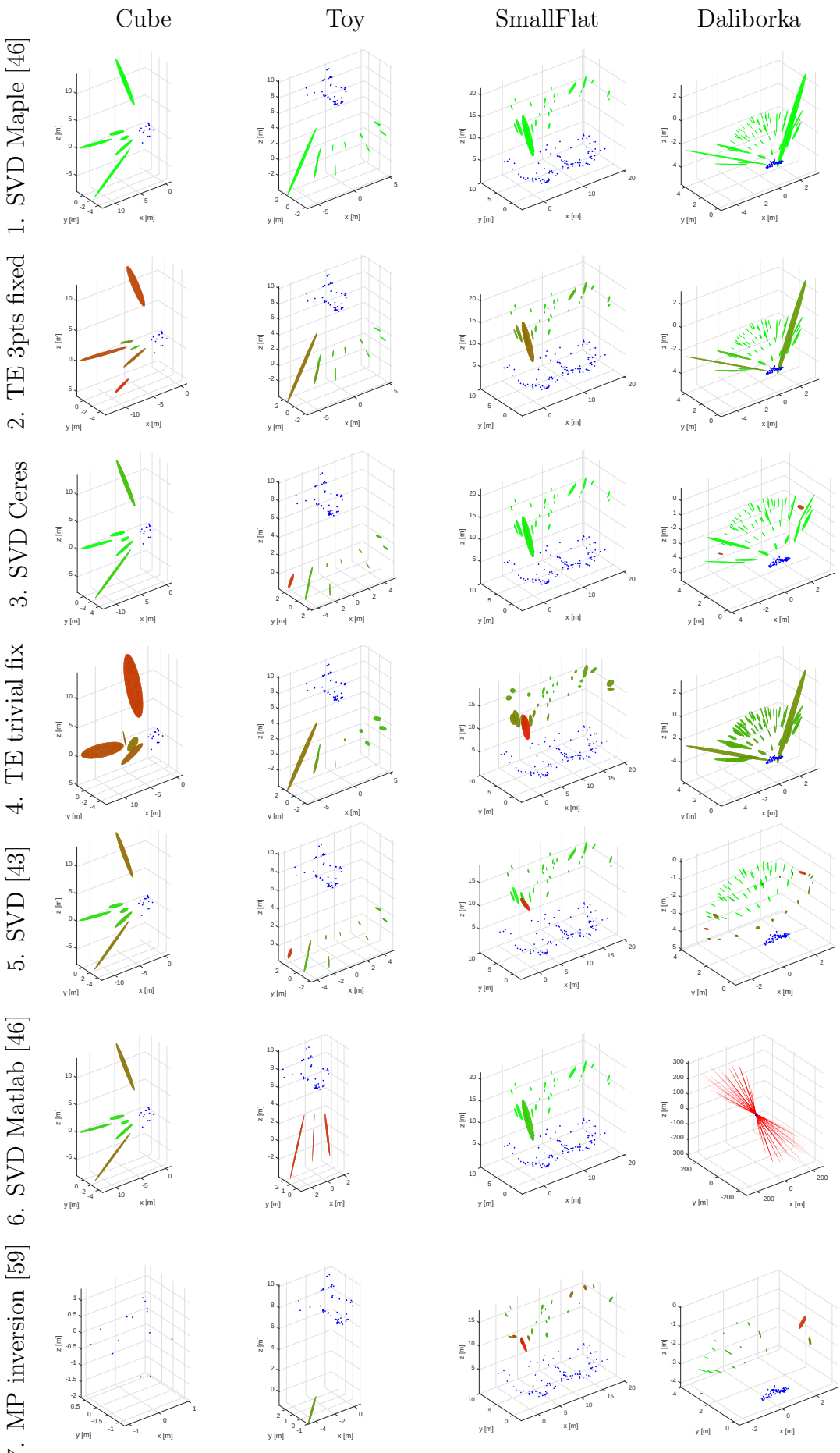


Figure 6: The comparison of the algorithms from Table 3 for the datasets 1-4 from Table 2

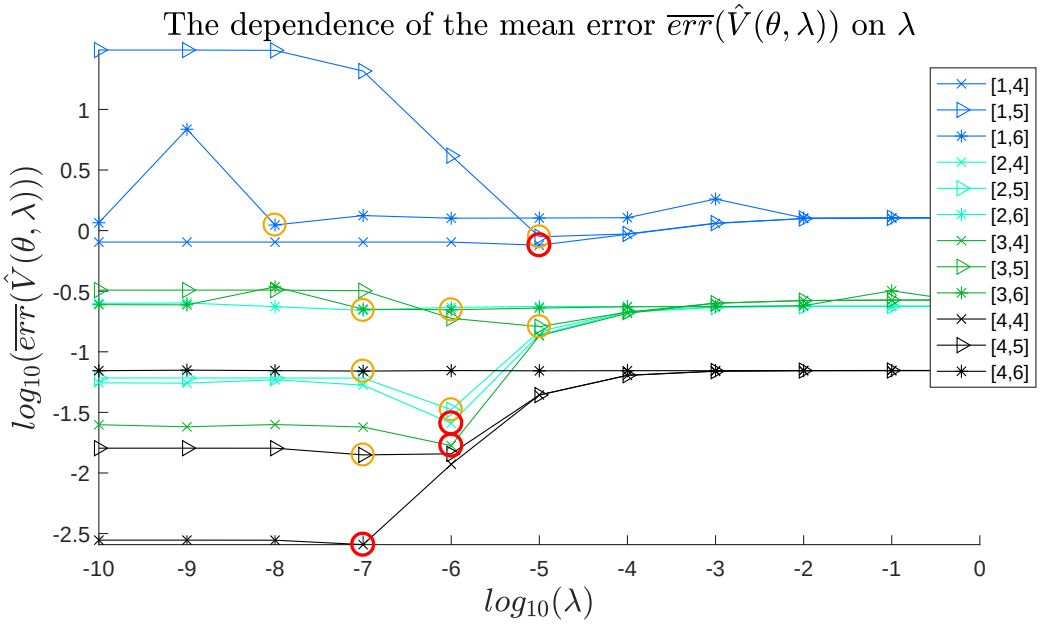


Figure 7: Each point represents the mean of errors (described precisely in Section 3.4.3) of the uncertainty matrices for one [dataset,algorithm] and given damping term λ. The lambdas chosen by our algorithm are shown as red circles and the lambdas chosen by other algorithms are shown as orange circles.

Figure 6 shows the uncertainties. The order of rows corresponds to the order of algorithms in Table 3 (i.e. the first row corresponds to the SVD of M using Maple). You can see that the correct gauge-free approach [46] 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 fail, see algorithm 6 for Daliborka dataset. This problem was not solved by any of previous approaches [23, 43, 59] which are on rows 4,5,7. The algorithm 4 is an improved version of [23]. It scales Jacobian by suitably chosen R_p, see Section 3.3. You can see that the Lhuillier algorithm [43] (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 6). Our algorithm, TE inversion, has the opposite trend, i.e. the error decreases with the growing size of the reconstruction. Figure 5 shows the distribution of all errors (with the corresponding color coding) for the experiments in shown Figure 6.

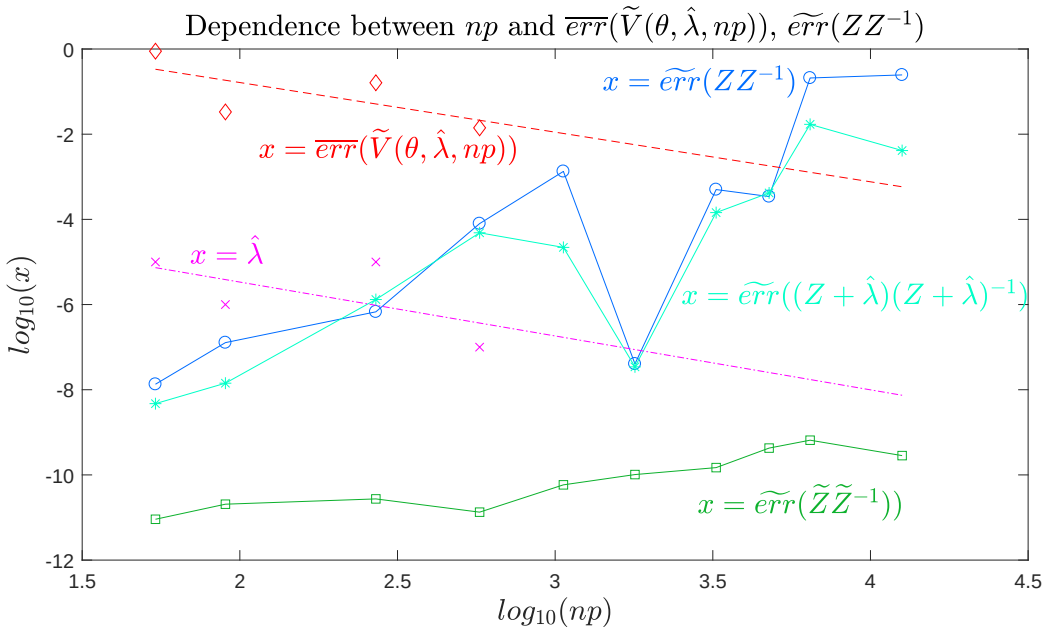


Figure 8: Each point (except $\hat{\lambda}$, the empirically selected value of λ) represents the mean of errors (described precisely in Section 3.4.3) of the uncertainty matrices for TE inversion algorithm, one dataset and $\hat{\lambda}$ based on number of camera parameters $n\,p$

The inversion of $Z+\lambda I$ is 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 dependent on parameter λ is shown in Figure 7. Error function $\overline{err}(\hat{V})$ is computed as the mean of the Frobenius norm of the elements of $\hat{V}-\hat{V}_{GT}$, which correspond to camera orientations and centers. It has been observed that the errors in covariances of extrinsic camera parameters are sufficient for finding suitable values of λ.

Figure 8 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 $\hat{\lambda}$ dependent on the number of camera parameters np) with increasing reconstruction size (i.e. the size of inverted matrix Z). The Figure 8 also shows (solid lines) the error

$$\widetilde{err}(ZZ^{-1})=\sum_{1}^{np}\frac{1}{10^4 np}\left(\Sigma_{k=1}^{10^4}(Z\,Z^{-1}-I)\boldsymbol{x}_k\right) \tag{61}$$

of inversion Z^{-1} where $\boldsymbol{x}\in\mathbb{R}^{np}$ is a random vector with zero mean and

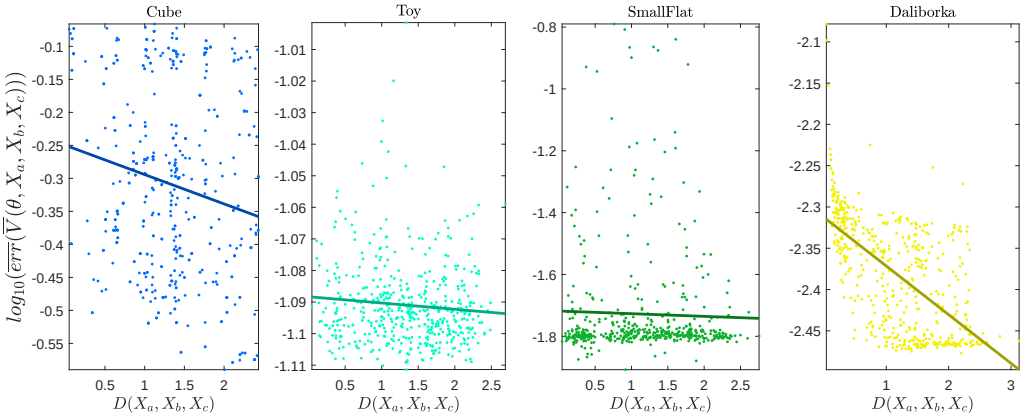


Figure 9: Each subplot represents one dataset (i.e. 1-4) and each point represents the mean of errors of uncertainty matrices for selected dataset and different sets of fixed points $\{X_a, X_b, X_c\}$. The function D is the sum of all points distances, i.e. $D(X_a, X_b, X_c) = \|X_a - X_b\| + \|X_a - X_c\| + \|X_b - X_c\|$.

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 45) 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

$$\hat{\lambda} = 10^{-1.2653 \log_{10}(n) - 2.9415} \tag{62}$$

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

3.4.4 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 [63]) 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

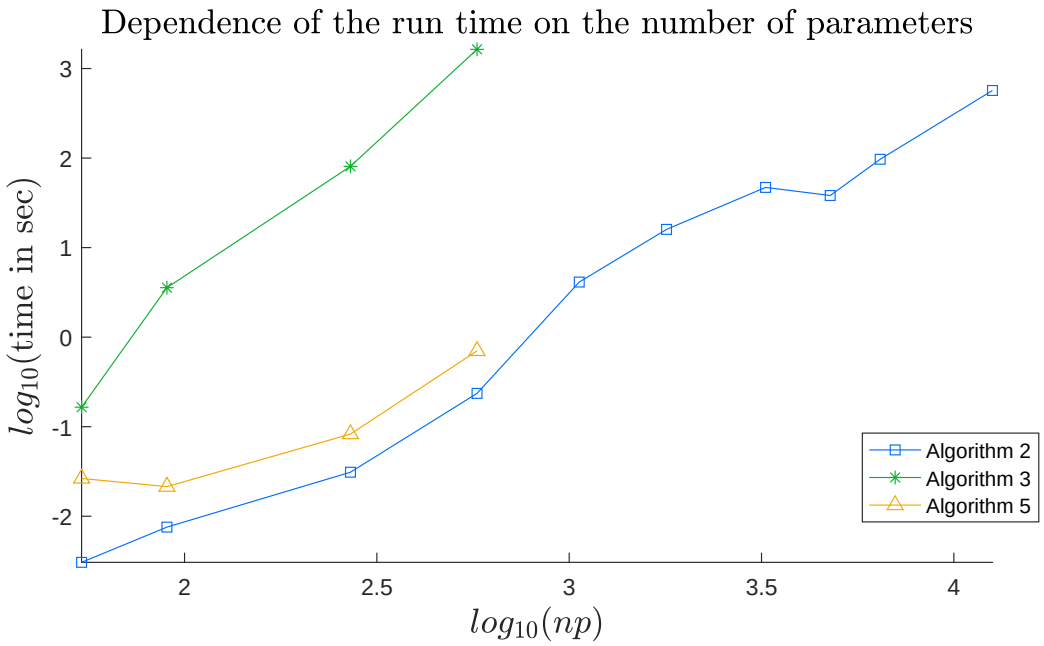


Figure 10: The dependency between the number of camera parameters of the reconstruction (equal the dimension of Z) and the run time of the algorithm. Each point represents evaluation of one dataset from Table 2

Marianska without reduction of 3D points has 243681 reconstruction parameters and requires about 470GB for dense representation of matrix M. Thus, we cannot use current implementation of Ceres 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. Our 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 [64, 65].

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 10. 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 cannot 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 operating system.

3.5 Conclusion and future work

Previous work for evaluating the quality [23, 43, 46] of the reconstruction by error propagation from measurements to the estimated parameters was based on Moore-Penrose pseudoinversion (i.e. Singular Value Decomposition [66]) which is computationally challenging and mostly imprecise for real datasets because of a wide range of values in the information matrix. We proposed a method which computes the approximation of the inversion and the M-P pseudoinversion [59] of Fisher 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. The additional analysis may lead to more precise evaluation of the uncertainty of the points in 3D.

4 Goals of the thesis

The main purpose of the thesis is to address the shortcomings of 3D scene analysis and provide improvements in terms of performance, robustness and applicability.
In particular, we identify following problems that are worth solving:

1. **Solving the problem of uncertainty propagation**
   We have presented a new algorithm for the propagation of the uncertainty for large 3D scenes. However, we believe that local propagation of the uncertainty (i.e. between few cameras and their points) can iteratively converge to the same results. Such approach might be faster and allows the computation of uncertainty for any scene regardless its size.

2. **Comparing the current relative and absolute pose solvers**
   There is no comparison of absolute and relative pose solvers (e.g. [28, 29, 31, 32]) in sense of the second moments (i.e. the precision of estimated parameters is not known). We believe that such comparison may lead to better selection of the model for a particular scene and makes reconstruction pipelines more robust.

3. **Speeding up the Structure from Motion algorithm**
   The SfM usually reconstruct as many 3D points as possible and optimize them almost in each iteration. The information about the precision of scene parameters may allow us removing or not optimizing the least conditioned 3D points and cameras. The optimization of well-conditioned sub-scene should be faster and more robust.

We have studied the problem 1 in [59] discussed how to analyze the properties of the 3D scene. We described the scene approximately by first two moments. Our algorithm propagates these moments more precisely on much larger scenes than the previous algorithms [23, 43, 49], i.e. thousands of cameras and millions of 3D points.

We empirically observed that the uncertainty of the parameters of a scene change a little when we remove few 3D points. Thus, we believe that well-selected subset of input parameters may be sufficient to estimate covariances of the 3D scene and it may be possible to employ the unscented transformation [67] or the iterative local propagation (problem 1) to further improve the uncertainty propagation process.

Finally, we would like to apply the results from problem 1 in practical applications: comparing the current absolute and relative pose solvers (problem 2), creating the most conditioned sub-scene and speeding up the SfM (problem 3).

Bibliography

[1] Elias N Malamas, Euripides GM Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. A survey on industrial vision systems, applications and tools. *Image and vision computing*, 21(2):171–188, 2003.

[2] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 24(2):237–267, 2002.

[3] Waymo. Waymo. `https://waymo.com`, 2013.

[4] Google. Atap project tango. `https://pix4d.com`, 2014.

[5] Pix4D. Pix4dmapper. `https://pix4d.com`, 2011–2017.

[6] ProViDE. Planetary robotics vision data exploitation. `http://www.provide-space.eu`, 2013.

[7] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[8] Jared Heinly, Johannes Lutz Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[10] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[12] Fabian Langguth, Kalyan Sunkavalli, Sunil Hadap, and Michael Goesele. Shading-aware multi-view stereo. In *European Conference on Computer Vision*, pages 469–485. Springer, 2016.

[13] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[14] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014.

[15] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3248–3255, 2013.

[16] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others. Openmvg. an open multiple view geometry library. `https://github.com/openMVG/openMVG`.

[17] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[18] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[19] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[20] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[21] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

[22] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443, 2006.

[23] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[24] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

[25] Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael Rabbat, and Hervé Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Transactions on Big Data*, 2017.

[26] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586. ACM, 2011.

[27] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

[28] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *BMVC*, volume 2, page 2008, 2008.

[29] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[30] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. *Computer Vision–ECCV 2008*, pages 500–513, 2008.

[31] Martin Bujnák. Algebraic solutions to absolute pose problems. *Ph. D. dissertation. Czech Technical University, Prague.*, 2012.

[32] Changchang Wu. P3. 5p: Pose estimation with unknown focal length. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2440–2448, 2015.

[33] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pages 29–42. Springer, 2010.

[34] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Multicore bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3057–3064. IEEE, 2011.

[35] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[36] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[37] Kin Leong Ho and Paul Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3):261–286, 2007.

[38] Manfred Klopschitz, Christopher Zach, Arnold Irschara, and Dieter Schmalstieg. Generalized detection and merging of loop closures for video sequences. In *Proc. 3D Data Processing, Visualization, and Transmission*, volume 2, 2008.

[39] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós. A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, 57(12):1188–1197, 2009.

[40] Jose Henrique Brito, Roland Angst, Kevin Koser, and Marc Pollefeys. Radial distortion self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1368–1375, 2013.

[41] Cenek Albl, Akihiro Sugimoto, and Tomas Pajdla. Degeneracies in rolling shutter sfm. In *European Conference on Computer Vision*, pages 36–51. Springer, 2016.

[42] James Kirchner. Data analysis toolkit: Uncertainty analysis and error propagation. University of California, 2016.

[43] Maxime Lhuillier and Mathieu 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.

[44] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[45] Wolfgang Förstner and Bernhard P Wrobel. *Photogrammetric Computer Vision*. Springer, 2016.

[46] Ken-ichi Kanatani and Daniel D Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5):2017–2028, 2001.

[47] Kenichi Kanatani. *Statistical optimization for geometric computation: theory and practice*. Courier Corporation, 2005.

[48] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[49] Lukas Polok, Viorela Ila, and Pavel Smrz. 3d reconstruction quality analysis and its acceleration on gpu clusters. 2016.

[50] Calyampudi Radhakrishna Rao, Calyampudi Radhakrishna Rao, Mathematischer Statistiker, Calyampudi Radhakrishna Rao, and Calyampudi Radhakrishna Rao. *Linear statistical inference and its applications*, volume 2. Wiley New York, 1973.

[51] Fuzhen Zhang. *The Schur Complement and Its Applications*. Springer US, 2005.

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

[53] Raman Balasubramanian, Sukhendu Das, and Krishnan 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.

[54] Abdelkrim Belhaoua, Sophie Kohler, and Ernest Hirsch. Error evaluation in a stereovision-based 3d reconstruction system. *EURASIP Journal on Image and Video Processing*, 2010(1):1, 2010.

[55] Joachim Höhle and Michael 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.

[56] Philipp Schaer, Jan Skaloud, S Landtwing, and Klaus Legat. Accuracy estimation for laser point cloud including scanning geometry. In *Mobile Mapping Symposium 2007, Padova*, number TOPO-CONF-2008-015, 2007.

[57] Alexis H Rivera-Rios, Fai-Lung Shih, and Michael 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.

[58] Soon-Yong Park and Murali Subbarao. A multiview 3d modeling system based on stereo vision techniques. *Machine Vision and Applications*, 16(3):148–156, 2005.

[59] Michal Polic and Tomas Pajdla. Uncertainty computation in large 3d reconstruction. In *Scandinavian Conference on Image Analysis*, pages 110–121. Springer, 2017.

[60] Maple 2016. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. `http://matlab.com`.

[61] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[62] Martin A Fischler and Robert 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.

[63] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3.3. http://eigen.tuxfamily.org, 2010.

[64] Shweta Jain-Mendon and Ron 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.

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

[66] Neil Muller, Lourenço Magaia, and Ben M Herbst. Singular value decomposition, eigenfaces, and 3d reconstructions. *SIAM review*, 46(3):518–545, 2004.

[67] Simon J Julier. The scaled unscented transformation. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, pages 4555–4559. IEEE, 2002.