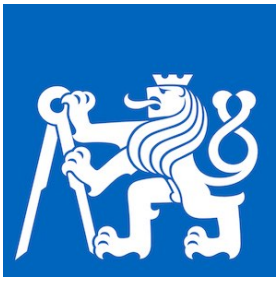
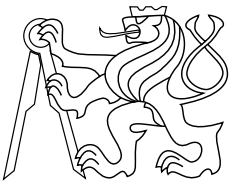
CENTER FOR
MACHINE PERCEPTION



CZECH INSTITUTE
OF INFORMATICS
ROBOTICS AND
CYBERNETICS



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD THESIS

Uncertainty in Structure from Motion Algorithms

PhD Thesis

Michal Polic

michal.polic@cvut.cz

February 28, 2023

Available at
people.ciirc.cvut.cz/~policmic

Thesis Advisor: doc. Ing. Tomáš Pajdla, Ph.D.
Thesis Co-advisor: RNDr. Zuzana Kúkelová, Ph.D.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement ARtwin No. 856994, and SPRING No. 871245.

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

Abstract

Computer vision algorithms, such as Structure from Motion, Simultaneous Localization and Mapping, and Multi-View Stereo, generate three-dimensional scenes for a wide range of applications in industry and entertainment (e.g., robot navigation, self-driving cars, and virtual reality). Although a significant amount of research has been devoted to maximum likelihood estimates of scene parameters, advanced statistical inputs are not typically incorporated, even though they have the potential to improve the speed, accuracy, and robustness of 3D reconstruction. This thesis aims to provide a comprehensive guide on modeling and utilizing uncertainty in Structure from Motion. Specifically, the thesis focuses on describing the uncertainty of image keypoints and affine regions, i.e., the estimate of the detector uncertainty. We present a new approach for better modeling the keypoints' covariance matrices and the positional uncertainty of the affine regions. Next, the uncertainties of feature point transformations are estimated using a large-scale dataset of homographies. This allows us to create the first estimate of the orientation and scale uncertainty of detected regions by the SIFT detector. The thesis presents a new general scheme for propagating uncertainty in minimal camera geometry estimation problems, along with a library of related functions. The challenges of uncertainty propagation through the projection function are addressed and overcome using two developed methods. Finally, the thesis applies uncertainty propagation to minimal problems, demonstrating the speedup of the robust model estimator. Moreover, a new accuracy-based criterion for camera model selection is presented and tested, along with an extension that benefits from multiple reprojection error thresholds. Selecting of an appropriate camera model results in a more accurate and faster reconstruction. In summary, the thesis is the guide to modeling, propagating, and utilizing uncertainty in Structure from Motion.

Abstrakt

Algoritmy počítačového vidění, jako jsou Structure from Motion, Simultaneous Localization and Mapping a Multi-View Stereo, se používají k vytváření trojrozměrných scén pro širokou škálu aplikací v průmyslu a zábavě (např. navigaci robotů, autonomní vozidla a virtuální realitu). V minulosti bylo věnováno odhadům parametrů 3D rekonstrukce významné úsilí, nicméně zahrnutí pokročilejších statistických dat je využito zřídka i přes potenciál zlepšit kvalitu 3D rekonstrukce. Tato disertační práce si klade za cíl vytvořit komplexního průvodce modelováním a využití neurčitostí v rámci Structure from Motion algoritmů. Konkrétně se práce zaměřuje na popis neurčitostí obrazových bodů a afinních oblastí, tj. odhad nejistoty detektorů. Práce představuje nový přístup pro lepší modelování kovariančních matic detekovaných bodů a přesnější odhad neurčitosti pozice afinních oblastí. Dále jsou neurčitosti transformací mezi detekovanými body odhadnuty pomocí vytvořeného datasetu homografií, což umožnilo odvodit první odhad neurčitosti orientace a měřítka regionů detekovaných SIFT detektorem. Práce navíc prezentuje nový obecný postup pro propagaci neurčitostí v minimálních problémech a knihovnu funkcí, které tuto neurčitost propagují. Propagace neurčitosti prostřednictvím projekční funkce je navržena pomocí dvou nových metod. Na závěr prezentuje disertační práce příklad využití neurčitostí při řešení minimálních problémů, tj. demonstruje zrychlení algortimu, který počítá robustní odhad řešení. Dále je představeno a testováno nové statistické kritérium pro výběr modelu kamer na základě jejich neurčitostí. Výběr vhodného modelu kamery vede k přesnější a rychlejší rekonstrukci. Disertační práce poskytuje návod, jak modelovat, propagovat, a využít neurčitosti v rámci Structure from Motion.

Authorship

I hereby confirm that all the results presented in this dissertation were achieved through my own research. The content of individual sections was developed based on research publications that I published in cooperation with co-authors of these papers, my thesis supervisor doc. Ing. Tomáš Pajdla, Ph.D., and my co-supervisor RNDr. Zuzana Kúkelová, Ph.D.

Acknowledgements

I would like to express my sincere gratitude to my colleagues and friends at CIIRC for their invaluable support throughout my research. I am deeply thankful to my supervisor, Tomáš Pajdla, for his excellent guidance, exceptional ideas, and endless patience in helping me overcome various challenges. I am also grateful to my co-advisor and friend, Zuzana Kúkelová, whose mathematical expertise, support, and helpful feedback have been instrumental in shaping my work. I am also indebted to Wolfgang Förstner, who introduced me to uncertainty modeling and propagation and collaborated closely with me on the topics covered in this thesis, offering unique and valuable insights into various challenges. Lastly, but certainly not least, I am thankful to my family, friends, and especially my loving wife, who have supported me throughout this long and arduous journey.

I gratefully acknowledge the support of the European Union's Horizon 2020 research and innovation programme under grant agreement ARtwin No. 856994, and SPRING No. 871245.

Contents

1 Notation

Abbreviations

AC	Accuracy-based Criterion / Affine Correspondence (if clear from context)
ACS	Accuracy-based Camera Model Selection
AIC	Akaike Information Criterion
ANN	Approximate Nearest Neighbors
CDF	Cumulative Distribution Function
IC	Information Criterion
KL	Kullback-Leibler (distance)
LACS	Learned Accuracy-based Camera Model Selection
LSM	Least Squares Matching
MC	Monte Carlo (simulation)
MDL	Minimum Description Length
MP	Moore-Penrose (inversion)
MVS	Multi-View Stereo
PC	Point Correspondence
PDF	Probability Density Function
RANSAC	Random Sample Consensus
SfM	Structure from Motion
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SPRT	Sequential Probability Ratio Test
SVD	Singular Value Decomposition

General variables

$\mathtt{A}_i$	i-th affine transformation of neighbourhoods $\boldsymbol{u}_i \rightarrow \boldsymbol{u}'_i$
$\widetilde{\mathtt{A}}_i$	i-th reference affine transformation of neighbourhoods $\boldsymbol{u}_i \rightarrow \boldsymbol{u}'_i$
$\mathtt{A}_{u_i}$	matrix $\in \mathbb{R}^{2\times 2}$ transforming the unit circle to an affine region boundary (related to the keypoint $\boldsymbol{u}_i$)
$\hat{\boldsymbol{C}}_l^{(i)}$	estimated l-th camera center using camera model $\mathbf{M}_i$
δ	reprojection threshold
$\hat{\boldsymbol{\epsilon}}^{(i)}$	estimated reprojection error $\boldsymbol{u} - \boldsymbol{p}^{(i)}(\hat{\boldsymbol{\theta}}^{(i)})$ using the camera model $\mathbf{M}_i$
$\boldsymbol{e}_{\text{vec,l}}$	Euler vector of the l-th camera (a rotation axis multiplied by a rotation angle)
$\mathtt{E}$	essential matrix
$\mathtt{E}_k$	identity matrix of the dimension $\mathbb{R}^{k\times k}$
f_l	focal length of the l-th camera
$\mathtt{F}$	fundamental matrix
$\mathtt{G}_\mathtt{S}$	multivariate Gaussian using a matrix $(\mathtt{A}_{u_i}\mathtt{A}_{u_i}^\top)/9$ as the kernel

$\mathtt{G}_{x,t}$	1D differentiation kernel in x direction using standard deviation t
$\mathtt{G}_{y,t}$	1D differentiation kernel in y direction using standard deviation t
$h^{(i)}$	radial distortion function of the camera model $\mathbf{M}_i$
$\mathtt{H}$	homography matrix
$\mathtt{H}_\theta$	nullspace of the column space $\mathtt{J}$
χ^2_N	chi-squared distribution of N degrees of freedom
$\mathtt{I}$	grayscale image $\mathtt{I} \in \mathbb{R}^{w \times h}$ (using width: w [px], and height: h [px])
$\mathtt{I}_{x,t}$	smooth image gradient in the x direction of an image $\mathtt{I}$, using a standard deviation t for the differentiation kernel
$\mathtt{I}_{y,t}$	smooth image gradient in the y direction of an image $\mathtt{I}$, using a standard deviation t for the differentiation kernel
$\mathtt{J}$	Jacobian matrix of projection equations estimated in $\hat{\boldsymbol{\theta}}$
$K^{(i)}$	number of parameters in the reconstruction $\hat{\boldsymbol{\theta}}^{(i)}$
$\mathtt{K}_l$	calibration matrix of the l-th camera
$\mathcal{L}$	loss function of a reprojection error
$L^{(i)}$	number of cameras $\hat{\boldsymbol{P}}^{(i)}$
$\mathcal{M}$	set of n camera models $\mathcal{M} = \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_n\}$
$\boldsymbol{m}_{l,m}$	keypoint in a camera coordinate system $\in \mathbb{R}^3$, related to $\hat{\boldsymbol{P}}_l$, and $\hat{\boldsymbol{X}}_m$
$M^{(i)}$	number of 3D points $\hat{\boldsymbol{X}}^{(i)}$
$\mathbf{M}_i$	i-th camera model
$\mathbf{M}_{B\|D}$	radial distortion camera model with B polynomial and D division parameters
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Gaussian distribution (defined by the mean $\boldsymbol{\mu}$ and covariance matrix Σ)
N	number or keypoints $\boldsymbol{u}$ ($N^{(i)}$ is the number of registered inliers in $\boldsymbol{\theta}^{(i)}$)
$\Omega(\hat{\boldsymbol{\theta}})$	weighted squared reprojection error
$p^{(i)}$	projection equation according to the camera model $\mathbf{M}_i$ constraints
$\boldsymbol{P}_l$	l-th camera composed of $\{f_l, \boldsymbol{u}_{\text{pp,l}}, \boldsymbol{e}_{\text{vec,l}}, \boldsymbol{t}_l\}$
$\mathtt{R}_{kj}$	rotation matrix from the camera $\mathtt{P}_j$ to $\mathtt{P}_k$, i.e., $\mathtt{R}_{kj} = \mathtt{R}_k \mathtt{R}_j^\top$
$\widetilde{\mathtt{R}}_{kj}$	reference rotation matrix from the camera $\mathtt{P}_j$ to $\mathtt{P}_k$, i.e., $\mathtt{R}_{kj} = \mathtt{R}_k \mathtt{R}_j^\top$
$\mathtt{R}_l$	rotation matrix of the l-th camera
$\mathcal{S}$	set of tuples $(l, m) \in \mathcal{S}$, every tuple (l, m) contains indexes of the m-th 3D point visible in the l-th camera
$\boldsymbol{\theta}$	reconstruction defined us $\{\boldsymbol{P}, \boldsymbol{X}, \boldsymbol{\theta}_{rd}\}$
$\hat{\boldsymbol{\theta}}^{(i)}$	estimated reconstruction $\{\hat{\boldsymbol{P}}^{(i)}, \hat{\boldsymbol{X}}^{(i)}, \hat{\boldsymbol{\theta}}^{(i)}_{rd}\}$ for the camera model $\boldsymbol{M}_i$
$\boldsymbol{\theta}_{rd}$	radial distortion parameters of a reconstruction
$\boldsymbol{t}_{kj}$	translation vector from the camera $\mathtt{P}_j$ to $\mathtt{P}_k$, i.e., $\boldsymbol{t}_{kj} = \boldsymbol{C}_k - \boldsymbol{C}_j$
$\widetilde{\boldsymbol{t}}_{kj}$	reference translation vector from camera $\mathtt{P}_j$ to $\mathtt{P}_k$, i.e., $\boldsymbol{t}_{kj} = \boldsymbol{C}_k - \boldsymbol{C}_j$
$\boldsymbol{t}_l$	translation vector of the l-th camera ($\boldsymbol{t}_l = -\mathcal{R}_e(\hat{\boldsymbol{e}}_{\text{vec,l}})\boldsymbol{C}_l$)
$\boldsymbol{u}$	vector of all keypoints $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$ in an image coordinates

$\boldsymbol{u}_i$	i-th keypoint in an image coordinates, i.e., $\boldsymbol{u}_i \in \mathbb{R}^2$ [px]
$\boldsymbol{u}_{l,m}$	keypoint in an image coordinates, related to $\hat{\boldsymbol{P}}_l$, and $\hat{\boldsymbol{X}}_m$
$\hat{\boldsymbol{u}}_{l,m}^{(i)}$	estimated projection $\boldsymbol{p}^{(i)}(\hat{\boldsymbol{P}}_l^{(i)}, \hat{\boldsymbol{X}}_m^{(i)}, \hat{\boldsymbol{\theta}}_{rd}^{(i)})$ for the camera model $\boldsymbol{M}_i$
$\boldsymbol{u}_{\text{pp,l}}$	principal point of the l-th camera
$\boldsymbol{X}_m$	m-th point in 3D

Operators

abs	absolute value of an entity (e.g., a scalar, or vector elements)
a2h	converts a vector from affine coordinates into homogeneous coordinates
cond	condition number
dim	dimension of an entity (e.g., set, tuple, list, vector, matrix)
$\mathbb{D}$	dispersion operator
eig	eigenvalue decomposition
$\mathbb{E}$	expectation operator
h2a	convert a vector in homogeneous coordinates into affine coordinates
med	median of an entity (e.g., a set, tuple, list, vector, matrix)
null	nullspace of a matrix (i.e., the nullspace of the column space of a matrix)
qr	QR decomposition
$\mathcal{R}_e$	function converting an Euler vector $\boldsymbol{e}_{\text{vec}}$ to the rotation matrix $\in \mathbb{R}^{3\times3}$
rank	maximal number of linearly independent columns of a matrix
svd	SVD decomposition
tr	trace of a matrix
vec	reshape an input entity to the vector by the column-wise concatenation
$\|.\|$	determinant of a matrix
$\|\|.\|\|_2$	Euclidean norm
$[\boldsymbol{v}]_\times$	skew-symmetric matrix of a vector $\boldsymbol{v} \in \mathbb{R}^3$ in the form $\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$
$()^+$	MP inversion of a matrix
$*$	convolution operator
$\odot$	Hadamard product (i.e., the element-wise multiplication)

Key concepts section

γ	damping term added to a Fisher information matrix
$\bar{h}$	constraints of a model parameters (e.g., $\|\|\text{vec}(\mathtt{H})\|\|_2 = 1$)
H_0	zero hypothesis
$\bar{\mathtt{H}}$	derivative of a model parameter constraints $\bar{h}$
$\lambda_{\text{rd,l}}$	first coefficient of a division radial distortion model of the l-th camera
$\mu_{\underline{x}}$	mean value of the random variable $\underline{x}$
m_r	(raw or central) moment of the r-th degree

$p(x)$	probability density function, i.e., $dP_{\underline{x}}(x)/dx$
$P(a_i)$	probability of an event $a_i \in \mathcal{S}_\text{e}$
$\mathtt{\Sigma}$	covariance matrix
$\mathcal{S}_\text{e}$	space of events
$\mathtt{S}_{\bar{h}}$	matrix fixing the gauge of a covariance matrix using $\bar{h}$ constraints
$\underline{x}$	random variable mapping events to real values, i.e., $\underline{x} : \mathcal{S}_\text{e} \rightarrow \mathbb{R}$

Uncertainty of measurements section

$N_{\mathcal{N}_i}$	number of pixels in an image region described by the matrix $\mathtt{A}_i$
$\mathcal{N}_u$	neighbourhood of the keypoint $\boldsymbol{u}_i$
r_m	redundancy number for the 3D point $\boldsymbol{X}_m$
r_{u_i}	radius of a circular region related to the keypoint $\boldsymbol{u}_i$
$\mathtt{R}_{u_i u_i}$	redundancy matrix $\in \mathbb{R}^{2\times 2}$ of the keypoint $\boldsymbol{u}_i$
σ_n	standard deviation of a pixel intensities noise (called image noise)
σ_{u_i}	standard deviation of the i-th keypoint assuming isotropy ($\mathtt{\Sigma}_{u_i u_i} = \sigma_{u_i}\mathtt{E}_2$)
$\sigma_{u_{i,1}}$	standard deviation of the first coordinate of the keypoint $u_{i,1}$
σ_x	standard deviation of a scalar x
$\mathtt{\Sigma}_{u_i u_i}$	covariance matrix of the keypoint $\boldsymbol{u}_i$
s_{u_i}	i-th keypoint scale
t	standard deviation of smoothing and differentiation Gaussian kernel
$\mathtt{W}_{u_i u_i}$	Fisher information matrix $\in \mathbb{R}^{2\times 2}$ of the keypoint $\boldsymbol{u}_i$
$\hat{\boldsymbol{y}}_{l,m}^{(i)}$	normalized estimate of the reprojection error $\hat{\boldsymbol{\epsilon}}_{l,m}^{(i)}$ using camera model $\mathbf{M}_i$, i.e., $\underline{\hat{\boldsymbol{y}}}^{(i)} \in \mathcal{N}(\mathbf{0}, \mathbf{1})$

Measurement transformation uncertainty section

α_{u_i}	angular transformation of the i-th correspondence $\phi_i' - \phi_i$
$\widetilde{\alpha}_{u_i}$	reference angular transformation of the i-th correspondence derived from related reference homography matrix $\widetilde{\mathtt{H}}$
δ_cond	condition number threshold
$\Delta\alpha_i$	difference between a measured and the reference angular transformation $\widetilde{\alpha}_i - \alpha_i$
Δr_{u_i}	ratio of the scale ratios $r_{u_i}/\widetilde{r}_{u_i}$
ϵ_{u_i}	symmetric positional residual of the i-th correspondence with respect to the ground truth transformation derived from related homography matrix $\widetilde{\mathtt{H}}$
$\boldsymbol{n}_l$	l-th plane normal in the world coordinate system $\in \mathbb{R}^3$
ϕ_i, ϕ_i'	orientation angle of the i-th correspondence, i.e., the feature points in an image pair
p_{i3}, p_{i4}	scalar values that realize sheer transformations derived from the affine transformation $\mathtt{A}_i$

r_{u_i}	scale ratio of the i-th correspondence scales s'_i/s_i
$\widetilde{r}_{u_i}$	reference scale ratio of the i-th correspondence scales derived from the matrix $\widetilde{\mathtt{A}}_i$
s_i, s'_i	scale of the i-th correspondence, i.e., the feature points in an image pair

Uncertainty in SfM

$\epsilon_{\Sigma_{P_l}}$	relative error of the l-th camera covariance matrix w.r.t. $\mathtt{Q}_{\mathrm{PP}}$
$\mathtt{Q}_{\mathrm{PP}}$	mean absolute magnitude of camera parameters

Applications of the uncertainty modelling

$\mathtt{J}_A$	Jacobian matrix of a common set of parameters $\boldsymbol{\theta}_A$ ($\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_B^{(i)}\}$)
$\mathtt{J}_B$	Jacobian matrix of a set of parameters $\boldsymbol{\theta}_B^{(i)}$ ($\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_B^{(i)}\}$)
$\mathtt{S}^{(i)}$	S-transformation fixing the gauge of a information matrix so that a common set of parameters $\boldsymbol{\theta}_A$ is independent of the rest of the parameters
T_1	run time of the fastest sub-reconstruction that registered L cameras
T_d	maximal time limit to register L cameras

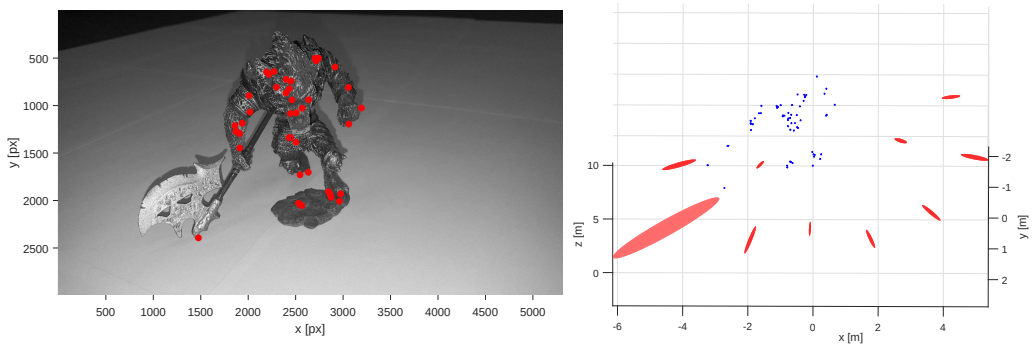
2 Introduction

The digitization of the physical world has garnered significant attention due to its wide range of applications, such as quality verification of industrial products [1], robot navigation [2], localization [3], self-driving cars [4], virtual reality [5] and more [6, 7]. As a result, achieving accurate and robust three-dimensional scene reconstruction is an important objective for many computer vision algorithms, including Structure from Motion (SfM) [8], Simultaneous Localization and Mapping (SLAM) [9], and Multi-View Stereo (MVS) [10]. Recent advancements in this field have shown the potential for reconstructing geometry from vast photo collections [11, 12]. Using a single computer, we can create 3D models of entire cities from pictures taken by consumer cameras. These models can be composed of millions of 3D points and constructed from as many as hundreds of thousands of photos [8, 10].

What is a reconstruction?
A digital representation of a real-world environment is referred to as a 3D reconstruction. It is usually the output of the sparse reconstruction process, explicitly the SfM or SLAM, which produces a collection of camera intrinsics (such as the focal length and radial distortion coefficients), camera poses (representing the position and orientation of the camera), and the coordinates of 3D points.

What to analyse in a reconstruction?
Estimating the 3D reconstruction involves detecting unique points in images loaded with a positional noise. The primary objective of this thesis is to determine how the uncertainty of detected image points (i.e., the input uncertainty) impacts the quality of the estimated reconstruction (i.e., the output uncertainty) and uncover the underlying relationships between reconstruction parameters (such as the focal length and the camera rotation). These characteristics, i.e., relationships and accuracy, can be described by the first few moments of the reconstruction. While SfM and MVS calculate the first moment (i.e., the Maximum Likelihood (ML) estimate of the reconstruction) in most of the reconstruction pipelines [8, 13, 14], this thesis focuses on the second moment (i.e., the covariance matrix) of the 3D reconstruction, which is rarely studied. We present practical methods for estimating, propagating, and utilizing the uncertainty in the scope of the algorithms employed in SfM.

Why to analyse a reconstruction?
The iterative nature of sparse reconstruction methods (SfM, SLAM) means that early-stage errors can significantly impact the final 3D reconstruction. By understanding the hidden relationships between scene parameters and their dependence



(a) (b)

Figure 1: The propagation of the second moment (the covariance matrix) of the keypoints (a) to the second moment of the camera poses (b). The red ellipsoids visualize where are the camera centers likely to be, i.e., the standard ellipsoids of covariance matrices of camera centers. Blue dots correspond to the reconstructed 3D points.

on the uncertainty of input observations, we can improve a number of computer vision tasks, such as selecting the best camera model [**15**][1], weighting or filtering the most unconstrained parameters (e.g., 3D points), planning recording trajectories [16], and speeding up the reconstruction process [**17**]. The knowledge of the uncertainty can help us check iteratively added cameras and prevent incorrect extensions of partial reconstructions, leading to faster and more robust reconstruction pipelines. Additionally, estimating 3D points uncertainty allows more sophisticated smoothing of reconstructed surfaces in dense reconstructions [18,19]. Moreover, accurate relative pose estimates can improve the selection of the first reconstruction pair in sequential SfM. The thesis presents two applications in detail.

[1]The publications of the author of this thesis are highlighted in bold letters for easy distinction from other citations.

3 State of the Art

To investigate the properties of a reconstruction, it is essential to understand the reconstruction process. The mathematical models describing the relationship between different scene parameters (e.g., relative and absolute pose constraints, and projection functions) significantly impact the distribution of inaccuracies within the final 3D model.

One way to analyze the reconstruction process is to propagate the uncertainty from observations to the relative pose parameters, such as the essential or fundamental matrix. Alternatively, it's possible to focus on the accuracy of the reconstruction itself by utilizing the relationships described by projection equations. Both approaches are discussed in detail in this thesis.

This section provides a brief overview of the reconstruction process and summarizes its advantages and drawbacks. We also discuss the information available at the input of standard reconstruction pipelines and the main algorithms used to propagate the input uncertainty into various reconstruction parameters, such as the essential or fundamental matrices and camera poses. Finally, we investigate statistical methods for suitable mathematical model estimation from input observations.

3.1 The reconstruction process

There are two main approaches how to reconstruct a scene. The global ones (e.g. [20, 21]) and the iterative ones (e.g. [8, 22]). This work focuses on the SfM approach, which can be viewed as a generalization of SLAM, i.e., an unordered set of images is assumed, and the pose of the first camera is not fixed. The SfM can be categorized as an iterative method that repeatedly extends a partial reconstruction.

The reconstruction process typically begins with *detecting feature points* in input images. Feature points can be detected using handcrafted techniques, such as SIFT [23], SURF [24], and MESR [25], or using trained detectors, such as SuperPoint [26], D2-Net [27], and R2D2 [28]. In addition to detecting feature points, the detectors typically also describe the local neighborhood around each detected point using a unique feature vector (descriptor), e.g., [23,24]. Once feature points have been detected and described, *tentative matches* between them are established based on an Approximate Nearest Neighbors (ANN) search [29–32]. The similarity between feature points is typically computed as the dot product of their descriptors.

The relationship between correspondences and scene parameters is based on projective geometry [33]. The *relative pose* solver verifies tentative matches using an assumed camera model. For example, if the camera calibration is known (i.e.,

the intrinsic parameters), we can employ the essential matrix solver developed by Nister [34]. Epipolar geometry constraints (for cameras with unknown intrinsic parameters) between tentative correspondences allow for estimating both cameras' relative pose and focal lengths. Moreover, we can assume even more general constraints for tentative matches and calculate the radial distortion parameters as in Kukelova [35]. The camera model constraints, the number of correspondences fitted by the estimated geometrical model, and keypoints uncertainty influence the uncertainty of the estimated model. Assuming one set of calculated parameters (e.g., essential matrix) for a selected camera model, correspondences with a smaller error than a specified threshold are called *inliers*. The error can be algebraic, geometrical, weighted by a loss function, or weighted by the accuracy of the measurements. The minimal solution to the camera geometry estimation problem is usually estimated by an extension of RANSAC [36]. Recently, trainable approaches for selecting inliers (e.g., SuperGlue [37]) have been published. Furthermore, some algorithms (e.g., SparseNCNet [38] or Patch2Pix [39]) combine all the previous steps of relative pose estimation into a single trainable network.

After verifying tentative correspondences in the relative pose estimation step, the SfM algorithm selects an *initial pair* of cameras and establishes the global coordinate system. Typically, this selection is made heuristically based on the number of correspondences and the viewing angle between camera pairs as done by Schoenberger [8]. The iterative part of the algorithm begins by *triangulating* 3D points from the verified feature points in the first camera pair. Next, a new camera is added to the partial reconstruction by solving the *absolute pose* problem, which involves computing the extrinsic camera parameters (i.e., orientation and position) given the input *2D - 3D* correspondences [40, 41]. There are many absolute pose solvers available that can also estimate some of the intrinsic camera parameters, such as the focal length studied in Kukelova [42], principal point published in Larsson [43], or radial distortion [44, 45]. The algorithm iterates between adding new cameras and triangulating new 3D points until all images have been registered or no more correspondences can be utilized.

The above-described process registers images extending the first camera pair, leaving out cameras without enough feature points visible from other views. The estimated camera poses and 3D points are typically refined through an efficient nonlinear optimization after a few iterations (of adding new cameras), using a method called *Bundle Adjustment* (BA) [46]. The optimization is often performed using the Ceres nonlinear least squares solver [47]. The BA minimizes the distance between the feature points and the bundles of rays emanating from the 3D points, creating projections in the images [33]. It is common to run BA at the end of the reconstruction pipeline and optimize the 3D model repeatedly during the reconstruction process.

3.2 Properties of the reconstruction process

Current reconstruction pipelines usually utilize SfM algorithms, which can handle an unordered set of input images [8, 13, 14, 22, 48]. This approach benefits from the large number of improvements that have been made over the past few decades. There are numerous advancements in the robust model estimation technique, as well as relative and absolute pose solvers for different parameters, such as radial distortion, tangential distortion, focal length, or rolling shutter. Additionally, multiple implementations of each solver exist based on different geometric relationships, such as angles between rays, distances between 3D points, ratios of distances between 3D points, and others.

One of the main drawbacks of the iterative SfM approach is that it can lead to a local optimum, which is particularly evident in the "loop closing" problem [49,50]. In this problem, a loop of tens or more cameras often ends in a different position than where it started. Current reconstruction pipelines [8, 13, 14] optimize the reconstruction whenever few cameras are registered to partial reconstruction to decrease accumulated camera drift. However, such an approach slows down the reconstruction process [20, 21] and works only partially.

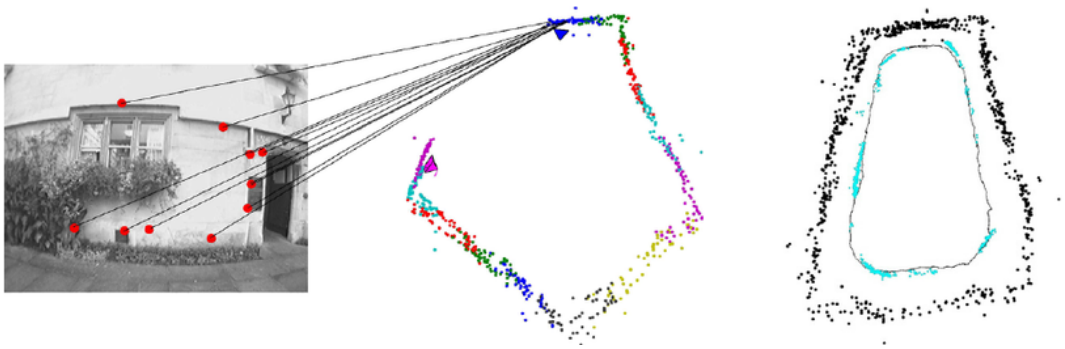


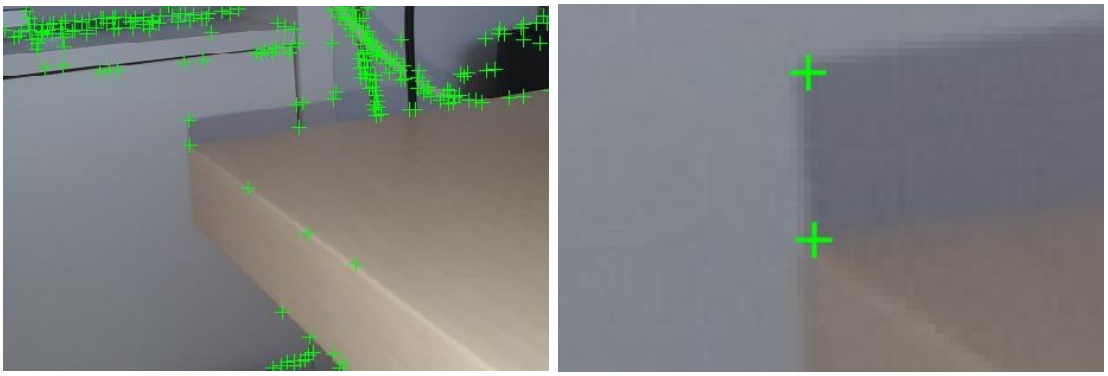
Figure 2: This figure visualizes sparse reconstruction created from images like the one on the left subfigure. The images were recorded on a trajectory following a loop, shown in the right subfigure. Yet the result of SfM (middle subfigure) leads to a loop closure problem because of accumulated drift. This figure is from Wilson [51].

3.3 The statistics of a reconstruction

The reconstruction process begins with the detection of feature points. While many algorithms focusing on this task have been published and extensively compared [52], only a few studies have evaluated the uncertainty of observations. This

subsection provides an overview of the main approaches used to describe the uncertainty of the keypoints (i.e., the coordinates of centers of detected regions in the images). The subsequent subsections discuss methods for propagating the uncertainty from measurements to estimated parameters and techniques that focus on the propagation of uncertainty in the context of SfM. Lastly, this section discusses uncertainty utilization to improve the robustness and accuracy of the reconstruction.

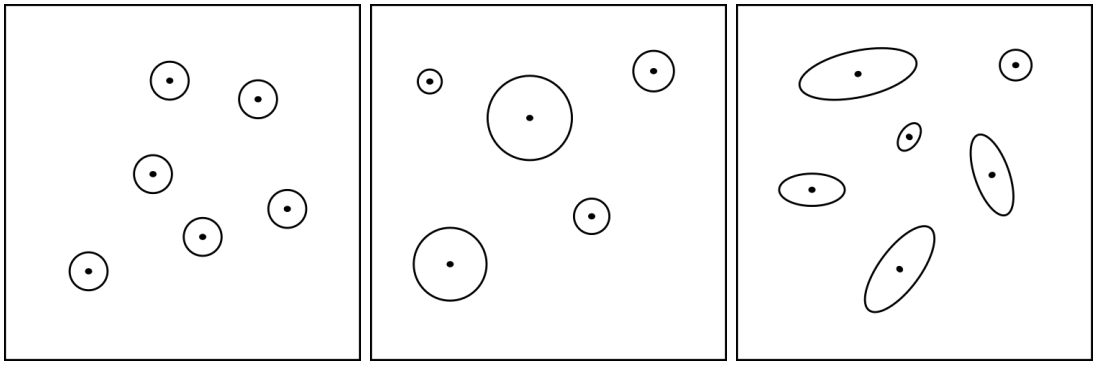
Uncertainty of the observations. We can illustrate the basic properties of feature point detectors with a simple example of applying the corner detector, such as the Harris operator [53], on an image of a desk (see Fig. 3). Does the Harris operator detect the corner of the desk? $\rightarrow$ No, it cannot. The reason is



(a) Original image (b) Detail of detected corner

Figure 3: The feature points detected by Harris operator [53] on a image of a desk.

that detectors work on top of 2D images rather than 3D space. These detectors select unique patches, such as corners or blobs in 2D, and only approximate the projections of unique points in 3D as shown by Kanatani [54]. The detection depends on various factors such as material properties, environmental conditions, camera parameters, and image noise. Moreover, as each keypoint is unique, no positional covariance matrix can be measured by standard statistical approaches. The detection operator always leads to the same feature points in a single image, making it impossible to measure their variance directly. The uncertainty of detected keypoints can be estimated by *uncertainty analysis of template matching* of image intensities. Assuming an ideal camera with a linear transfer function, image noise variance increases linearly with pixel intensity [55,56]. In practice, estimating the variance function for each keypoint coordinate is complex due to the internal camera processing. As far as we know, no publications discuss the uncertainty of the remaining parameters of the feature points, e.g., the scale and orientation



(a) (b) (c)

Figure 4: Visualization of the standard ellipsoids for Gaussian models of keypoints noise: (a) isotropic homogeneous, (b) isotropic inhomogeneous, (c) anisotropic inhomogeneous

of SIFT detector. The estimate by Monte Carlo (MC) simulation would require multiple samples with different lighting, reflection, contrast, etc., for all camera poses, which cannot be achieved in practice.

Most of the authors and reconstruction pipelines [8, 22, 52, 57] assume *isotropic homogeneous* noise model of keypoints, Fig. 4a. This model assumes the same uncertainty for all keypoints. The *isotropic inhomogeneous* noise model visualized in Fig. 4b can be described by covariance matrices composed of a unit matrix scaled by variances of individual keypoints [16, 58, 59]. Many authors estimate these variances from a reprojection error depending on the selected camera model, as noted in Kanatani [54]. Unfortunately, the uncertainty of the keypoint does not depend on the camera model but on lighting, contrast, viewing angle, etc.. This approach is correct only if the camera intrinsics are known in advance. The experimental evaluation conducted by Förstner [60] shows that the standard deviation of the Lowe detector lies, in the case of the known camera model, in the order of the rounding error, i.e., $\approx 1/3$ pixels. The authors [60] also derived an empirical formula for the standard deviation of the SIFT [61] detector based on the scale level in the image scale space pyramid. Images at a higher scale space level can be seen as blurred. Therefore, the standard deviation is higher. Lastly, several methods have been proposed [62–65] to estimate the covariance matrix from grayscale images. It can be splited into residual-based approaches [62,63] and the derivative-based approaches [64,65]. Higher-order estimates of uncertainties of feature points are difficult or impossible to estimate. We discuss the keypoint covariance matrix estimation and the extension of related state-of-the-art methods in Sec. 6.

The uncertainty of the measurements transformation. Traditionally, geometric relations are estimated from keypoint correspondences between the two images [33,66], ignoring that correspondences are often rather established between image regions.

Having the ground truth transformation (e.g., a homography matrix), we can decompose it and evaluate the uncertainty of the transformation of individual elements of feature points (e.g., the keypoint shift, scale coefficient, or change of the orientation). Such an approach approximates the MC simulation for individual feature point transformations by calculating the average variance from all the feature point transformations. As an example, the transformation of keypoints (i.e., their shift) is evaluated (using the classical statistic methods, i.e., calculating the variance of the reprojection error) in most of the papers about minimal solvers, e.g., [43,44,67]. The first uncertainty evaluation of other feature point parameters in **[68]** contains an estimation of the scale and orientation transformation uncertainty for OpenCV implementation of SIFT detector. The main results from **[68]** are shown in Sec. 7.

Suppose the ground truth transformation between measurements is unavailable. In that case, we can use the known uncertainty of the inputs (e.g., the approximated image noise) and ML estimator utilizing these inputs to estimate the transformation parameters (e.g., using the template matching of affine regions to estimate the affinity transformation). As a maximum likelihood estimator, the least squares matching (LSM) provides the covariance matrix of the estimated parameters. The LSM method is, for example, used for refining affine correspondences [69–71] reaching standard deviations for parallaxes down to below 1/100 pixel as shown in Haralick [72]. Intensity-based refinement for pose estimation has been used in papers [73, 74].

The uncertainty propagation. The standard approach for propagating uncertainty from measurements to model parameters using selected constraints can be categorized according to several criteria. For instance, the propagation can be forward or backward, and the system can be linear or nonlinear, over-parameterized or not. Various works have described these different schemes, including [33,60,75]. The uncertainty associated with homography estimation from four or more points can be estimated using methods such as SVD [66, 76] and Lie groups [77]. The uncertainty of an estimated fundamental matrix has been computed using the SVD method presented in Sur [78], as well as the minimal representations approach introduced by Csurka [79]. Similarly, the uncertainty of the essential matrix has been evaluated using a minimal representation method, as demonstrated in work by Förstner [60]. There were also published several specific extensions for computing the uncertainty of various setups, such as lines, proposed by Bal-

asubramanian [80]; edges, studied by Belhaoua [16]; laser scans, investigated in studies by Höhle [81] and Schaer [82]; and stereo setups, explored in studies by Rivera-Rios [59] and Park [83]. However, these authors typically rely on heuristics to approximate covariances instead of following the uncertainty propagation method. To our knowledge, the uncertainty analysis for affine transformations was first published in Barath **[84]**. We introduced there a general scheme for uncertainty propagation in minimal problems, where the constraints are in implicit form extended about constraints on model parameters. Additionally, we provide a practical guide for making affine correspondences work well in camera geometry computation. The uncertainty propagation for minimal problems is further discussed in Sec. 8.1.

The effect of employing covariance matrices in the statistically optimal estimation of the Homography and Fundamental matrix was evaluated in Kanazawa [85]. The authors concluded that employing accuracy for relative pose calculation improved the results, but the effect was negligible for Harris [53] and SUSAN [86] detectors. However, several hyperparameters, such as the size of the differentiation and smoothing kernel, are missing in the experimental evaluation of this publication. Therefore, we could not replicate provided experiments. Conversely, recent papers have highlighted the importance of accurate localization of feature points [26–28, 38, 61] and their influence on reconstruction accuracy, as shown in Lindenberger [87], and localization, as studied in Zhou [88].

The uncertainty of a reconstruction. Propagating the uncertainties of input measurements to the reconstruction parameters, such as camera poses and 3D point positions, is challenging. The relationship between 3D points and their projections is described by a suitable projection function, such as those discussed in [33, 43, 44]. The nonlinear projection function is usually simplified by its first-order approximation using its Jacobian matrix [33,89], and the uncertainty is propagated in a backward manner from the measurements to the parameters using a nonlinear over-parameterized system of equations defined by appropriate projection functions. However, the projection equations are typically over-parameterized and do not fully constrain the 3D scene, as mentioned by Morris [90]. This means that the reconstruction can be shifted, rotated, and scaled without changing the image projections.

Hartley [33] suggested a three-step approach for propagating uncertainty in an over-parameterized system: 1) mapping the parameter space to the set of *essential parameters*, 2) computing the inverse of the information matrix of the essential parameters, and 3) mapping the inverse information matrix back to the original parameter space. However, this approach becomes challenging for large-scale reconstruction, where finding the essential parameters is difficult. As a result, this

approach is often only applicable to minimal solvers, as the one in Förstner [60], rather than the reconstruction itself.

Kanatani [75] describes the uncertainties in the context of changing regularisation conditions, called gauge transformations. The uncertainty of a parameter is infinite if it can be adjusted freely without changing the reprojection error. Therefore, we are primarily interested in estimating *inner geometry* (e.g., angles and ratios of distance) and its *inner precision*, defined in Förstner [60]. Inner precision is invariant to gauge changes. A natural choice of the fixation of gauge, which leads to the uncertainty of inner geometry, is to fix seven degrees of freedom caused by the invariance of the projection function [33, 60, 75].

There are many different choices of regularisation conditions. For example, SLAM [9] assumes a fixed camera pose and fixed scale of the first image pair baseline, which makes the Fisher information matrix full rank. In the case of SLAM, fast Cholesky decomposition can be used to invert a Schur complement matrix, as well as other techniques for fast covariance matrix computation [91, 92]. Some papers, such as from Ila [93] or Polok [94], claim to address uncertainty computation in SfM. Nevertheless, they assume a full-rank Fisher information matrix and do not deal with gauge of estimated covariance matrix.

The gauge-free approach and the *normal form of the covariance matrix* were introduced by Kanatani [75]. One way to estimate it is to use the Moore-Penrose (MP) inversion of the Fisher information matrix, as studied by Förstner [60]. The pseudoinverse A^+ of A is equal to the inverse of A on the range of A, and sends the orthogonal complement of the range of A to the zero vector, as described in Ben [95]. Note that the residual vector is perpendicular to the range of A, and thus the pseudoinverse minimizes the sum of the squared Mahalanobis distances of the residuals to the zero vector.

Lhuillier proposed a method to speed up the MP inversion of the information matrix for SfM frameworks by decomposing the information matrix and computing the MP pseudoinverse of the Schur complement of the submatrix of 3D point parameters. This submatrix has the same size as the block of camera parameters and is much smaller than entire information matrix. The decomposition was also extended in Polok [94]. However, this decomposition does not satisfy the rank additivity condition defined in Tian [96]. Lhuillier provided proof of the existence of a correction term that allows the use of this decomposition in his work [58]. However, there is no connection between the proof and the correction term actually used.

The papers **[97–99]** address the aforementioned challenges. Two main approaches are elaborated in this thesis. The first approach **[98]** involves adding a damping term to the information matrix. Next, the Taylor expansion is used to

estimate the inversion of the information matrix at a point where the damping term equals zero. This strategy leads to an iterative scheme for estimating the uncertainty of the reconstruction. The second approach [**99**], involves extending the information matrix about the nullspace of the Jacobian of the projection equations. This approach offers a faster and more robust one-step solution estimating the covariance matrix. Further details are in Sec. 8.2.

The reconstruction statistics utilization can be done in many ways, and we will highlight two approaches: The first approach is to use the uncertainty of the estimated model to improve relative pose estimation. This is demonstrated in Barath [**84**]. The authors used uncertainty propagation for the early rejection of degenerated models. The second approach employs uncertainty to compare the quality of several reconstructions built from the same images but using different camera models. This is shown in [**15**]. The authors utilized the estimated covariance matrix to compare several reconstructions with the goal of identifying which camera model leads to the most accurate and reliable reconstruction.

The related work to the first application is the following. The uncertainty of relative pose estimation assumes the input to be either keypoints [24, 26–28, 61] or affine correspondences [100–103]. We derived the uncertainty for several geometrical problems and found that the estimated uncertainty is related to the number of inliers found during the model verification. Therefore, we could employ the probability of having a suitable relative pose model in the SPRT [104], leading to faster convergence. Related work to uncertainty propagation using an essential set of parameters of the relative pose transformation is mentioned above and studied in [60, 77, 79]. However, we utilized a new scheme simplifying the uncertainty propagation derivation leading to the same results.

The related work to the second application is the following. SfM pipelines use many hyperparameters that are hard to set in practice. Generally, the model selection based on various statistics and criteria is a well-studied problem that received considerable attention [54, 105–114]. The Akaike criterion (AIC [105]) is based on the first-order estimate of the Kullback-Leibler (KL) distance between the densities given by the data and true (unknown) density function. AIC computes the likelihood of the fitted model parameters and its bias correction. Hurvich's AICc [106] is a second-order estimate of the KL distance, which can be seen as an extension of AIC for small sample sizes. Takeuchi's TIC [115] is another extension of AIC, which shrinks the model parameters towards the maximum entropy distribution and therefore is more robust if the correct model is not in the set of candidates models. Bozdogan's CAIC [107] adjusts AIC by the assumption that the order of the models does not change if the sample size increases. Schwarz's BIC [108] is motivated by approximating the marginal probability density of the data under the

model, which leads to a higher magnitude of bias correction w.r.t. AIC. Rissanen's MDL [110] is derived from the minimal code length necessary for describing the data. A valuable extension of the AIC, MDL up to geometric G-AIC, G-MDL was introduced by Kanatani in [54]. It highlights that the accuracy depends primarily on the physical properties of the observed 3D structure. All the approaches above do assume observations without outliers. The simplest robust IC is Ronchetti's RAIC [112]. It generalizes the ML-estimator to an M-estimator, which minimizes a robust loss function of the residuals. This idea can be applied to ICs mentioned above, as in, e.g., RBIC [113] and RTIC [114]. Watanabe's WAIC and WBIC [116] assume known priors on the model parameters. However, the camera model selection in SfM by standard Information Criterion (IC) does not work for several reasons. First, the reconstruction has a singular statistical model due to the gauge freedom, i.e., the likelihood function of having a "good" model cannot be derived using the normal distribution, as studied in Watanabe [116]. Secondly, the prior distribution of the reconstruction parameters (e.g., camera poses and 3D points) is unknown. Thus, the Bayesian methods cannot be used either. Third, for different camera models and different reprojection thresholds, the final 3D reconstruction contains different numbers of registered 3D points and cameras, i.e., the size of the data is not constant. Finally, standard ICs assume that residuals depend only on the selected model, which is not the case of reprojection error that also depends on physical properties (e.g., lighting and view angle). The most related work is done by Kanatani [54], who derived the G-AIC and G-MDL information criteria. These criteria were applied in Kinoshita [117] to choose between affine and projective camera models. However, G-AIC and G-MDL methods do not work well for the camera model selection task because this task has a singular statistical model as described in Watanabe [116]. Another approach to radial distortion model selection was presented in [118,119]. That approach assumes correspondences between planar calibration boards with a fixed number of detected observations without considering any outliers and simplifies used camera models to homographies between pairs of images.

4 Contribution of the Thesis

Modern SfM pipelines [10, 22] and recent minimal solvers [33, 34, 74, 120–122] assume the isotropic homogeneous noise of the observations, treating feature points indistinguishably. Utilizing uncertainties can improve the accuracy of estimated parameters and avoid unnecessary computations, such as the verification of degenerated solutions of minimal problems and extending reconstructions about inaccurate cameras. This thesis provides a comprehensive review of state-of-the-art literature and methods for working with uncertainties in SfM. We begin with feature point uncertainty estimation, followed by uncertainty propagation to algebraic solutions of minimal problems and sparse reconstructions. Finally, we present two applications that benefit from estimated uncertainty. This work is the guideline for working with uncertainty in SfM. In particular, our contributions are the following:

1. The thesis **summarizes the key concepts** of uncertainty modeling, uncertainty propagation, gauges, information criteria, camera models, the relationship between cameras, and optimization of the reconstruction. This is an important background that allows readers to understand the utilization of uncertainty in SfM.

2. The thesis **unifies theory about modelling the keypoints noise**, summarizes the state-of-the-art approaches, and **presents the first visual comparison** of their results.

3. We present a **new extension** of the keypoint noise modeling that allows us to improve the **estimate of a covariance matrix for keypoints**.

4. We present a **new generalization** of keypoint uncertainty modeling that allows us to **estimate the accuracy of affine region** position.

5. The thesis presents how to estimate the **uncertainty of feature point transformations between images** derived from a large-scale dataset of homographies [**68**]. We extend this publication to include **the first estimate of the SIFT scale and orientation uncertainty**.

6. Next, we **propose a generalized approach for the propagation of uncertainty for minimal problems**. It simplifies the derivation of the uncertainty propagation for individual minimal problems by employing the constraints between parameters. The theory was published in [**17**]. This thesis extends the publication by **creating a software of uncertainty propagation functions for several minimal solvers and verifying their robustness** at `https://github.com/michalpolic/pose_uncertatinty_lib`.

7. We present two approaches for the **uncertainty propagation from key-points to reconstructions**. The Taylor expansion algorithm was presented in the paper [**98**], and the following Nullspace bounding method in [**99**]. The thesis summarizes the most relevant results from these papers. The Taylor expansion algorithm is the first approach that allows expressing the uncertainty of the inner geometry of a large-scale reconstruction correctly, while the Nullspace bounding method has superior robustness and faster execution time.

8. The thesis presents **an application of uncertainty estimates in minimal problems as an initialization for the SPRT test**. We empirically found that the distribution of the high-inlier ratio, defined as the ratio of found inliers to the maximal number of inliers for the calculated model, depends on the model uncertainty. Using this dependency, we initialize the SPRT test with the probability of having a high-inlier ratio model estimated. It speeds up the computationally extensive verification step of the robust model estimator. The results were published in [**17**].

9. The last contribution shows the benefits of the uncertainty propagation to the reconstruction. We present **the first accuracy-based criterion** that works on **automatic camera model selection task**. This method was published in [**15**] and leads to superior reconstruction accuracy and faster execution times.

4.1 Scientific publications

Here we list all the publications of the author of this thesis for the completeness and relation to the content of this thesis. The papers whose contributions are used in the thesis:

- Barath, D., Mishkin, D., Polic, M., Förstner, W., & Matas, J. (2023). A Large-Scale Homography Benchmark. Conference on Computer Vision and Pattern Recognition.
- Barath, D., Polic, M., Förstner, W., Sattler, T., Pajdla, T., & Kukelova, Z. (2020). Making affine correspondences work in camera geometry computation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16 (pp. 723-740). Springer International Publishing.
- Polic, M., Steidl, S., Albl, C., Kukelova, Z., & Pajdla, T. (2020). Uncertainty based camera model selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5991-6000).
- Polic, M., Forstner, W., & Pajdla, T. (2018). Fast and accurate camera covariance computation for large 3d reconstruction. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 679-694).
- Polic, M., & Pajdla, T. (2017, October). Camera uncertainty computation in large 3D reconstruction. In 2017 International Conference on 3D Vision (3DV) (pp. 282-290). IEEE.
- Polic, M., & Pajdla, T. (2017). Uncertainty computation in large 3d reconstruction. In Image Analysis: 20th Scandinavian Conference, SCIA 2017, Tromsø, Norway, June 12–14, 2017, Proceedings, Part I 20 (pp. 110-121). Springer International Publishing.

The papers not utilized in the thesis:

- Dubenova, M., Zderadickova, A., Kafka, O., Pajdla, T., & Polic, M. (2022, September). D-InLoc++: Indoor Localization in Dynamic Environments. In Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings (pp. 246-261). Cham: Springer International Publishing.
- Albl, C., Kukelova, Z., Larsson, V., Polic, M., Pajdla, T., & Schindler, K. (2020). From two rolling shutters to one global shutter. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2505-2513).

4.2 Thesis Outline

The thesis is a comprehensive guideline for modeling, propagating, and utilizing uncertainty in Structure from Motion (SfM). It starts with describing key mathematical concepts and variables, then discusses uncertainty in image observations. Then, the uncertainty propagation to algebraic solutions of minimal problems and the reconstruction are derived. The thesis concludes by presenting two applications that benefit from the uncertainty of estimated parameters. In more detail, the thesis is organized as follows:

- Section 5 describes the *key concepts*, including statistical basics and geometrical relationships between cameras, points, and observations. This section introduces the necessary mathematical notation of the main entities utilized further and describes key relations between them. The first subsection discusses ways of *modeling the uncertainty* and introduces the approach we are using in the following text. After that, we focus on *functions of random variables*, i.e., the uncertainty propagation using linear, non-linear, and implicit functions. It is followed by the description of the *coordinate systems and gauges* in which the uncertainty is expressed. The next subsection discusses the main statistical ideas on how to estimate a suitable mathematical model for given data, i.e., the *information criterion for model selection*. The rest of this section focuses on the geometrical aspects of SfM. We show the description of the most important *camera models*, related projection functions, mathematical notation of the camera, 3D points, observations, and the key geometrical relationships between them. Further, we describe the constraints of the *relative pose* problem, i.e., the geometry between a pair of cameras. The last subsection shows the *bundle adjustment* optimization of the reconstruction.

- Section 6 provides a comprehensive overview of the state-of-the-art techniques for estimating the *uncertainty of the measurements*. The first subsection focuses on the estimation of *keypoints uncertainty using circular regions*, which covers how to estimate the variance or covariance matrix of the keypoints. We discuss three different noise models for keypoint coordinates, namely isotropic homogeneous, isotropic inhomogeneous, and anisotropic inhomogeneous models. We also extend the anisotropic inhomogeneous noise model by normalizing the expectation of the weighted squared residuals. This extension is further adapted for the description of *keypoints uncertainty using affine regions* and is followed by an *evaluation* of all the approaches.

- Section 7 outlines our approach for estimating the *measurement transformation uncertainty*, such as the angular transformation variance for a pair

of feature points. We describe the creation of a large-scale dataset of homographies that are used to calculate the *positional transformation*, *scale transformation*, and *angular transformation*. By measuring the differences between the derived ground truth transformations and the measured ones, we can determine the variances of the transformations. Furthermore, the variance of individual parameters is propagated to the uncertainty of the feature points. The section concludes with an *evaluation* of the presented approach.

- Section 8 presents methods for estimating the *uncertainty in SfM*. The section is divided into two main parts. The first part introduces a generalized approach for propagating the uncertainty of minimal problems. This approach explains how to use the uncertainty of feature points to estimate the uncertainty of the algebraic solution of a minimal problem. The second part discusses possible approaches for propagating uncertainty from keypoints to the reconstruction, including camera extrinsic and intrinsic parameters, as well as 3D points. This section describes the Taylor expansion algorithm and the Nullspace bounding method. Finally, all of these approaches are *evaluated* in the last subsection.

- Section 9 showcases two applications of uncertainty modeling. The first one is an *uncertainty-based robust model estimator*, which involves utilizing the probability of having a high inlier ratio as initialization of the SPRT test to speed up the computationally expensive verification step of a RANSAC-based method. The second application presents how to detect the camera model from input images automatically. This is achieved by introducing an accuracy-based criterion (AC) that evaluates the "goodness" of the mathematical model for given data. Further, two methods: accuracy-based camera model selection (ACS) and learned accuracy-based camera model selection (LACS), estimate the camera model for input images. The last subsection, *evaluation*, demonstrates the improvements the two uncertainty modeling applications gained.

- Section 10 presents the *conclusion*, i.e., summarize the content of the thesis.

5 Key concepts

This section provides a brief overview of the basic concepts and methods that are utilized and further extended in the subsequent sections. The first part introduces the term uncertainty and presents several mathematical models for describing it. We then discuss how to propagate uncertainty from measurements to the output of a function and present information criteria for evaluating how well a mathematical model fits the observed data. The second part focuses on the geometrical constraints applied in computer vision, particularly the Structure from Motion algorithm. We begin by describing a camera and then proceed to discuss the relationship between pairs of cameras and the relationship between cameras and points in 3D. Finally, we describe the optimization method used to minimize the reprojection errors.

5.1 Modeling the uncertainty

In order to describe the accuracy or uncertainty of an event, such as the noise added to a feature point, several concepts must be introduced. The most common definition of probability is based on Kolmogorov's Axiomatic Definition, which assumes a space $\mathcal{S}_\text{e}$ of events $A_i \in \mathcal{S}_\text{e}$, each with a probability of occurrence $P(A_i)$, and the following axioms:

$$P(A_i) \geq 0 \tag{1}$$

$$P(\mathcal{S}_\text{e}) = 1 \tag{2}$$

$$P(A_1 \cup A_2) = P(A_1) + P(A_2) \qquad \text{if } A_1 \cap A_2 = \emptyset. \tag{3}$$

When the outcome events a_i from an experiment are non-numerical, they can be described by a function $\underline{x} : \mathcal{S}_\text{e} \to \mathbb{R}$ that maps events to real numbers $\underline{x} = \underline{x}(a_i)$. This function is called a *random variable* and describes the whole experiment, while $x(a_i)$ (without the underscore) describes the outcome of one specific trial. The random variable is regularly denoted by $\underline{x}$, omitting the events $a_i \in \mathcal{S}_\text{e}$.

There are two commonly used methods for describing a random variable, namely the *cumulative distribution function* (CDF) $P_{\underline{x}}(x) = P(\underline{x} < x)$ and the *probability density function* (PDF) $p_{\underline{x}}(x) = dP_{\underline{x}}(x)/dx$. If the measurements are in continuous space, we typically choose $\underline{x}(x) = x$ and omit the lower index of CDF and PDF if it is clear from the context. In other words, we use $P(x)$ and $p(x)$ instead.

The normal or Gaussian distribution is a well-known example of a probability density function $p(x)$. It approximates a sum of a large number of independent,

identically distributed random variables with bounded variance, shown by Papoulis [123]. For a single variable $\underline{x} \sim \mathcal{N}(\mu, \sigma^2)$, the density function is given by

$$p(x) = g(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \tag{4}$$

Here, μ and σ describe the mean and variance of this symmetric distribution function. For a k-dimensional vector of variables (e.g., the reconstruction $\boldsymbol{x} := \boldsymbol{\theta}$), the joint probability density function is a multi-dimensional Gaussian distribution $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, defined by the density function

$$p(\boldsymbol{x}) = g(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^k \Sigma}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}. \tag{5}$$

where $\boldsymbol{\mu}$ is the mean vector (e.g., the ML estimate of the reconstruction $\boldsymbol{\mu} := \hat{\boldsymbol{\theta}}$), and Σ is the covariance matrix (e.g., the covariance matrix of estimated parameters $\Sigma := \Sigma_{\hat{\theta}\hat{\theta}}$). This distribution is useful for modeling uncertainties in a wide range of computer vision tasks, including SfM.

Modeling the uncertainty of feature points using the Gaussian distribution allows describing the anisotropic inhomogeneous noise, shown in Figure 4. This means that the accuracy of each feature point is described by its own $\mathbb{R}^{2\times2}$ covariance matrix. The covariance matrix of the maximum likelihood estimate can be approximated by its Cramer-Rao lower bound, which is the inverse of the Fisher information matrix. Kanazawa [85] has shown that the accuracy of a feature point is proportional to the structure tensor of the image intensity function in the feature point neighborhood, and can be derived from template matching. Concretely, the structure tensor is a symmetric positive semi-definite matrix $\mathbb{R}^{2\times2}$ describing the covariance matrix of a keypoint up to a noise variance factor. In other words, a Multivariate Gaussian distribution can describe the keypoints uncertainty as the reconstruction uncertainty. Moreover, assuming independently detected feature points, we can compose one common covariance matrix for all keypoints as a block-diagonal matrix of $\mathbb{R}^{2\times2}$ covariance matrices for individual keypoints.

To approximate the parameters of a Gaussian distribution for a set of events or measurements, we can use the first few moments. For feature points, since there is only one measurement for each, we typically assume isotropic noise or the structure tensor of the image intensity [54, 58, 85]. Moments can be defined for

both continuous and discrete measurements, i.e.

$$m_r = \int_{x=-\infty}^{\infty} (x-c)^r p(x) dx \qquad r \geq 0 \tag{6}$$

$$m_r = \sum_{i=1}^{\infty} (x_i - c)^r P(x_i) \qquad r \geq 0 \tag{7}$$

and can be either *raw moments* or *central moments*. Raw moments have $c = 0$, while central moments have $c = \mathbb{E}(\underline{x})$, where $\mathbb{E}$ is the *expectation operator*. For continuous or discrete probability distributions, the expectation operator is defined as:

$$\mu_{\underline{x}} \doteq \mathbb{E}(\underline{x}) = \int_{x=-\infty}^{\infty} x \; p(x) dx \tag{8}$$

$$\mu_{\underline{x}} \doteq \mathbb{E}(\underline{x}) = \sum_{i=1}^{\infty} x_i P(x_i). \tag{9}$$

The raw moment of the first degree can be used to estimate the mean of the Gaussian distribution, i.e., for $r = 1$ and vector of random variables $\boldsymbol{\mu}_{\underline{\boldsymbol{x}}} \doteq \mathbb{E}(\underline{\boldsymbol{x}}) = \boldsymbol{m}_1$. Further, the central moment of the random variable (or variables) of rth degree and $c = \mathbb{E}(\underline{x})$ (or $\boldsymbol{c} = \mathbb{E}(\underline{\boldsymbol{x}})$) approximates the variance (or covariance matrix) of the Gaussian (or Multi-Dimensional Gaussian) distribution

$$\sigma^2_{\underline{x}} = \int_{x=-\infty}^{\infty} (x - \mathbb{E}(\underline{x}))^2 p(x) dx \tag{10}$$

$$\Sigma_{\underline{x}\underline{x}} = \mathbb{E}[(\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}}))(\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}}))^\top]. \tag{11}$$

5.2 Functions of random variable

The uncertainty of feature points can be modeled by a random variable with a suitable probability distribution. This probability distribution can be further propagated to the parameters of the SfM. In general, we assume a vector of random variables $\underline{\boldsymbol{x}}$, which is described by the probability density function $p(\boldsymbol{x})$, and a function $\boldsymbol{y} = f(\boldsymbol{x})$. The goal is to derive the probability density function $p(\underline{\boldsymbol{y}})$. We focus on the propagation of Multi-Dimensional Gaussian distribution that is commonly used to describe the uncertainty of keypoints [58, 60, 85]. This section briefly describes the propagation of the probability distribution for *linear*, *non-linear*, and *implicit functions of random variables*.

If we assume a *linear function* that express $\boldsymbol{y}$ from parameters $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})$, we can write $\boldsymbol{y} = f(\boldsymbol{x})$ as function

$$\underline{\boldsymbol{y}} = \mathtt{A}\underline{\boldsymbol{x}} + \boldsymbol{b} \tag{12}$$

leading to the $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \Sigma_{yy})$ distribution with the mean value

$$\begin{aligned} \mathbb{E}(\underline{\boldsymbol{y}}) &= \mathtt{A}\mathbb{E}(\underline{\boldsymbol{x}}) + \boldsymbol{b} && (13) \\ \boldsymbol{\mu}_y &= \mathtt{A}\boldsymbol{\mu}_x + \boldsymbol{b} && (14) \end{aligned}$$

and the covariance matrix

$$\begin{aligned} \Sigma_{yy} &= \mathbb{E}[(\underline{\boldsymbol{y}} - \mathbb{E}(\underline{\boldsymbol{y}}))(\underline{\boldsymbol{y}} - \mathbb{E}(\underline{\boldsymbol{y}}))^\top] \\ &= \mathbb{E}[(\mathtt{A}\underline{\boldsymbol{x}} - \mathtt{A}\mathbb{E}(\underline{\boldsymbol{x}}))(\mathtt{A}\underline{\boldsymbol{x}} - \mathtt{A}\mathbb{E}(\underline{\boldsymbol{x}}))^\top] \\ &= \mathtt{A}\mathbb{E}[(\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}}))(\underline{\boldsymbol{x}} - \mathbb{E}(\underline{\boldsymbol{x}}))^\top]\mathtt{A}^\top \\ &= \mathtt{A}\Sigma_{xx}\mathtt{A}^\top. \end{aligned} \tag{15}$$

Thus the random variables $\underline{\boldsymbol{y}}$ have distribution $\underline{\boldsymbol{y}} \sim \mathcal{N}(\mathtt{A}\boldsymbol{\mu}_x + \boldsymbol{b}, \mathtt{A}\Sigma_{xx}\mathtt{A}^\top)$ for linear function $\underline{\boldsymbol{y}} = \mathtt{A}\underline{\boldsymbol{x}} + \boldsymbol{b}$ and $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})$.

Usually, we have a set of estimated parameters $\hat{\boldsymbol{x}}$ (e.g., the reconstruction $\hat{\boldsymbol{x}} := \hat{\boldsymbol{\theta}}$) and the *non-linear function* $\underline{\boldsymbol{y}} = f(\underline{\boldsymbol{x}})$ (e.g., the projection function $f := p$). In this case, the function f can be replaced by its linearized version, i.e.,

$$\underline{\boldsymbol{y}} \approx f(\hat{\boldsymbol{x}}) + \mathtt{J}d\underline{\boldsymbol{x}} \qquad \textit{where} \quad \mathtt{J} = [\mathtt{J}_j] = \frac{\partial f(\hat{\boldsymbol{x}})}{\partial x_j}. \tag{16}$$

It is beneficial to extend this notation to multiple functions $\boldsymbol{f} = \{f_1, \ldots, f_I\}$ (e.g., the projection equations for individual keypoints $\boldsymbol{f} := \boldsymbol{p}$) and write

$$\underline{\boldsymbol{y}} \approx \boldsymbol{f}(\hat{\boldsymbol{x}}) + \mathtt{J}d\underline{\boldsymbol{x}} \qquad \textit{where} \quad \mathtt{J} = [\mathtt{J}_{ij}] = \frac{\partial f_i(\hat{\boldsymbol{x}})}{\partial x_j}. \tag{17}$$

Then, the first-order approximation of the density function $p(\underline{\boldsymbol{y}})$ equals

$$\mathbb{E}(\underline{\boldsymbol{y}}) \approx \boldsymbol{\mu}_y = \boldsymbol{f}(\boldsymbol{\mu}_x) \qquad \mathbb{D}(\underline{\boldsymbol{y}}) \approx \Sigma_{yy} = \mathtt{J}\Sigma_{xx}\mathtt{J}^\top \tag{18}$$

In short, we can write $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{f}(\boldsymbol{\mu}_x), \mathtt{J}\Sigma_{xx}\mathtt{J}^\top)$ for nonlinear function $\underline{\boldsymbol{y}} = \boldsymbol{f}(\underline{\boldsymbol{x}})$ and $\underline{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_{xx})$.

The propagation of probability distribution described above is known as the forward propagation of uncertainty. Assuming that $\boldsymbol{f}$ represents the set of projection functions, eq. (18) propagates the uncertainty of parameters (e.g., the camera poses) to the observations (i.e., the keypoints). For the sake of simplicity, we also introduce the concept of *backward propagation of uncertainty*, which propagates the probability distribution from observations to the parameters (e.g., from the

keypoints to the reconstruction). In the case of linear function $\underline{\boldsymbol{y}} = \mathtt{A}\underline{\boldsymbol{x}} + \boldsymbol{b}$ and $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \Sigma_{yy})$ the random variables $\underline{\boldsymbol{x}}$ has distribution

$$\underline{\boldsymbol{x}} \sim \mathcal{N}\left(\mathtt{A}^{-1}\boldsymbol{\mu}_y - \boldsymbol{b}, \left(\mathtt{A}^\top \Sigma_{yy}^{-1} \mathtt{A}\right)^{-1}\right). \tag{19}$$

We can also extend this formula for the *non-linear function*, i.e., $\underline{\boldsymbol{y}} \approx \boldsymbol{f}(\hat{\boldsymbol{x}}) + \mathtt{J} d\underline{\boldsymbol{x}}$ assuming $\underline{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \Sigma_{yy})$. The random variable $\underline{\boldsymbol{x}}$ propagated by non-linar function $\boldsymbol{f}$ has distribution

$$\underline{\boldsymbol{x}} \sim \mathcal{N}\left(\boldsymbol{f}^{-1}(\boldsymbol{y}), \left(\mathtt{J}^\top \Sigma_{yy}^{-1} \mathtt{J}\right)^{-1}\right). \tag{20}$$

Note that the function $\boldsymbol{f}^{-1}(\boldsymbol{y})$ does not exist when the system of equations is not regular, as is the case with the projection functions in SfM.

The following paragraph is related to the propagation of probability distribution using *implicit functions*. In computer vision, many geometric relationships, such as relative pose constraints, are in the form of implicit functions $f(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = \mathbf{0}$, and it can be difficult to express them in the form of $\underline{\boldsymbol{y}} = \boldsymbol{f}(\underline{\boldsymbol{x}})$. For example, the Homography matrix $\mathtt{H} \in \mathbb{R}^{3\times 3}$ constraints can be written as

$$\begin{bmatrix}\boldsymbol{u}_2^\top & 1\end{bmatrix} \mathtt{H} \begin{bmatrix}\boldsymbol{u}_1 \\ 1\end{bmatrix} = 0 \tag{21}$$

for each pair of keypoints $\boldsymbol{u}_1 \in \mathbb{R}^2$,$\boldsymbol{u}_2 \in \mathbb{R}^2$ in the first and second image. However, deriving $\mathtt{H} = \boldsymbol{f}(u_1, u_2)$ is more complicated. Using the Taylor expansion, i.e., the first-order approximation, we can write

$$d\boldsymbol{f}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = \mathtt{A}_{\mathrm{x}}\, d\underline{\boldsymbol{x}} + \mathtt{B}_{\mathrm{y}}\, d\underline{\boldsymbol{y}} = \mathbf{0} \qquad \textit{where} \quad \mathtt{A}_{\mathrm{x}} = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{x}}; \mathtt{B}_{\mathrm{y}} = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{y}}. \tag{22}$$

If the matrix $\mathtt{B}_{\mathrm{y}}$ is invertible, then $d\underline{\boldsymbol{y}} = \mathtt{B}_{\mathrm{y}}^{-1}\mathtt{A}_{\mathrm{x}}\, d\underline{\boldsymbol{x}}$ and we have linearized formula transforming $\underline{\boldsymbol{x}} \to \underline{\boldsymbol{y}}$. Therefore, the covariance of $\underline{\boldsymbol{y}}$ leads to

$$\Sigma_{yy} = \mathtt{B}_{\mathrm{y}}^{-1}\mathtt{A}_{\mathrm{x}}\Sigma_{xx}\mathtt{A}_{\mathrm{x}}^\top \mathtt{B}_{\mathrm{y}}^{-\top} \tag{23}$$

5.3 Coordinate systems and gauges

Geometrical constraints in computer vision are typically defined up to some degree of freedom. The reconstruction's random variables are subject to two gauges: the *gauge of the coordinate system* and the *gauge of the covariance matrix*. Changing the coordinate system gauge, i.e., the origin, rotation, and scale of basis vectors, is called a *K-transformation*. In contrast, the change of the covariance matrix

gauge is called an *S-transformation*. It is essential to consider these gauges, as the choice of coordinate system can affect numerical stability. For instance, the scale of keypoints can be chosen in pixels or pixels multiplied by the inverse of the image width. Although both coordinate systems satisfy the same geometrical constraints, the second one is more numerically stable. The focal length in pixels has a larger variance than the other camera parameters, leading to a large range of values in the Jacobian matrix of the projection equation. These values are squared in the information matrix (see eq. (18)). The gauge of the covariance matrix is critical for a unique description of the uncertainty. If we scale, rotate, or shift the camera poses and 3D points, the image projections remain the same, making the uncertainty of the parameters infinite. Assuming seven parameters fixed, e.g., the first camera pose is at the origin, the rotation is the unit matrix, and the baseline length equals one (as in SLAM), the covariance matrix of camera centers increases with the distance from the first camera. Such a gauge of the covariance matrix is misleading because we do not expect that cameras in SfM have larger uncertainty if they are far from the first camera. Therefore, it is crucial to properly fix the gauge of the covariance matrix to correspond to the uncertainty of inner geometry.

The transformation of the coordinate system is straightforward. We focus on the transformation of the gauge of the covariance matrix. Let us assume a set of equally weighted constraints, such as $\bar{\boldsymbol{h}}(\hat{\boldsymbol{x}}) = 0$ for $\underline{\boldsymbol{x}} \sim \mathcal{N}(\hat{\boldsymbol{x}}, \Sigma_{\hat{x}\hat{x}})$. These constraints are applied only to the parameters and fix all degrees of freedom. The Jacobian of these constraints can be expressed as:

$$\bar{\mathtt{H}} = \frac{\partial \bar{\boldsymbol{h}}(\hat{\boldsymbol{x}})}{\partial \boldsymbol{x}} \,. \tag{24}$$

This Jacobian allows us to express the S-matrix, which fixes the gauge of the covariance matrix such that $\bar{\boldsymbol{h}}(\hat{\boldsymbol{x}}) = 0$. The S-matrix and transformed covariance matrix are defined as:

$$\mathtt{S}_{\bar{h}} = \mathtt{I} - \bar{\mathtt{H}}(\bar{\mathtt{H}}^\top \bar{\mathtt{H}})^{-1} \bar{\mathtt{H}}^\top \tag{25}$$

$$\overline{\Sigma}_{xx} = \mathtt{S}_h \Sigma_{xx} \mathtt{S}_{\bar{h}}^\top \,. \tag{26}$$

More details about S-transformation can be found in Baarda [124] and Forstner [60].

5.4 Information criterion for model selection

In order to perform a reconstruction, we need to choose constraints between measurements and parameters. For example, the camera model may assume a pinhole, polynomial, or division radial distortion model. The relative pose constraints may

be realized by a homography, essential, or fundamental matrix. However, methods for parameter estimation (such as minimal solvers) and optimization (such as gradient descent) are typically designed on top of a single geometric model, which is selected manually [8, 22, 48, 57]. To automate this process, we need to consider key concepts for model selection based on input data.

Hypothesis testing can be carried out using several statistical methods. If the distribution of residuals based on a hypothesis is not consistent with the residuals calculated from measurements and estimated parameters to a significant degree, the hypothesis is rejected. The statistical criterion for testing geometric hypotheses, such as testing that the variance of a normally distributed squared residuals is consistent with a known variance, can be formulated in the form of a chi-squared test (χ^2). Mathematically, assuming the null hypothesis $H_0 : \underline{x} \sim \mathcal{N}(\mu_x = 0, \sigma_x^2)$ (e.g., $\underline{x}$ is a random variable of residuals between projections of 3D points and corresponding observations), the variance of the reprojection error can be empirically determined based on the detection level in the scale pyramid of Lowe detector [60]. The sum of normalized squared residuals, given by

$$\underline{X}^2(\underline{x}) = \frac{\Sigma_{n=1}^{N}(\underline{x}_n - \mu_x)^2}{\sigma^2} \tag{27}$$

should follow the χ_N^2 distribution with N degrees of freedom if the hypothesis H_0 holds. If the value of $\underline{X}^2(\underline{x})$ is larger than the critical value $\chi_{\alpha,N}^2$, the hypothesis is rejected with a significance level of α.

There are several disadvantages of hypothesis testing. Firstly, while a hypothesis can be rejected, it cannot be accepted, and we cannot prove that the mathematical model is suitable. Secondly, we require knowledge of the accuracy of the sensor (such as related noise of feature point locations) and the specified significance level (such as 1% or 5%). However, these parameters are not always known in advance. Thirdly, the standard statistical approach assumes a study of asymptotic properties in the limit of a large number of data points, such as all reprojection errors. However, when focusing on the distribution of individual feature points, we have only a single measurement and cannot even compute such an essential measure as the sample average. Therefore, the usage of standard statistical methods for model selection is limited as described by Kanatani [125].

Another approach for selecting a suitable model is to use a heuristic measuring the "goodness" of a fit of the data to a mathematical model concerning some statistical property. For example, one such heuristic is the Akaike information criterion (AIC), which measures the amount of information lost by a model, as published in Akaike [105]. Another heuristic is the minimum description length (MDL), which measures the minimal length of the code necessary for describing

the data, as shown by Rissanen [110]. These types of heuristics are collectively known as "information criteria" (IC). The basic ideas of the well-known ICs are described in Sec. 3. The Accuracy-based Criterion **[15]** is discussed in Sec. 9.

5.5 Camera models

The camera device captures light reflected from the objects in its view frustum and forms a 2D image. This process is described by geometric constraints between elements in 3D space and their correspondences in the image. The constraints reflect physical properties such as focal length, principal point, or lens distortion. The commonly used constraints to mathematically describe the cameras are called *camera models* $\boldsymbol{M}_i \in \mathcal{M}$. In fact, we assume n camera models $\mathcal{M} = \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_n\}$ for the purpose of their comparison. The best fitting camera model to the observed data is called $\boldsymbol{M}_b$. The 3D reconstruction, denoted by $\boldsymbol{\theta}^{(i)}$, for a single camera model $\boldsymbol{M}_i$ is built from the observations $\boldsymbol{u}^{(i)}$ in the images. The vector of observations assumes only the inliers that are employed in the reconstruction. In the most general form, we assume the projection function $\boldsymbol{p}^{(i)}$ that projects the 3D points into the images, expressed as

$$\boldsymbol{u}^{(i)} = \boldsymbol{p}^{(i)}(\boldsymbol{\theta}^{(i)})\,. \tag{28}$$

In practice, we estimate the reconstruction $\hat{\boldsymbol{\theta}}^{(i)}$ from measured observations $\boldsymbol{u}^{(i)}$ using techniques such as SFM [8], Theia [57], RealityCapture [48], or OpenMVG [22]. However, due to the complexity of the problem, the projections $\hat{\boldsymbol{u}}^{(i)}$ of 3D points do not generally coincide with the observed keypoints $\boldsymbol{u}^{(i)}$, resulting in a reprojection error (as seen in equation eq. (44)). Furthermore, some projections may not have any measured counterparts in all images. To account for this, we assume an index set $\mathcal{S}$ that determines which points are visible in each camera. The reconstruction of the camera model $\boldsymbol{M}_i$ can be mathematically formulated as

$$\hat{\boldsymbol{\theta}}^{(i)} = \{\hat{\boldsymbol{P}}^{(i)}, \hat{\boldsymbol{X}}^{(i)}, \hat{\boldsymbol{\theta}}_{rd}^{(i)}\} \tag{29}$$

It consists of $M^{(i)}$ 3D points $\hat{\boldsymbol{X}}^{(i)} = \{\hat{\boldsymbol{X}}_1^{(i)}, \hat{\boldsymbol{X}}_2^{(i)}, \ldots, \hat{\boldsymbol{X}}_{M^{(i)}}^{(i)}\}$, $L^{(i)}$ cameras $\hat{\boldsymbol{P}}^{(i)} = \{\hat{\boldsymbol{P}}_1^{(i)}, \hat{\boldsymbol{P}}_2^{(i)}, \ldots, \hat{\boldsymbol{P}}_{L^{(i)}}^{(i)}\}$, and the radial distortion parameters $\hat{\boldsymbol{\theta}}_{rd}^{(i)}$. Single projection $\hat{\boldsymbol{u}}_{l,m} \in \mathbb{R}^2$ of point $\hat{\boldsymbol{X}}_m \in \mathbb{R}^3$ in the image plane related to camera $\hat{\boldsymbol{P}}_l$ is described by

$$\hat{\boldsymbol{u}}_{l,m}^{(i)} = \boldsymbol{p}^{(i)}(\hat{\boldsymbol{P}}_l^{(i)}, \hat{\boldsymbol{X}}_m^{(i)}, \hat{\boldsymbol{\theta}}_{rd}^{(i)}) \qquad \forall (l,m) \in \mathcal{S}. \tag{30}$$

where l is the index of the camera and m is the index of the 3D point. In the following text, the camera $\hat{\boldsymbol{P}}_l \in \mathbb{R}^9$ is composed of the focal length $\hat{f}_l \in \mathbb{R}$, the

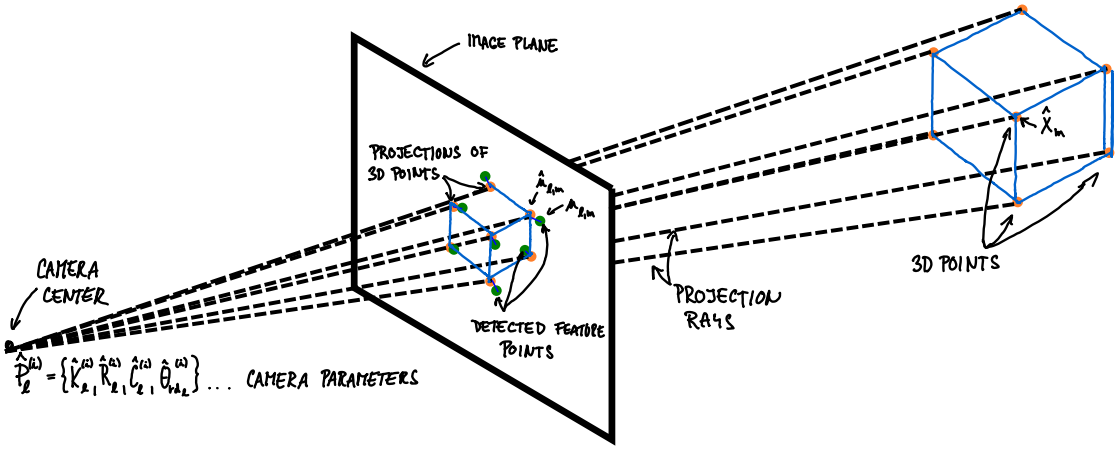


Figure 5: Visualization of camera parameters and projection of the 3D points.

principal point $\boldsymbol{u}_{\text{pp,l}} \in \mathbb{R}^2$, Euler vector $\hat{\boldsymbol{e}}_{\text{vec,l}} \in \mathbb{R}^3$ (i.e., a rotation axis multiplied by a rotation angle, which can be transformed into the rotation matrix by $\mathcal{R}_e(\hat{\boldsymbol{e}}_{\text{vec,l}}) \in \mathbb{R}^{3\times3}$), and the translation $\hat{\boldsymbol{t}}_l \in \mathbb{R}^3$. Assuming one camera model $\boldsymbol{M}_i$, the index $()^{(i)}$ and the indices $()_{l,m}$ are skipped whenever it is clear from context.

The radial distortion parameters $\hat{\boldsymbol{\theta}}_{rd}$ are applied on projections realized by the simple pinhole camera model to reflect the lens distortion. In general, the distortion function $\boldsymbol{h}^{(i)}$ depends on the distance $r_{\text{dist,k}}$ of the k-th keypoint $\boldsymbol{u}_k$ from the distortion center. For simplicity, let's assume that the distortion center is at the principal point $\hat{\boldsymbol{u}}_{\text{pp}}$. The projection function $\boldsymbol{p}^{(i)}$ for the camera model $\boldsymbol{M}_i$ with radial distortion can be expressed as

$$\hat{\boldsymbol{u}}_k^{(i)} = \hat{f}\, \boldsymbol{h}^{(i)}(\hat{r}^2_{\text{dist,k}}, \hat{\boldsymbol{\theta}}_{rd})\tilde{\boldsymbol{u}}_k + \hat{\boldsymbol{u}}_{\text{pp}}\,, \tag{31}$$

where $\hat{r}^2_{\text{dist,k}} = ||\tilde{\boldsymbol{u}}_k||^2$ and $\tilde{\boldsymbol{u}}_k$ is the projection of m-th 3D point before applying radial distortion

$$\tilde{\boldsymbol{u}}_k = \begin{bmatrix} \tilde{u}_{k,1} \\ \tilde{u}_{k,2} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{X}}_{1,m}/\tilde{\boldsymbol{X}}_{m,3} \\ \tilde{\boldsymbol{X}}_{2,m}/\tilde{\boldsymbol{X}}_{m,3} \end{bmatrix}. \tag{32}$$

The 3D point $\tilde{\boldsymbol{X}}_m$ is in the camera coordinates obtained by rotating and translating the point $\hat{\boldsymbol{X}}_m$

$$\tilde{\boldsymbol{X}}_m = \mathcal{R}_e(\hat{\boldsymbol{e}}_{\text{vec}})\hat{\boldsymbol{X}}_m + \hat{\boldsymbol{t}}\,. \tag{33}$$

The radial distortion function is usually modeled as a rational function [126], i.e.

$$\boldsymbol{h}^{(i)}(\hat{r}^2_{\text{dist,k}}, \hat{\boldsymbol{\theta}}_{rd}) = \frac{1 + \hat{k}_1\hat{r}^2_{\text{dist,k}} + \hat{k}_2\hat{r}^4_{\text{dist,k}} \ldots \hat{k}_R\,\hat{r}^{2B}_{\text{dist,k}}}{1 + \hat{d}_1\hat{r}^2_{\text{dist,k}} + \hat{d}_2\hat{r}^4_{\text{dist,k}} \ldots \hat{d}_D\,\hat{r}^{2D}_{\text{dist,k}}}\,, \tag{34}$$

where $\hat{k}_j$ and $\hat{d}_q$ are parameters of the radial distortion model. The most common models are *polynomial (Brown) models* with $\hat{d}_q = 0$, $\forall q$, or *division models* with $\hat{k}_j = 0$, $\forall j$. We denote the radial distortion model eq. (34) with the first B non-zero parameters $\hat{k}_j$ and the first D non-zero parameters $\hat{d}_q$ as $\boldsymbol{M}_{B|D}$. $\boldsymbol{M}_{0|0}$ is the simple pinhole camera model with no radial distortion. Different SfM pipelines use different camera models, e.g. COLMAP [8] uses $\boldsymbol{M}_{0|0}$, $\boldsymbol{M}_{1|0}$, $\boldsymbol{M}_{2|0}$, $\boldsymbol{M}_{3|3}$[2], Meshroom [127] uses $\boldsymbol{M}_{0|0}$, $\boldsymbol{M}_{3|0}$ and Theia [57] uses $\boldsymbol{M}_{0|0}$, $\boldsymbol{M}_{2|0}$.

Observations $\boldsymbol{u}_{l,m}$ that satisfy $\mathcal{L}(\hat{\epsilon}_{l,m}^{(i)}) = \mathcal{L}(||\boldsymbol{u}_{l,m} - \hat{\boldsymbol{u}}_{l,m}^{(i)}||) < \delta$, for some threshold δ, are *the inliers* of the model $\boldsymbol{M}_i$. The $\mathcal{L}$ is a *loss function* to robustify the 3D reconstruction estimate and its optimization.

5.6 Relative pose

The relative pose between two perspective views is an essential part of SfM. Assuming a 3D point $\boldsymbol{X}_m$ projected by two pinhole cameras $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ into keypoints $\boldsymbol{u}_{1,m}$ and $\boldsymbol{u}_{2,m}$ according to eq. (30), the epipolar constrains relate these keypoints in the first and second cameras (using camera model $\boldsymbol{M}_{0|0}$). If we assume intrinsic geometry encapsulated in relative pose geometry, the constraints can be written as an implicit equation given by the *fundamental matrix* $\mathtt{F} \in \mathbb{R}^{3\times3}$ of rank two

$$\begin{bmatrix}\boldsymbol{u}_{2,m}^\top & 1\end{bmatrix} \mathtt{F} \begin{bmatrix}\boldsymbol{u}_{1,m} \\ 1\end{bmatrix} = 0 \tag{35}$$

The fundamental matrix projects points $\boldsymbol{u}_{1,m}$ in homogeneous coordinates into lines $\boldsymbol{l}_{2,m} = \mathtt{F}\begin{bmatrix}\boldsymbol{u}_{1,m} & 1\end{bmatrix}^\top$ in the second image. If the eppipolar constrains holds, the lines $\boldsymbol{l}_{2,m}$ are coincided with points $\boldsymbol{u}_{2,m}$, i.e., for each keypoint and corresponding line holds $\begin{bmatrix}\boldsymbol{u}_{2,m}^\top & 1\end{bmatrix}\boldsymbol{l}_2 = 0$. The fundamental matrix can be further expressed in the terms of camera parameters as

$$\mathtt{F} = \mathtt{K}_2^{-\top}[-\boldsymbol{t}_{21}]_\times \mathtt{R}_{21}\mathtt{K}_1^{-1} = \mathtt{K}_2^{-\top}\mathtt{R}_2[\boldsymbol{C}_2 - \boldsymbol{C}_1]_\times \mathtt{R}_1^\top \mathtt{K}_1^{-1}\,. \tag{36}$$

If the calibration matrices $\mathtt{K}_1$, $\mathtt{K}_2$ are known, we can constrain observations in camera coordinate system $\begin{bmatrix}\boldsymbol{m}_{l,m} & 1\end{bmatrix}^\top = \mathtt{K}^{-1}\begin{bmatrix}\boldsymbol{u}_{l,m} & 1\end{bmatrix}^\top$ by *essential matrix* called $\mathtt{E}$. Note that $\boldsymbol{u}_{l,m}$ is in the image coordinate system, expressed in pixels. Therefore, we get

$$\begin{bmatrix}\boldsymbol{m}_{2,m}^\top & 1\end{bmatrix} \mathtt{E} \begin{bmatrix}\boldsymbol{m}_{1,m} \\ 1\end{bmatrix} = 0 \tag{37}$$

where

$$\mathtt{E} = [-\boldsymbol{t}_{21}]_\times \mathtt{R}_{21}. \tag{38}$$

[2]In COLMAP, $\boldsymbol{M}_{3,3}$ includes also tangential distortion terms.

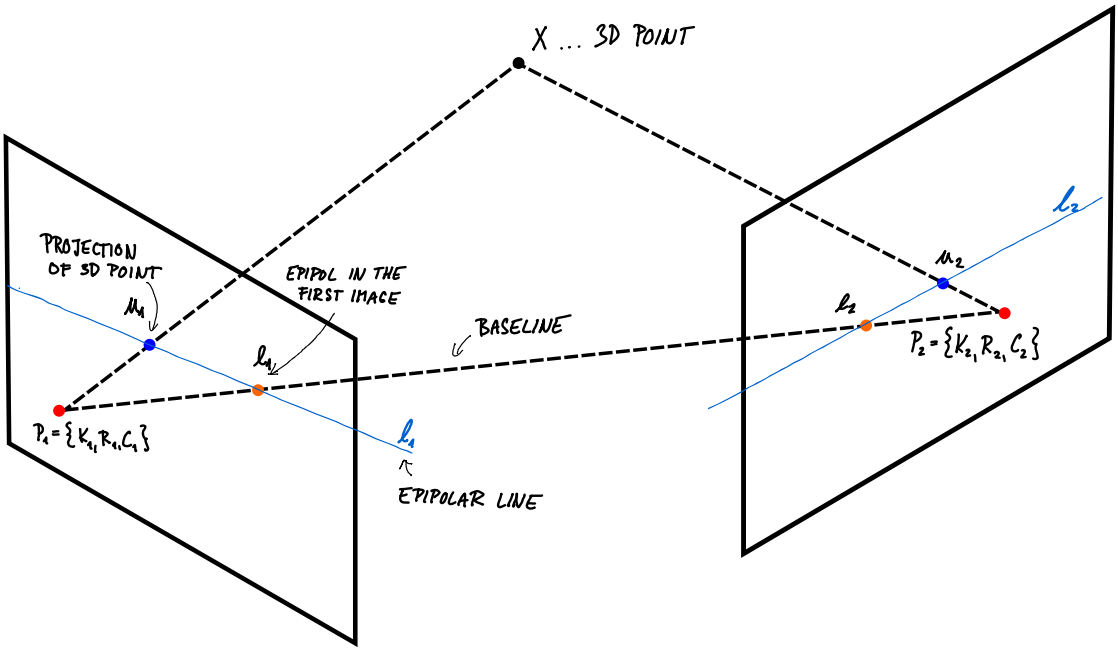


Figure 6: Visualisation of epipolar geometry for $\boldsymbol{M}_{0|0}$, i.e., $\boldsymbol{\theta}_{rd} = \emptyset$.

Having the keypoint correspondences between two planes in a 3D scene, the fundamental matrix degenerate into *homography matrix* noted by $\mathtt{H} \in \mathbb{R}^{3\times3}$. The homography matrix has full rank and map keypoints $\boldsymbol{u}_{1,m}$ in the first image at keypoints $\boldsymbol{u}_{2,m}$ in the second one up to a factor γ, i.e.,

$$\gamma \begin{bmatrix} \boldsymbol{u}_{2,m}^\top & 1 \end{bmatrix} = \mathtt{H} \begin{bmatrix} \boldsymbol{u}_{1,m} \\ 1 \end{bmatrix}. \tag{39}$$

The previous equations hold for the pinhole camera model without radial distortion. However, radial distortion can be introduced through various constraints, such as the assumption of the division model with one parameter for each image in an image pair. This relative pose model is described in Kukelova [35], and its constraints can be written in the form of an implicit equation

$$\boldsymbol{u}_{2,m}^\top(\lambda_{\mathrm{rd},2})\, \mathtt{F}\, \boldsymbol{u}_{1,m}(\lambda_{\mathrm{rd},1}) = 0 \tag{40}$$

where keypoint $\boldsymbol{u}_{l,m}(\lambda_{\mathrm{rd},l}) = [\boldsymbol{u}_{l,m}, (1 + \lambda_{\mathrm{rd},l}||\boldsymbol{u}_{l,m}||_2^2)]^\top$. In the last two decades, several solvers have been developed that calculate the relative pose of two views from keypoint correspondences under different constraints, such as assuming a different number of radial distortion parameters. Furthermore, a currently active research topic discussed in **[17]**, is the use of *affine correspondences* (AC) instead of *point correspondences* (PC). An affine correspondence, realized by triplet

$(\boldsymbol{u}_i, \boldsymbol{u}'_i, \mathtt{A}_i)$, assumes that the neighborhood of $\boldsymbol{u}_i$ is mapped by an affine transformation $\mathtt{A}_i \in \mathbb{R}^{2\times 2}$ into the second image in the neighborhood of keypoint $\boldsymbol{u}'_i$. The main advantage of solvers using affine correspondences is that they introduce more constraints, and fewer of them are required to estimate the model parameters. For example, only two matches instead of four are required to estimate a homography matrix, and only three correspondences instead of seven are required to estimate the fundamental matrix [33, 128]. This increases the probability of drawing an all-inlier sample, thereby decreasing the required number of iterations of RANSAC [129].

5.7 Bundle Adjustment

Bundle adjustment, introduced by Triggs [46], is a key method in computer vision. It involves non-linear optimization of the "bundles" of rays that intersect camera centers $\hat{\boldsymbol{C}}_l$, points in images $\hat{\boldsymbol{u}}_{l,m}$, and related 3D points $\hat{\boldsymbol{X}}_m$ where $(l, m) \in \mathcal{S}$. Bundle adjustment is necessary for propagating uncertainty from measurements to the reconstruction. The uncertainty of the keypoints is propagated using a linearized version of the non-linear projection function. Thus, if the estimated reconstruction is far from the optimal reconstruction, the uncertainty propagation leads to poor and inaccurate results.

This optimization is generally used as the final step of SfM and improves the accuracy of calculated camera poses and positions of 3D points. It allows the use of the covariance matrices of keypoints in images, employs a robust loss function $\mathcal{L}$ to avoid degeneracy, and provides the Maximum Likelihood Estimate (MLE) of the reconstruction.

Given a single camera model $\boldsymbol{M}_i$, the optimization problem seeks to minimize the loss function $\mathcal{L}(\hat{\epsilon}^{(i)}_{l,m}) = \mathcal{L}(||\boldsymbol{u}_{l,m} - \hat{\boldsymbol{u}}^{(i)}_{l,m}||_2)$ for the correspondences $(l, m) \in \mathcal{S}$. For brevity, we will drop the model index in the following paragraphs. Without loss of generality, we can write the minimization task of all correspondences as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{P}_l, \boldsymbol{X}_m, \boldsymbol{\theta}_{rd}}{\arg\min} \left(\sum_{\forall (l,m) \in \mathcal{S}} \mathcal{L}(||\boldsymbol{p}(\hat{\boldsymbol{P}}_l, \hat{\boldsymbol{X}}_m, \hat{\boldsymbol{\theta}}_{rd}) - \boldsymbol{u}_{l,m}||_2) \right). \tag{41}$$

One common optimization method used to solve this equation is the Levenberg-Marquardt algorithm [33]. Since the projection function is non-linear, we need to linearize it using the first-order Taylor series expansion, which results in the following equation

$$\boldsymbol{p}(\boldsymbol{\theta}) = \hat{\boldsymbol{u}} + \mathtt{J}\Delta\boldsymbol{\theta} \tag{42}$$

where $\mathtt{J}$ is the derivative of $\boldsymbol{p}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ in linearization point $\hat{\boldsymbol{\theta}}$. Further $\Delta\boldsymbol{\theta} = (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$ denote a small change of the reconstruction. Let us assume that the loss

function $\mathcal{L}(\hat{\boldsymbol{\epsilon}})$ does the weighting by the assumed accuracy of the observations. If the covariance matrix of all keypoints Σ_{uu} is not known, the standard reconstruction software assumes $\Sigma_{uu} = \mathtt{E}_{2N}$. The sum of weighted squares of residuals, i.e., the *Mahalanobis distances* of residuals to zero vector, can be expressed as

$$\Omega(\hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\epsilon}}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1}\, \hat{\boldsymbol{\epsilon}} \tag{43}$$

with residuals

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}(\hat{\boldsymbol{\theta}}) = \boldsymbol{u} - \boldsymbol{p}(\hat{\boldsymbol{\theta}}) \tag{44}$$

and covariance matrix

$$\Sigma_{\hat{\epsilon}\hat{\epsilon}} \approx \Sigma_{uu}/(2N - K). \tag{45}$$

The variable $K = \dim(\hat{\boldsymbol{\theta}})$ represents the dimension of the 3D reconstruction. In the case of a local or global optimum, the partial derivative of any function is equal to zero. Therefore, for the maximum likelihood estimate of the reconstruction $\boldsymbol{\theta}$ in the linearization point $\hat{\boldsymbol{\theta}}$ holds

$$\frac{\partial \Omega(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\top} = 2 \left(\frac{\partial \boldsymbol{\epsilon}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\top} \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1}\, \boldsymbol{\epsilon}(\boldsymbol{\theta}) \right) = \mathtt{J}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1}\, \boldsymbol{\epsilon}(\boldsymbol{\theta}) = \mathbf{0}\,. \tag{46}$$

This, in combination with eq. (44), leads to the *normal equation* given by

$$(\mathtt{J}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1} \mathtt{J})\, \Delta\boldsymbol{\theta} = \mathtt{J}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1} (\hat{\boldsymbol{u}} - \boldsymbol{u})\,. \tag{47}$$

The Levenberg-Marquardt algorithm adds a damping term γ to control the update step, and solves the resulting equation

$$\Delta\boldsymbol{\theta} = (\mathtt{J}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1} \mathtt{J} + \gamma I)^{-1} \mathtt{J}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1} (\hat{\boldsymbol{u}} - \boldsymbol{u}). \tag{48}$$

iteratively to approach the optimum solution. This improves the stability of the algorithm and prevents the optimization from getting stuck in a local minimum. The addition of a damping term γ helps to control the size of the update step, ensuring the convergence of the algorithm towards the global optimum. A key benefit of this formulation is that the damping term makes the matrix $\mathtt{J}^\top \Sigma_{uu}^{-1} \mathtt{J} + \gamma \mathtt{E}$ full rank, allowing it to be efficiently inverted. In SfM applications, where the equation $\mathtt{J}^\top \Sigma_{\hat{\epsilon}\hat{\epsilon}}^{-1} \mathtt{J}$ has seven degrees of freedom, the use of damping term enables the calculation of the inversion, avoiding the slower and less numerically stable MP inversion.

6 Uncertainty of the measurements

The reconstruction process begins with detecting unique feature points in images. Each feature point, as defined in [61], describes a single circular region in the image that can be further extended to a local affine region [130]. SfM algorithms estimate mathematical models, such as relative or absolute pose, from sets of correspondences of feature points between images. In this section, we discuss how to estimate the uncertainty of the keypoint, which refers to the coordinates of the center of the circular or affine region. To our knowledge, the uncertainty analysis of the keypoint of affine region has not been published yet. We discuss these topics in subsections 6.2 and 7.

6.1 Keypoints uncertainty using circular regions

Each keypoint $\boldsymbol{u}_{l,m} \in \mathbb{R}^{2\times 2}$ is located with a certain degree of accuracy inherent to the detection process. This accuracy depends on various factors such as the viewing angle, feature point contrast, lighting, camera sensor noise to signal ratio, etc. However, the keypoint accuracy is independent of the estimated reprojection error $\hat{\boldsymbol{\epsilon}}_{l,m}$, unless a suitable mathematical model of the physical camera and its intrinsic parameters are known in advance, as discussed in Kanatani [54].

There are several methods for modeling the uncertainty of keypoints. This section describes these models, summarizes the state-of-the-art methods for their estimation, discusses their limitations, and presents a new improved approach.

Isotropic homogeneous noise model: Most of the current reconstruction pipelines [8, 22, 57] and detectors [26, 61, 88] assume the isotropic homogeneous noise model of keypoints uncertainty, i.e., $\Sigma_{uu} = \sigma_0^2, \mathtt{E}_{2N}$. This simplest model assumes that all the keypoints have the same accuracy. Such a simplification is commonly used because, to our knowledge, there is no library available for describing the accuracy of the most common detectors. Using this noise model, the *variance factor* for all keypoints can be estimated as

$$\hat{\sigma}_0^2 = \frac{\hat{\boldsymbol{\epsilon}}^\top \hat{\boldsymbol{\epsilon}}}{R}\,. \tag{49}$$

The redundancy $R = 2N - K$ is the number of measurements minus the dimension of the 3D reconstruction. Here, $\hat{\boldsymbol{\epsilon}} = \boldsymbol{u} - \hat{\boldsymbol{u}}$ realize the reprojection errors and $\hat{\Sigma}_{uu} = \hat{\sigma}_0^2 \mathtt{E}_{2N}$ is the estimated covariance matrix, indicating that the accuracy of all keypoints is the same up to a common scale. To estimate varying accuracy for each keypoint, we need to model the *noise variance factor* $\hat{\sigma}_{u_i}^2$.

Isotropic inhomogeneous noise model can be approximated according to Förstner [60], p680. This model assumes an unknown noise variance $\hat{\sigma}^2_{u_i} := \hat{\sigma}^2_{u_{l,m}}$ (where m is the index of 3D point, and l the index of camera) and a unit matrix approximate covariance matrix for each keypoint, i.e., $\hat{\Sigma}_{u_i u_i} = \hat{\sigma}^2_{u_i} \mathtt{E}_2$. We use the notation $(.)_{l,m} := (.)_{u_{l,m}}$ for brevity. The model is based on the intuition that feature points with larger scale may have larger noise variance, i.e., the $\hat{\sigma}^2_{u_i}(s^2_{u_i})$ depends on the squared feature point scale $s^2_{u_i}$. The goal is to estimate the empirical function $\hat{\sigma}^2_{u_i}(s^2_{u_i})$.

To achieve this goal, we need cameras following a known camera model with known intrinsic and extrinsic parameters, and remove the distortion caused by inappropriate modeling of the camera. Suppose we calibrated the camera in advance using a suitable camera model $\mathbf{M_i}$. In that case, we can estimate the keypoint variance from the weighted residuals such that they follow a centered Gaussian distribution. Expressed mathematically, the noise model, called $\underline{c}$, is composed of two independent noise sources. The static scale-space independent part $\underline{a}$ and the second part $\underline{b}$, which is a multiple of the feature point scale s, i.e., $\underline{c} = \underline{a} + \underline{b} s_{u_i}$. The random variable $\underline{a} \in \mathcal{N}(0, \sigma^2_a)$ and $\underline{b} \in \mathcal{N}(0, \sigma^2_b)$ lead to the variance model $\sigma^2_c = \sigma^2_a + \sigma^2_b s^2_{u_i}$ and the variance of the $\boldsymbol{u}_i$ keypoint $\hat{\sigma}^2_{u_i} = \hat{\sigma}^2_c(s_{u_i})$. To simplify the iterative keypoint variances estimation, we define the information matrix

$$\mathtt{W}_{u_i u_i} = \Sigma^{-1}_{u_i u_i} = (\Sigma_{a_i a_i} + \Sigma_{b_i b_i})^{-1} \tag{50}$$

where

$$\Sigma_{a_i a_i} = \sigma^2_{a_i} \mathtt{E}_2; \qquad \Sigma_{b_i b_i} = \sigma^2_{b_i} s^2_{u_i} \mathtt{E}_2. \tag{51}$$

The $\sigma^2_{a_i}$, $\sigma^2_{b_i}$ realize the variance components of keypoint $\boldsymbol{u}_i$. Each component of weighted squared residuals can be expressed as

$$\omega_{u_i} = ||\boldsymbol{\omega}_{a_i}||_2 + ||\boldsymbol{\omega}_{b_i}||_2 \tag{52}$$

where

$$\boldsymbol{\omega}_{a_i} = \boldsymbol{\epsilon}_i{}^\top \mathtt{W}_{u_i u_i} \Sigma_{a_i a_i} \mathtt{W}_{u_i u_i} \boldsymbol{\epsilon}_i; \qquad \boldsymbol{\omega}_{b_i} = \boldsymbol{\epsilon}_i{}^\top \mathtt{W}_{u_i u_i} \Sigma_{b_i b_i} \mathtt{W}_{u_i u_i} \boldsymbol{\epsilon}_i. \tag{53}$$

Moreover, the expected value of each weighted squared residual component

$$\mathbb{E}(\omega_{a_i}) = \mathrm{tr}(\mathtt{W}_{u_i u_i} \Sigma_{a_i a_i} \mathtt{W}_{u_i u_i} \Sigma_{\epsilon_i \epsilon_i}) \tag{54}$$

$$\mathbb{E}(\omega_{b_i}) = \mathrm{tr}(\mathtt{W}_{u_i u_i} \Sigma_{b_i b_i} \mathtt{W}_{u_i u_i} \Sigma_{\epsilon_i \epsilon_i}). \tag{55}$$

equals one if the weighted squared residuals follow the centered Gaussian distribution. The term $\mathtt{R}_{u_i u_i} = \mathtt{W}_{u_i u_i} \Sigma_{\epsilon_i \epsilon_i}$ express the redundancy number of the point $\boldsymbol{u}_i$. We can assume an approximation $r_m := r_{l,m}$, i.e., the redundancy number $r_{l,m}$ is the same for all the keypoints seeing the point $\boldsymbol{X}_m$. The redundancy number is approximated by

$$r_{l,m} \approx r_m = \frac{2N_m - 3}{2N_m} \tag{56}$$

where N_m is the number of keypoints (i.e., rays) that determine the position of the 3D point $\boldsymbol{X}_m$. This approximation is valid when a large number of inlier keypoints are present in camera $\boldsymbol{P}_l$, and the camera pose is fixed regardless of the presence of a single 3D point. In other words, we assume

$$\mathtt{R}_{u_i u_i} \approx r_m \mathtt{E}_2. \tag{57}$$

The expectations from equations (54), (55) can be used as the denominators of the weighted squared residuals leading to the estimated variance factors

$$\hat{\sigma}^2_{a_i} = \frac{\hat{\omega}_{a_i}}{\mathbb{E}(\hat{\omega}_{a_i})} \qquad \hat{\sigma}^2_{b_i} = \frac{\hat{\omega}_{b_i}}{\mathbb{E}(\hat{\omega}_{b_i})}. \tag{58}$$

Extending the notation about the k-th iteration index, the new variance components are estimated from previous or initial residuals $\boldsymbol{\epsilon}_i^{(k)}$ and variance components $(\hat{\sigma}^2_{a_i})^{(k)}$, $(\hat{\sigma}^2_{b_i})^{(k)}$, i.e.,

$$(\hat{\sigma}^2_{a_i})^{(k+1)} = \frac{(\hat{\omega}_{a_i})^{(k)}}{\mathbb{E}((\hat{\omega}_{a_i})^{(k)})} \qquad (\hat{\sigma}^2_{b_i})^{(k+1)} = \frac{(\hat{\omega}_{b_i})^{(k)}}{\mathbb{E}((\hat{\omega}_{b_i})^{(k)})}. \tag{59}$$

The covariance matrices for individual keypoints are composed as

$$\hat{\Sigma}^{(k+1)}_{u_i u_i} = \left((\hat{\sigma}^2_{a_i})^{(k+1)}(\hat{\sigma}^2_{a_i})^{(k)} \dots (\hat{\sigma}^2_{a_i})^{(1)} + (\hat{\sigma}^2_{b_i})^{(k+1)}(\hat{\sigma}^2_{b_i})^{(k)} \dots (\hat{\sigma}^2_{b_i})^{(1)} s^2_{u_i}\right) \mathtt{E}_2. \tag{60}$$

After the covariance matrix estimation, we need to run the bundle adjustment to optimize the reconstruction with new weights of the residuals. This process leads to the residuals $\boldsymbol{\epsilon}_i^{(k+1)}$. The remaining question is how to initialise the variance components $(\hat{\sigma}^2_{a_i})^{(1)}$, $(\hat{\sigma}^2_{b_i})^{(1)}$. The square root of eq. (58) with priory approximate variance of keypoint coordinates $(\hat{\sigma}^2_{l,m})^{(0)} := 1$ allow us to estimate the initial normalized reprojection errors

$$\hat{\boldsymbol{y}}^{(0)}_{u_i} = \hat{\boldsymbol{y}}^{(0)}_{l,m} = \frac{\hat{\boldsymbol{\epsilon}}^{(0)}_{l,m}}{(\hat{\sigma}^2_{l,m})^{(0)}\sqrt{r_{l,m}}} \approx \frac{\hat{\boldsymbol{\epsilon}}^{(0)}_{l,m}}{\sqrt{r_m}}. \tag{61}$$

To robustly estimate the initial variance components of function $\sigma^2_{u_i}(s^2_{u_i})$, each normalized residual $\hat{\boldsymbol{y}}^{(0)}_{u_i}$ of scale s_{u_i} is assigned into one of 30 bins $[s_b, s_{b+1}]$. The standard deviation $\hat{\sigma}^{(0)}_{b,b+1}$ of each bin $[s_b, s_{b+1}]$ is

$$\hat{\sigma}^{(0)}_{b,b+1} = 1.4826\,\mathrm{med}(\{\mathrm{abs}(\hat{\boldsymbol{y}}^{(0)}_{u_i}) | s_{u_i} \in [s_b, s_{b+1}]\}). \tag{62}$$

The factor $1/\mathcal{N}^{-1}(0.75) = 1.4826$ is used to scale the median of normalized residuals to the standard deviation. Eq. (62) lead to $\hat{\sigma}^{(0)}_i = \hat{\sigma}^{(0)}_{b,b+1}$ for each bin approximated by $s_i = s_b + (s_{b+1} - s_b)/2$ where $i \in \{1, \dots, B\}$. In other words,

the initialization of variance components can be seen as fitting a line (expressed by $(\hat{\sigma}^2_{a_i})^{(1)}$, $(\hat{\sigma}^2_{b_i})^{(1)}$) to *the keypoints scale - the standard deviation* scatter plot. The authors of [60] did not evaluate the iterative part of variance components estimation and directly utilize the initial estimate of the Lowe (SIFT) keypoint variance

$$\hat{\sigma}^{(0)}_{u_i} = \sqrt{0.13^2 + (0.05 s_{u_i})^2}. \tag{63}$$

to avoid the bundle adjustment and re-weighting of the residuals. The proposed method can be generalized for any scale-independent detector that provides information about the scale or window size of detected feature points. However, this approach has some limitations. It is derived from a single physical camera with a fixed field-of-view, fixed color mapping, and one environment captured in the images. Additionally, the variance estimate in eq. (63) is specific to the Lowe detector with fixed parameters, such as edge and peak thresholds, and does not involve an iterative process between keypoint covariance estimation and bundle adjustment. As a result, the coefficients $(\hat{\sigma}_{u_i})^{(k)}$ may vary for different hyperparameters of the detector. Furthermore, the undistortion of images can introduce a small error, as the radial model $\boldsymbol{M}_{2|0}$ used for undistortion may not be as accurate as implicit camera calibration, as shown in Schöps [131].

The anisotropic inhomogeneous noise model can be estimated from the structure tensor [132] describing the local neighborhood of the feature point. Its inversion, i.e., $\hat{\Sigma}^a_{u_i u_i} = \mathtt{T}^{-1}_{u_i}$ is proportional to the covariance matrix of keypoint $\boldsymbol{u}_i$. The model assumes the relationship $\Sigma_{u_i u_i} = \sigma^2_{0,u_i} \Sigma^a_{u_i u_i}$ where the variance factor is $\sigma^2_{0,u_i} = 1$. The linearized template matching model, realized by $\mathtt{T}_{u_i}$, is calculated by dyadic products of the smooth image gradients $\mathtt{I}_x$, $\mathtt{I}_y$, i.e., the vectors obtained by the image convolution with a smooth differentiating filter. Note that some authors (Kanazawa [85]) call the dyadic products of the smooth image gradients the Hessian matrix. That may be misleading because most of the literature denotes the Hessian matrix as the second derivative of the intensity levels of image pixels. Moreover, the paper [85] assumes a covariance matrix estimation from the circular regions without specifying their radius. We assume, based on the Lowe [61], that the radius of the circular region is approximately $3 s_{u_i}$, i.e., the feature point window $\mathcal{N}_i$ has the dimension at least $6 s_{u_i} \times 6 s_{u_i}$. A reasonable choice of the standard deviation of the differentiation kernel (for keypoint $\boldsymbol{u}_i$, according to the Dickscheid [132]) is $t_{u_i} = s_{u_i}/3$. Note that the $1/3$ factor is an empirically set hyperparameter. The Gaussian convolution kernel $\mathtt{G}_t$ can be defined as multiplication of two 1D convolution kernels

$$\mathtt{G}_t(x, y, t) = \left(\frac{1}{2\pi t} \exp^{-\frac{x^2}{2t^2}} \right) \left(\frac{1}{2\pi t} \exp^{-\frac{y^2}{2t^2}} \right) \tag{64}$$

allowing the derivation of 1D differential Gaussian kernels $\mathtt{G}_{x,t}$,$\mathtt{G}_{y,t}$ and calculation of smooth image gradients as the convolution

$$\mathtt{I}_{x,t}(x,y,t) = \mathtt{G}_{x,t}(x,y,t) * \mathtt{I}(x,y) \quad \text{where} \quad \mathtt{G}_{x,t} = \frac{\partial \mathtt{G}_t}{\partial x} = -\frac{x}{t^2}\mathtt{G}_t \tag{65}$$

$$\mathtt{I}_{y,t}(x,y,t) = \mathtt{G}_{y,t}(x,y,t) * \mathtt{I}(x,y) \quad \text{where} \quad \mathtt{G}_{y,t} = \frac{\partial \mathtt{G}_t}{\partial y} = -\frac{y}{t^2}\mathtt{G}_t. \tag{66}$$

Note that we suppressed the operator parameters, e.g., $\mathtt{G}_t := \mathtt{G}_t(x,y,t)$, $\mathtt{G}_{x,t} := \mathtt{G}_{x,t}(x,y,t)$, for brevity. The dyadic product of $\mathtt{I}_{x,t}$, $\mathtt{I}_{y,t}$ functions (can be seen as Hadamard product of the smooth gradient images) is the structure tensor

$$\mathtt{T} = \mathtt{G}_s * \begin{bmatrix} I_{x,t}I_{x,t} & I_{x,t}I_{y,t} \\ I_{x,t}I_{y,t} & I_{y,t}I_{y,t} \end{bmatrix}. \tag{67}$$

The definition in eq. (67) assumes as the output 4D tensor of values, i.e. three weighted squared smooth gradient images for each pixel. In practise, we have t_{u_i}, s_{u_i} (instead of t,s) related to $\boldsymbol{u}_i$ and the feature point region. The convolution with $\mathtt{G}_s(u_{i,1}, u_{i,2}, s_{u_i})$ for a single point $\boldsymbol{u}_i$ can be seen as the weighting (by Gaussian weights) of squared image derivatives. Therefore, $\mathtt{I}_x$, $\mathtt{I}_y$ evaluated far from $\boldsymbol{u}_i$ have neglectable contribution to $T_{u_i} \in \mathbb{R}^{2\times 2}$. We assume the window size $6s_{u_i}$ and weight squared smooth image gradients (in this window) by $\mathtt{G}_s$ weights based on their distance to $\boldsymbol{u}_i$. The σ_n realizes the standard deviation of the image noise, and $N_{\mathcal{N}_i}$ is the number of pixel intensities used in weighting by $\mathtt{G}_s$.The covariance matrix of a single feature point that describes a circular region in the image is (according Dickscheid [132]) calculated as the inversion of related structure tensor, i.e.,

$$\hat{\Sigma}^0_{u_i u_i} = \frac{\sigma_n}{N_{\mathcal{N}_i}} \mathtt{T}^{-1}_{u_i} \tag{68}$$

The anisotropic inhomogeneous noise model estimation has the benefit that the inversion in Equation eq. (68) can be calculated without prior knowledge of the physical camera and reconstruction. Only the images, feature points, differentiation and integration scales (s_{u_i} and t_{u_i}), and the standard deviation of the image noise (σ_n) are required as inputs. On the other hand, the isotropic inhomogeneous noise model requires camera calibration, image undistortion, and camera extrinsics prior to the iterative update of the $\sigma^2_{a_i}$ and $\sigma^2_{b_i}$ parameters of the empirical Equation eq. (63). Furthermore, there is no research that evaluates how well the empirical function $\sigma_{u_i}(s_{u_i})$ fits the noise variance factor for different camera devices, environments, hyperparameters of the SIFT detector, and other common detectors.

Our extension combines and generalizes the methods described above. We estimate the anisotropic inhomogeneous noise and simultaneously estimate the noise variance by normalizing the expectation of the weighted squared residuals. First, we estimate the covariance matrix $\hat{\Sigma}^0_{u_i u_i}$ as the weighted inversion of the structure tensor according to eq. (68). Next, we follow the Eqs. 50 - 59, using the redefined eq. (51) as

$$\Sigma_{a_i a_i} = \sigma^2_{a_i} \hat{\Sigma}^0_{u_i u_i}; \qquad \Sigma_{b_i b_i} = \sigma^2_{b_i} s^2_{u_i} \hat{\Sigma}^0_{u_i u_i}. \tag{69}$$

The variance components $(\hat{\sigma}^2_{a_i})^{(1)}$, $(\hat{\sigma}^2_{b_i})^{(1)}$ are estimated according equations (61), (62) assuming the weighting of each coordinate of $\hat{\boldsymbol{\epsilon}}^{(0)}_{l,m} \in \mathbb{R}^2$ by diagonal values of $\hat{\Sigma}^0_{u_i u_i} \in \mathbb{R}^{2\times 2}$, i.e.

$$\left(\hat{y}^{(0)}_{l,m}\right)_j \approx \frac{\left(\hat{\epsilon}^{(0)}_{l,m}\right)_j}{\left(\hat{\Sigma}^0_{u_i u_i}\right)_{jj} \sqrt{r_m}} \qquad \text{for} \quad j = \{1, 2\}. \tag{70}$$

This approach allows us to compensate the image noise σ_n distortion by applying various differentiation $\mathtt{G}_t$ and smoothing $\mathtt{G}_s$ kernels for individual keypoints.

6.2 Keypoints uncertainty using affine regions

State-of-the-art SfM pipelines such as COLMAP use affine region detectors for faster and more robust model estimation algorithms. However, to our knowledge, there is no approach for estimating the uncertainty of keypoints using related affine regions. Non-maxima suppression of the feature point scale is not straightforward in the 4D space of affine region parameters, which includes rotation, scale, and two shears. We assume that the keypoint $\boldsymbol{u}_i$ denotes the center of the affine region, and the matrix $\mathtt{A}_{u_i} \in \mathbb{R}^{2\times 2}$ transforms the unit circle to the ellipse bounding the region area, as shown in an example in Fig. 8. We propose to approximate the scale of the affine region as one-third of the mean region radius, which is the geometric mean of the ellipse semi-axes

$$s_{u_i} = \frac{\sqrt{\left(\lambda_{u_i,1} + \lambda_{u_i,2}\right)/2}}{3} \qquad \text{where} \quad [\lambda_{u_i,1}, \lambda_{u_i,2}] = \text{eig}(\mathtt{A}_{u_i} \mathtt{A}^T_{u_i}). \tag{71}$$

The structure tensor for a single affine region assumes a differentiation kernel with the same standard deviation $t_{u_i} = s_{u_i}/3$ and a convolution with a multivariate Gaussian $\mathtt{G_S}$ that follows the shape of the affine region, i.e.

$$\mathtt{T_A} = \mathtt{G_S} * \begin{bmatrix} I_{x,t} I_{x,t} & I_{x,t} I_{y,t} \\ I_{x,t} I_{y,t} & I_{y,t} I_{y,t} \end{bmatrix}. \tag{72}$$

The symmetric semi-definite kernel applied in the $\mathtt{G_S}$ operator follows the shape of the affine region, and we define it as

$$\mathtt{G_S}(\mathtt{A}_{u_i}) = \mathcal{N}\left(\mathbf{0}, \frac{1}{9}\ \mathtt{A}_{u_i}\mathtt{A}_{u_i}^T\right). \tag{73}$$

The unbiased estimate of keypoint covariance matrix is therefore redefined as

$$\hat{\Sigma}^0_{u_iu_i} = \frac{\sigma_n}{N_{\mathcal{N}_i}}\mathtt{T}_{\mathtt{A}}^{-1}(\boldsymbol{u}_i, \mathtt{A}_{u_i}) \tag{74}$$

where $N_{\mathcal{N}_i}$ is the number of pixels inside the affine region $\mathtt{A}_{u_i}$. This approach is particularly beneficial for elongated affine regions. In comparison with circular regions, we assume the image intensities inside the affine region and remove the influence of surrounding structures that are not employed in keypoint position estimation.

Extension: We can extend this approach to estimate the anisotropic inhomogeneous noise from affine regions and simultaneously estimate the noise variance by normalizing the expectation of the weighted squared residuals. The matrix $\hat{\Sigma}^0_{u_iu_i}$ estimated by eq. (74) can be employed in eq. (69) to approximate the variance components. Next, we follow equations (50) - (59) except of updated eq. (51). The variance components $(\hat{\sigma}^2_{a_i})^{(1)}$, $(\hat{\sigma}^2_{b_i})^{(1)}$ are estimated according equations (61), (62) assuming the weighting of each coordinate of $\hat{\boldsymbol{\epsilon}}^{(0)}_{l,m} \in \mathbb{R}^2$ by diagonal values of $\hat{\Sigma}^0_{u_iu_i} \in \mathbb{R}^{2\times 2}$ as in eq. (70). This approach for affine regions removes the influence of surrounding pixel intensities that are not used in keypoint position estimation and compensates for image noise σ_n distortion by applying various differentiation $\mathtt{G}_t$ and smoothing $\mathtt{G}_s$ kernels for individual keypoints.

6.3 Evaluation

In this evaluation section, we compare the proposed feature point noise estimators. Since the developed approaches have not yet been published, we present only a visual comparison highlighting the differences between the individual methods rather than a more in-depth statistical verification, which will be the subject of a future work. As our focus is on positional noise estimation, which is sensitive to distortions, we eliminate any other sources of deviations from the assumed mathematical model, such as radial distortion. To minimize the reprojection errors caused by radial distortion, it is necessary to choose a suitable camera model $\boldsymbol{M}_i$ and calculate the intrinsic camera parameters, including $\mathtt{K}^{(i)}$ and $\boldsymbol{\theta}^{(i)}_{rd}$, in advance. The authors of Förstner [60] used a polynomial distortion model with two parameters $\boldsymbol{M}_{2|0}$ to calibrate the camera and rectify captured images to a pinhole camera

Figure 7: The vizualization of the sparse reconstruction and images captured by Samsung S10e camera device for the *Library* dataset. This dataset consist of 8 cameras and 4137 points triangulated out of 15103 utilised observations.

model $\boldsymbol{M}_{0|0}$. In our experiments, we used the same camera models and Newton's method with numerical differentiation using central differences to undistort images, i.e., utilizing the COLMAP [8] undistortion module. The next step is to detect and describe feature points in the undistorted images using a scale-independent detector, such as the DoG [61] or MSER [25] detector, which provides a scale for each keypoint. The feature points allow the sparse reconstruction pipeline, such as COLMAP, to calculate the 3D points and camera poses. The reconstruction $\hat{\boldsymbol{\theta}}$ is optimized with the approximate initial covariance matrix $\Sigma_{uu}^{0} = \sigma_0^2 \mathrm{E}_{2N}$ and the approximate noise variance factor $\sigma_0^2 = 1$ by bundle adjustment, as described in Agarwal [133]. This allows us to obtain $\hat{\boldsymbol{\theta}}$ as close as possible to the unknown ground truth $\boldsymbol{\theta}$. The majority of the remaining reprojection errors $\hat{\boldsymbol{\epsilon}} = \boldsymbol{u} - \hat{\boldsymbol{u}}$ can be attributed to inaccuracies of the detector. This section starts with a description of the experiments to visualize individual noise models of keypoints on a simple dataset. We show the differences between covariance matrices estimated by using the circular and affine regions.

The dataset for evaluation of different noise models of keypoint accuracy are composed of: (1) *undistorted images* that are registered in the reconstruction $\hat{\boldsymbol{\theta}}$, (2) *feature points* that are marked as inliers, i.e., points that have exactly one

Figure 8: This example show affine regions detected by the COLMAP reconstruction pipeline. We calculate the circular regions as the geometric mean of the standard ellipse semiaxes lengths, which can be obtained as the Frobenius norm $||\mathtt{A}_{u_i}||_F$ multiplied by a factor of $1/\sqrt{2}$. The image patches corresponding to these regions are used to estimate the covariance matrices of the keypoint positions.

related tuple in the index set $\mathcal{S}$, and (3) *the reconstruction* called $\hat{\boldsymbol{\theta}}$ that consist of 3D points, camera intrinsic and camera extrinsic parameters.

The composition of the dataset was as follows: we first performed *intrinsic calibration* for each physical camera by capturing a set of images of the checkerboard pattern and computing the $\boldsymbol{M}_{2|0}$ model parameters using the Computer Vision Toolbox in Matlab. Next, we applied *undistortion* to the dataset images using Newton's method with numerical differentiation and central differences, which is implemented in COLMAP. We then performed *sparse reconstruction* to find the inlier keypoint correspondences between images and calculate the camera extrinsic, using COLMAP. Finally, we optimized the sparse reconstruction with the assumption of the isotropic homogeneous noise model of the keypoints positions by Bundle Adjustment using Ceres Solver.

The individual keypoint noise models are visualized on a created small dataset comprising 8 cameras and 4137 triangulated points out of 15103 keypoints, as shown in Fig. 7. Keypoints with reprojection errors larger than two pixels, comprising around 6% of the keypoints with the largest residuals, were filtered out. The reconstruction software extracts keypoints with their associated affine regions, defined by the keypoints $\boldsymbol{u}_i$ at the center and the matrices $\mathtt{A}_{u_i} \in \mathbb{R}^{2\times 2}$ transforming

the unit circle to the ellipse that bounds the affine region, as visualized in Fig. 8.

To compare the consistency of estimated uncertainties for individual keypoints across multiple images, an image pair with significant viewpoint and positional change was manually selected, and all the estimated noise models were plotted for each tenth inlier correspondence seen in both images, as shown in Fig. 11. The covariance matrices of keypoints are visualized using five times up-scaled standard ellipses to be visible for all noise models. The mean variance factor for the isotropic homogeneous noise model is $\hat{\sigma}_0^2 = 0.92^2$, calculated according eq. (49).

For the isotropic inhomogeneous noise model, the noise variance factor was assumed for each keypoint. Equations (56), (61), and (62) were used to estimate the empirical equation $\hat{\sigma}_u^2 = 0.782^2 + (0.036\, s_u)^2$ for our sample dataset. Hence, we adopted the same approach as Förster [60] and refrained from evaluating the iterative refinement of variance components after bundle adjustment optimization with reweighting reprojection errors in order to ensure consistency of results. This experiment result in observation that the parameters empirically found in [60] may not be universally applicable to other camera devices and datasets.

The next experiments focus on the evaluation of an anisotropic inhomogeneous noise model. Fig. 11c show the covariance matrices calculated according to eq. (68), i.e., without the correction of the expectation of squared weighted residuals. This estimate and the one in Fig. 11d assumed circular regions and isotropic kernel $\mathtt{G}_s$ weighting individual contributions of the squared smooth image gradients to the structure tensor. Nevertheless, the second approach in Fig. 11d estimate robustly the initial variance components of function $\sigma_{u_i}^2(s_{u_i}^2)$ by normalizing the reprojection errors according eq. (61). We assigned the normalized errors $\hat{\boldsymbol{y}}_{u_i}^{(0)}$ to thirty bins according to their related feature point scales and estimated standard deviation for each bin using eq. (62). It results in thirty pairs of mean scale (of each bin) and related standard deviation (of the keypoint). The function $\sigma_{u_i}^2(s_{u_i}^2)$ was realized by fitting the fifth-degree polynomial to these pairs. We use the estimated function to plot the standard ellipses in Fig. 11d. A similar approach, using affine regions and anisotropic kernel weighing the contributions to structure tensor, is visualized in Fig. 11e. Detailed visualization of covariance matrices for different noise models and kernels weighing the contributions to structure tensor is in Fig. 12.

The last experiment focuses on the iterative reweighting of the squared reprojection errors according to eq. (59). In each iteration, the covariance matrices are recomputed and utilized to optimize the reconstruction in BA. This approach leads to such covariance matrices that the expectation of the weighted squared reprojection errors equals one. For this purpose, we employed the initial estimate of covariance matrices according to eq. (74). Next, we calculated the variance components, composed the covariance matrix, and utilized it in the BA. We performed twenty iterations of this refinement. However, the multiples of covariance

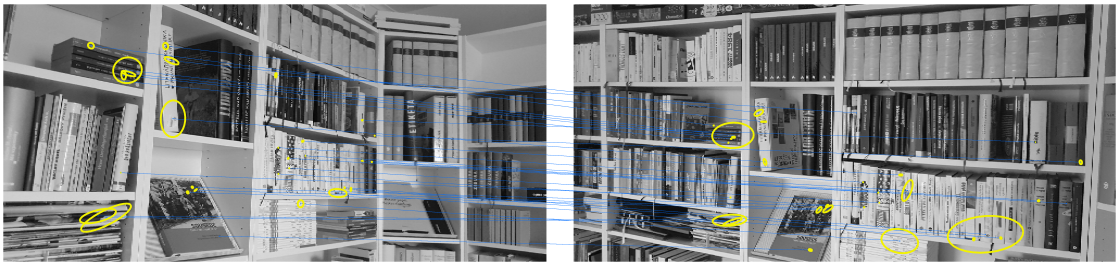


Figure 9: The visualization of estimated anisotropic inhomogeneous noise of affine regions using the iterative reweighting of the squared reprojection errors according to eq. (59).

components related to individual covariance matrices converged in five steps of this process. As we directly followed the eq. (59), each keypoint assumes its own variance components. Therefore, we were looking for $2N$ weights of N keypoints, which appeared prone to overfitting. The visualization of estimated covariance matrices, i.e., the standard ellipses, is in Fig. 9, and 10. The clustering w.r.t. similar scales, i.e., estimation of weights for sets of keypoints, may be a direction for future research.

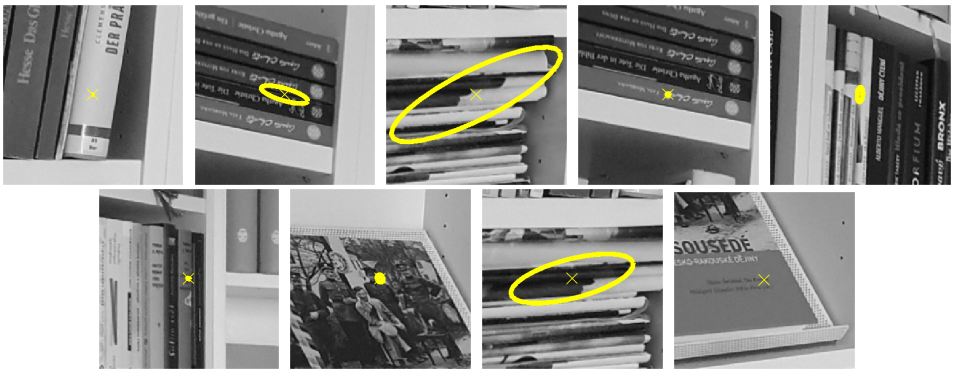
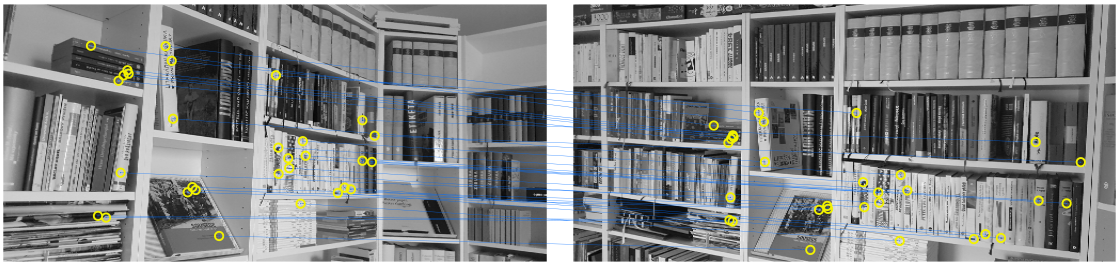
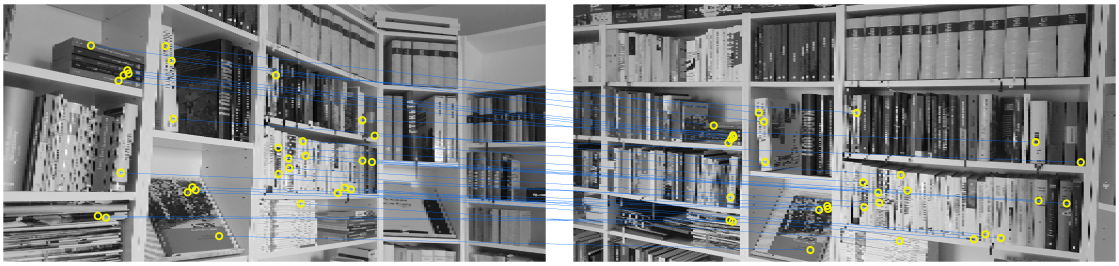


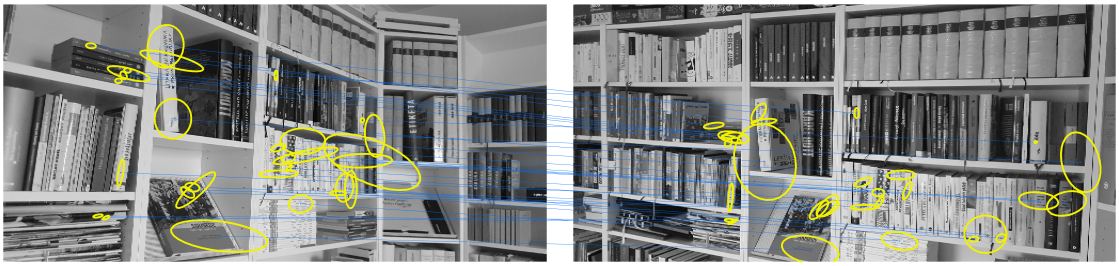
Figure 10: The detail of estimated standard ellipses of anisotropic inhomogeneous noise of affine regions using the iterative reweighting of the squared reprojection errors according to eq. (59). All the standard ellipses are five times up-scaled. This figure shows the same keypoints as Fig. 12.



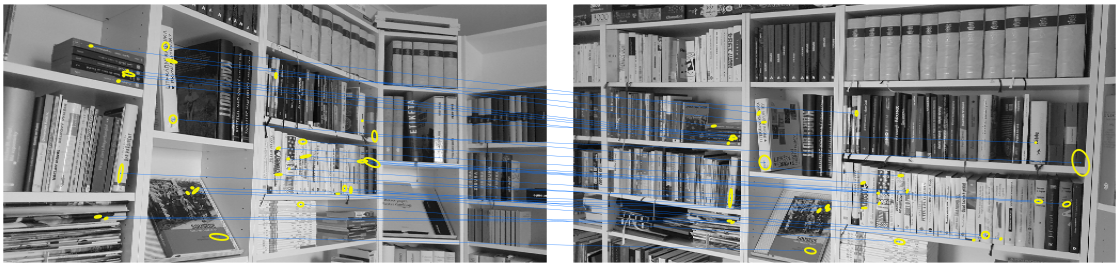
(a) Estimated isotropic homogeneous noise



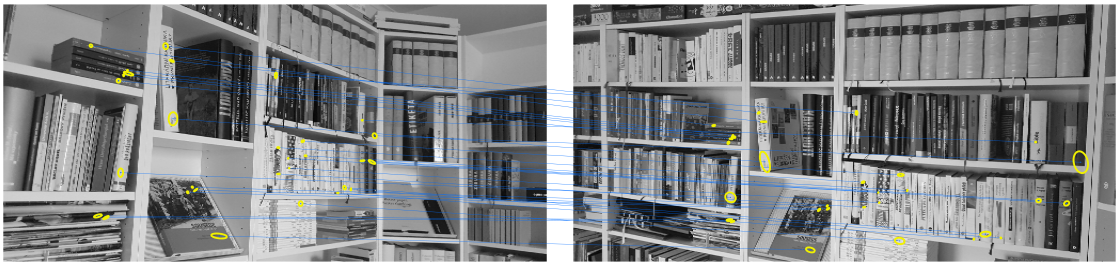
(b) Estimated isotropic inhomogeneous noise



(c) Estimated anisotropic inhomogeneous noise up to scale - circular regions



(d) Estimated anisotropic inhomogeneous noise - circular regions



(e) Estimated anisotropic inhomogeneous noise - affine regions

Figure 11: The visualization of different noise models on a pair of images from the Library dataset. All the standard ellipses are five times up-scaled. Fig. 12 shows the visualization of individual keypoint covariance matrices.

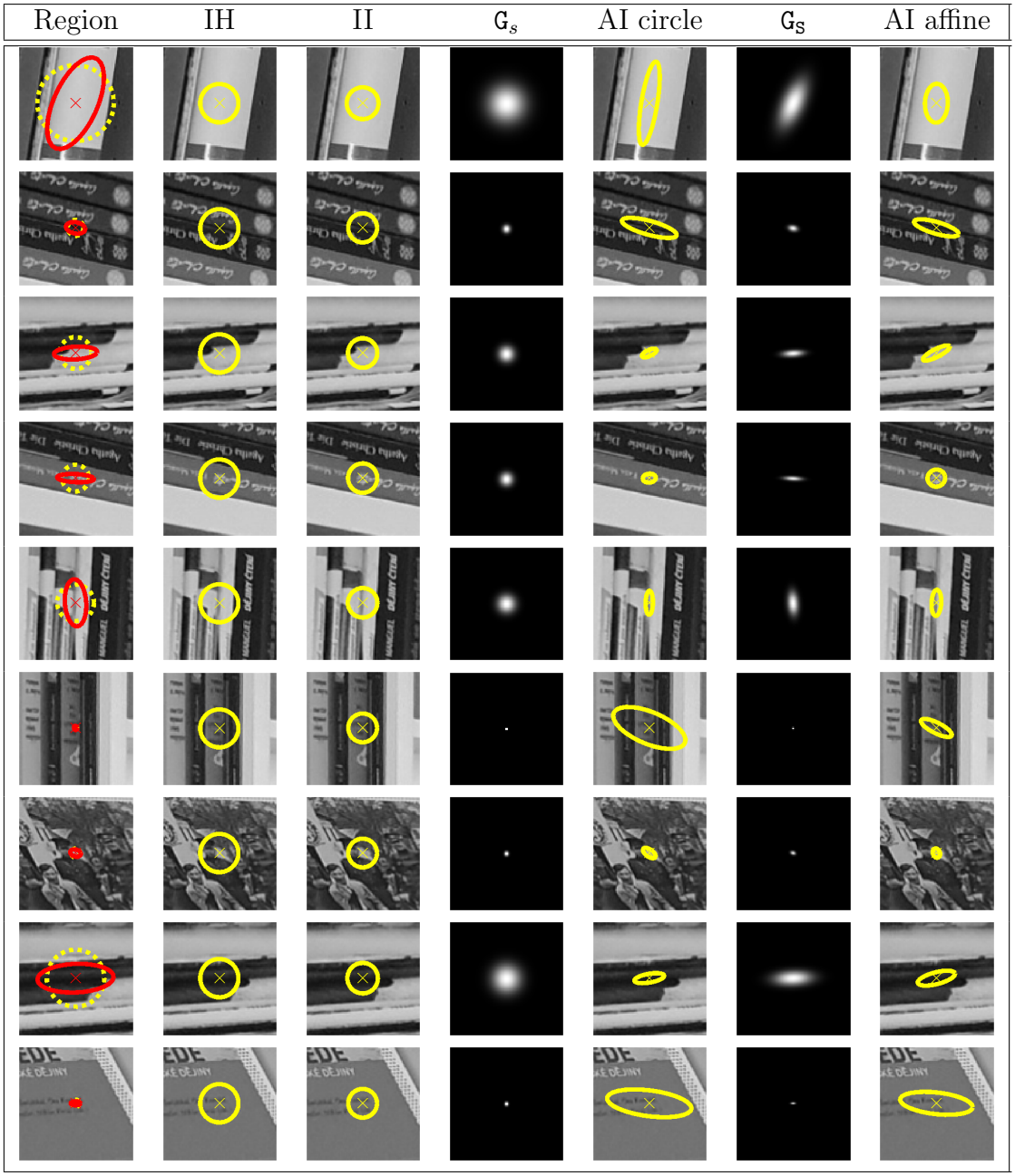


Figure 12: The visualization of keypoints and their covariance matrices for different noise models. All the standard ellipses are five times up-scaled. The first column show detected keypoint and related circular and affine regions. The abbreviation: "IH" is isotropic homogeneous noise, "II" is isotropic inhomogeneous noise, $\mathtt{G}_s$ is the isotropic Gaussian kernel, "AI circle" is anisotropic inhomogeneous noise using the weights $\mathtt{G}_s$, $\mathtt{G}_\mathtt{S}$ is the anisotropic Gaussian kernel, and "AI affine" is anisotropic inhomogeneous noise using the weights $\mathtt{G}_\mathtt{S}$.

7 Measurement transformation uncertainty

The uncertainty of keypoints can be estimated using the linearized model of template matching. However, as far as we know, no algebraic equation for propagation of uncertainty to the feature point scale, orientation, or sheer has not been published yet. One possible way is to follow the approach derived for keypoints, i.e., express the uncertainty propagation from the linearized model of the template matching. However, this is challenging because of the complicated relationship between scale space, feature point orientation, and an image template. Fortunately, we can propagate the uncertainty from the opposite direction, from the transformations of measurements. For example, we can estimate the positional residuals of a keypoint by transforming its location in the first image to the second image using the ground truth transformation and then measuring the residual with respect to the corresponding keypoint in the second image. By collecting enough residuals, we can estimate the standard deviation of the positional transformation. Likewise, we can use this approach to calculate the standard deviations of the scale and orientation transformations. This approach involves a large set of ground truth transformations and related measurements. While the propagation from measurement transformations to measurements is straightforward (mentioned at the end of this section), we mainly focus on the steps to determine the bias and variance of angular, scale, and positional transformations of detected correspondences.

Starting from a large dataset of homographies, the normalized homography implied by the l-th plane and transformation from camera $\mathtt{P}_j$ to $\mathtt{P}_k$ is expressed as $\mathtt{H}_{kj} = \mathtt{R}_{kj} - (\boldsymbol{t}_{kj}\boldsymbol{n}_l^\top)/d_l$, where $\boldsymbol{n}_l \in \mathbb{R}^3$ is the plane normal, and d_l is its intercept. Correspondences are denoted as $(\boldsymbol{u}_i, \phi_i, s_i, \boldsymbol{u}'_i, \phi'_i, s'_i)$ where $\phi_i \in [0, 2\pi)$ is the SIFT feature orientation, $s_i \in \mathbb{R}$ is the scale and the prime symbol denotes measurements in the second image. A reference similarity transformation (4 DoF) $\widetilde{\mathtt{A}}_i$ is estimated in the vicinity of the keypoint pair derived from $\widetilde{\mathtt{H}}_{kj}$ as in Barath [121]. Therefore, we have a unique affinty transformation of each correspondence of feature points. While the reference translations $\widetilde{\boldsymbol{t}}_{u_i}$ and scale ratios $\widetilde{r}_{u_i}$can be easily determined from $\widetilde{\mathtt{H}}_{kj}$ and $\widetilde{\mathtt{A}}_i$, the determination of the reference rotations $\widetilde{\alpha}_{u_i}$ requires care. There are two approaches to obtain reference rotations: (1) comparing direction vectors $\mathtt{d}(\phi'_i) = [\cos(\phi'_i)\ \sin(\phi'_i)]^\top$ in the second image with the transformed direction $\mathtt{d}(\phi_i) = [\cos(\phi_i)\ \sin(\phi_i)]^\top$ in the first image using a local reference affinity $\widetilde{\mathtt{A}}_i$, and (2) deriving a local rotation from the reference $\widetilde{\mathtt{H}}_{kj}$. The steps to extract the reference transformations are as follows:

1. approximate the projective transformation $\widetilde{\mathtt{H}}_{kj}$ by a local affinity $\widetilde{\mathtt{A}}_i \in \mathbb{R}^{2\times 2}$

2. decompose $\widetilde{\mathtt{A}}_i$ into scale ratio $\widetilde{r}_{u_i}$, rotation angle $\widetilde{\alpha}_{u_i}$, and two shears $\widetilde{\boldsymbol{p}}_{u_i} \in \mathbb{R}^2$

3. estimate the rotation angle between directional vectors $\mathrm{d}(\phi_i')$, $\widetilde{\mathtt{A}}_i\mathrm{d}(\phi_i)$

As shown, there are several approaches to derive the reference angular transformation $\widetilde{\alpha}_{u_i}$. We explored calculating it as the angle between the directional vectors $\angle(\mathtt{d}(\phi_i'), \widetilde{\mathtt{A}}_i\mathtt{d}(\phi_i))$, as well as three possible decompositions of $\widetilde{\mathtt{A}}_i$: QR, SVD, and exponential decomposition (additive decomposition of exponent $\widetilde{\mathtt{B}}_i$ of $\widetilde{\mathtt{A}}_i = \exp(\widetilde{\mathtt{B}}_i)$). Each of these methods leads to different reference rotations because $\widetilde{\alpha}_{u_i}$ is affected by the shears $\widetilde{\boldsymbol{p}}_{u_i}$ in $\widetilde{\mathtt{A}}_i$. When the shears are small, all of the methods yield similar results. The magnitude $||\widetilde{\boldsymbol{p}}_{u_i}||^2$ of the shears can be approximated by the condition number $\mathrm{cond}(\widetilde{\mathtt{A}}_i)$. In our evaluation, we assume only $\widetilde{\mathtt{A}}_i$ transformations with a condition number smaller than a chosen threshold. We then compare the reference transformations with the measured ones to estimate their uncertainty.

7.1 The positional transformation

The symmetric positional residual of each keypoint pair depends on the mean reprojection error

$$\epsilon_{u_i} = \sqrt{(|\boldsymbol{u}_i' - \widetilde{\mathtt{H}}_{kj}(\boldsymbol{u}_i)|_2^2 + |\boldsymbol{u}_i - \widetilde{\mathtt{H}}_{kj}^{-1}(\boldsymbol{u}_i')|_2^2)/8}. \tag{75}$$

Dividing the root mean squared error (RMSE) by $\sqrt{8}$ leads to a conservative estimate of ϵ_{u_i} and related standard deviation σ_i of all coordinates of the keypoints $\boldsymbol{u}_i = \begin{bmatrix} u_{i1} & u_{i2} \end{bmatrix}^\top$ and $\boldsymbol{u}_i' = \begin{bmatrix} u_{i1}' & u_{i2}' \end{bmatrix}^\top$. Hence, we assume $\mathbb{E}(\underline{\boldsymbol{u}}_i\underline{\boldsymbol{u}}_i^\mathsf{T}) = \mathbb{E}(\underline{\boldsymbol{u}}_i'(\underline{\boldsymbol{u}}_i')^\top) = \sigma_i^2\mathtt{I}_2$, which also holds for the errors $\underline{\boldsymbol{e}}_i = \underline{\boldsymbol{u}}_i - \mathbb{E}(\underline{\boldsymbol{u}}_i)$ and $\underline{\boldsymbol{e}}_i' = \underline{\boldsymbol{u}}_i' - \mathbb{E}(\underline{\boldsymbol{u}}_i')$. Linearizing $\boldsymbol{u}_i - \mathrm{h2a}(\mathtt{H}_{kj}\,\mathrm{a2h}(\boldsymbol{u}_i'))$ leads to $\boldsymbol{e}_i - \mathtt{A}_i(\boldsymbol{e}_i')$, and similarly for the second term. Thus, the RMSE, i.e., the expression under the squareroot in (75) is linearized to

$$\Omega_i = ||\boldsymbol{e}_i - \mathtt{A}_i(\boldsymbol{e}_i')||_2^2 + ||\boldsymbol{e}_i' - \mathtt{A}_i^{-1}(\boldsymbol{e}_i)||_2^2 \tag{76}$$

Now, we calculate the expectation of $\mathbb{E}(\underline{\Omega}_i)$ and obtain

$$\begin{aligned}
\mathbb{E}(\underline{\Omega}_i) &= \mathbb{E}\left((\boldsymbol{e}_i - \mathtt{A}_i(\boldsymbol{e}_i'))^\top(\boldsymbol{e}_i - \mathtt{A}_i(\boldsymbol{e}_i')) + (\boldsymbol{e}_i' - \mathtt{A}_i^{-1}(\boldsymbol{e}_i))^\top(\boldsymbol{e}_i' - \mathtt{A}_i^{-1}(\boldsymbol{e}_i))\right) && (77)\\
&= \mathbb{E}\left(\boldsymbol{e}_i^\top\boldsymbol{e}_i + (\boldsymbol{e}_i')^\top\mathtt{A}_i^\top\mathtt{A}_i\boldsymbol{e}_i' + (\boldsymbol{e}_i')^\top\boldsymbol{e}_i' + \boldsymbol{e}_i^\top\mathtt{A}_i^{-\top}\mathtt{A}_i^{-1}\boldsymbol{e}_i\right) && (78)
\end{aligned}$$

With $\mathrm{tr}(\mathtt{RS}) = \mathrm{tr}(\mathtt{SR})$, thus $\mathtt{a}^\mathsf{T}\mathtt{S}\mathtt{a} = \mathrm{tr}(\mathtt{a}^\mathsf{T}\mathtt{S}\mathtt{a}) = \mathrm{tr}(\mathtt{S}\mathtt{a}\mathtt{a}^\mathsf{T})$ and therefore

$$\begin{aligned}
\mathbb{E}(\underline{\Omega}_i) &= \mathbb{E}\left(\boldsymbol{e}_i^\top\boldsymbol{e}_i + \boldsymbol{e}_i'^\top\mathtt{A}_i^\top\mathtt{A}_i\boldsymbol{e}_i' + \boldsymbol{e}_i'^\top\boldsymbol{e}_i' + \boldsymbol{e}_i^\top\mathtt{A}_i^{-T}\mathtt{A}_i^{-1}\boldsymbol{e}_i\right) && (79)\\
&= \mathbb{E}\left(\mathrm{tr}(\boldsymbol{e}_i\boldsymbol{e}_i^\top) + \mathrm{tr}(\mathtt{A}_i^\top\mathtt{A}_i\boldsymbol{e}_i'\boldsymbol{e}_i'^\top) + \mathrm{tr}(\boldsymbol{e}_i'\boldsymbol{e}_i'^\top) + \mathrm{tr}(\mathtt{A}_i^{-T}\mathtt{A}_i^{-1}\boldsymbol{e}_i\boldsymbol{e}_i^\top)\right) && (80)\\
&= \mathrm{tr}(\mathbb{E}(\boldsymbol{e}_i\boldsymbol{e}_i^\top)) + \mathrm{tr}(\mathtt{A}_i^\top\mathtt{A}_i\mathbb{E}(\boldsymbol{e}_i'\boldsymbol{e}_i'^\top)) + \mathrm{tr}(\mathbb{E}(\boldsymbol{e}_i'\boldsymbol{e}_i'^\top)) + \mathrm{tr}(\mathtt{A}_i^{-T}\mathtt{A}_i^{-1}\mathbb{E}(\boldsymbol{e}_i\boldsymbol{e}_i^\top)) && (81)\\
&= \mathrm{tr}(\mathtt{I}_2)\sigma_i^2 + \mathrm{tr}(\mathtt{A}_i^\top\mathtt{A}_i)\sigma_i^2 + \mathrm{tr}(\mathtt{I}_2)\sigma_i^2 + \mathrm{tr}(\mathtt{A}_i^{-\mathsf{T}}\mathtt{A}_i^{-1})\sigma_i^2 && (82)\\
&= (4 + \mathrm{tr}(\mathtt{A}_i^\top\mathtt{A}_i) + \mathrm{tr}(\mathtt{A}_i^{-\mathsf{T}}\mathtt{A}_i^{-1}))\sigma_i^2\,. && (83)
\end{aligned}$$

If we focus on the eigenvalues $\lambda_{1,2}(\mathtt{A}_i^\mathsf{T}\mathtt{A}_i)$, the equation can be rewritten to

$$\mathbb{E}(\underline{\Omega}_i) = (4 + \lambda_1 + \lambda_2 + 1/\lambda_1 + 1/\lambda_2)\sigma_i^2 \geq 8\sigma_i^2 \tag{84}$$

since $x + 1/x = (1-x)^2/x + 2 \geq 2$ for $x > 0$. Hence, if $\lambda_1 = \lambda_2 = 1$, thus for a pure rotation, the value $\epsilon_{u_i}^2$ is an unbiased estimator for σ_i^2.

7.2 The scale transformation

The scale transformation is evaluated using the ratios $r_{u_i} = s_i'/s_i$ of a keypoint pair. Its ratio $\Delta r_{u_i} = r_{u_i}/\widetilde{r}_{u_i}$ to the reference ratio

$$\widetilde{r}_{u_i} = \sqrt{|\widetilde{\mathtt{A}}_i|} \tag{85}$$

should lead to $\mathbb{E}(\Delta r_{u_i}) = 1$. Further, we use a weighted log-ratio, measured as $\rho_{u_i} = \log(\Delta r_{u_i})/\widetilde{r}_{u_i}$ which should follow $\mathbb{E}(\rho_{u_i}) = 0$, and takes into account the intuition, that larger scales are less accurate.

7.3 The angular transformation

The angular transformation derived from the measurements equals $\alpha_{u_i} = \phi_i' - \phi_i$. Having the reference angular transformation $\widetilde{\alpha}_i$, its difference to the measured one is given by $\Delta\alpha_i = \widetilde{\alpha}_i - \alpha_i$.

Comparing direction vectors. The directional vector $\boldsymbol{d}_{u_i} = [\cos(\phi_i)\ \sin(\phi_i)]^\top$, which realize the orientation of the first keypoint, can be transformed into the second image by multiplying it with the local approximation of the affinity transformation

$$\widetilde{\boldsymbol{d}'}_{u_i} = \widetilde{\mathtt{A}}_i \boldsymbol{d}_{u_i}. \tag{86}$$

The matrix $\widetilde{\mathtt{A}}_i \in \mathbb{R}^{2\times 2}$ does not include the projective part, and enables us to capture the similarity transformation assumed by the detector up to the shears. To obtain the angle in the interval $[-\pi, \pi]$, we can use the following equation

$$\Delta\alpha_{\mathrm{d}_i} = \angle(\boldsymbol{d}'_{u_i}, \widetilde{\boldsymbol{d}'}_{u_i}) = \operatorname{atan2}(|||[\boldsymbol{d}'_{u_i}, \widetilde{\boldsymbol{d}'}_{u_i}]||, (\boldsymbol{d}'_{u_i})^\top \widetilde{\boldsymbol{d}'}_{u_i}). \tag{87}$$

Partitioning of an affinity. We assume that the matrix $\widetilde{\mathtt{A}}_i \in \mathbb{R}^{2\times 2}$ locally approximates the homography $\widetilde{\mathtt{H}}_{kj} \in \mathbb{R}^{3\times 3}$. The goal of comparing SIFT orientations is to determine the rotation component $\tilde{\mathtt{R}}_{u_i}$ of the affinity $\widetilde{\mathtt{A}}_i$ and compare it to the angle between the directions of corresponding keypoints.

We consider three alternatives for determining the rotational component of $\widetilde{\mathtt{A}}_i$, which are as follows:

1. QR-decomposition
2. SVD-decomposition
3. Exponential decomposition

1) *Rotation from QR-decomposition of an affinity* $\mathtt{A}_i$. Assuming that the affinity is a concatenation of a shear matrix $\mathtt{S}_{u_i}$ and a subsequent rotation with $\mathtt{R}_{u_i}$

$$\mathtt{A}_i = \mathtt{R}_{u_i}\mathtt{S}_{u_i} \tag{88}$$

the classical QR-decomposition is defined as

$$\mathtt{R}_{\mathrm{qr},\mathrm{A}_i} := \mathtt{R}_{u_i} \quad \text{with} \quad [\mathtt{R}_{u_i}, \mathtt{S}_{u_i}] = \mathrm{qr}(\mathtt{A}_i)\,. \tag{89}$$

In case the affinity is defined by the reverse sequence, i.e.

$$\mathtt{A}_i = \mathtt{S}_{u_i}\mathtt{R}_{u_i} \tag{90}$$

the QR decomposition of the transposed matrix must be taken

$$\mathtt{R}_{\mathrm{qr},\mathrm{A}_i^\top} := \mathtt{R}_{u_i}^\top \quad \text{with} \quad [\mathtt{R}_{u_i}, \mathtt{S}_{u_i}] = \mathrm{qr}(\mathtt{A}_i^\top)\,. \tag{91}$$

If there are no shears, i.e., the shear matrix is a scaled unit matrix, the two rotations $\mathtt{R}_{\mathrm{qr},\mathrm{A}_i}$ and $\mathtt{R}_{\mathrm{qr},\mathrm{A}_i^\top}$ are the same, otherwise, they differ.

2) *Rotation from SVD-decomposition of* $\mathtt{A}_i$. Assuming that the affinity is decomposable as two rotations sandwiching an individual scaling,

$$\mathtt{A}_i = \mathtt{U}_{u_i}\mathtt{S}_{u_i}\mathtt{V}_{u_i}^\top \quad \text{with} \quad \mathtt{S}_{u_i} = \begin{bmatrix} s_{u_i,1} & 0 \\ 0 & s_{u_i,2} \end{bmatrix}, \tag{92}$$

where the shears are represented by the rotation $\mathtt{V}_{u_i}$ and the ratio $s_{u_i,1}/s_{u_i,2}$. The SVD yields the rotation

$$\mathtt{R}_{\mathrm{svd},\mathrm{A}_i} := \mathtt{U}_{u_i}\mathtt{V}_{u_i}^\top \quad \text{with} \quad [\mathtt{U}_{u_i}, \mathtt{S}_{u_i}, \mathtt{V}_{u_i}] = \mathrm{svd}(\mathtt{A}_i)\,. \tag{93}$$

Transposing $\mathtt{A}_i$ does not change the resulting rotation. However, the rotation resulting from the SVD-decomposition is only identical to that of the QR-decomposition if the affinity is a scaled rotation.

3) *Rotation from an exponential decomposition.* Another approach for deriving the rotation component is by using the matrix exponential. The affinity $\mathtt{A}_i$ can be written as an exponential of a matrix $\mathtt{B}_i$, i.e.,

$$\mathtt{A}_i = \mathrm{e}^{\mathtt{B}_i} \tag{94}$$

If the matrix $\mathtt{B}_i$ is zero, i.e., $\mathtt{B}_i = \mathtt{0}$, then the affinity is a unit transformation. We can now decompose the exponent additively using the following form

$$\mathtt{B}_i = \sum_j p_{ij} \mathtt{B}_{ij} \tag{95}$$

where the four basic 2×2 matrices are

$$\mathtt{B}_{i1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathtt{B}_{i2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{96}$$

$$\mathtt{B}_{i3} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathtt{B}_{i4} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{97}$$

Hence

$$\mathtt{A}_i = \mathrm{e}^{p_{i1}\mathtt{B}_{i1} + p_{i2}\mathtt{B}_{i2} + p_{i3}\mathtt{B}_{i3} + p_{i4}\mathtt{B}_{i4}}. \tag{98}$$

If we consider each summand individually, the four parameters correspond to: (1) scaling with $\log p_{i1}$, (2) rotation by p_{i2} [rad], (3) first shear, which is opposite scaling of axes, and (4) second shear, which is opposite rotation of axes. The rotational component can be obtained using the relation

$$\mathtt{R}_{u_i} = \exp(p_{i2}\mathtt{B}_{i2}). \tag{99}$$

Furthermore, for the first shear, we explicitly have

$$\exp\left(\begin{bmatrix} 0 & p_{i4} \\ p_{i4} & 0 \end{bmatrix}\right) = \begin{bmatrix} \mathrm{e}^{-p_{i4}/2} + \mathrm{e}^{p_{i4}/2} & \mathrm{e}^{p_{i4}/2} - \mathrm{e}^{-p_{i4}/2} \\ \mathrm{e}^{-p_{i4}/2} - \mathrm{e}^{-p_{i4}/2} & \mathrm{e}^{-p_{i4}/2} + \mathrm{e}^{p_{i4}/2} \end{bmatrix} \tag{100}$$

$$\overset{q_{i4}=\mathrm{e}^{p_{i4}/2}}{=} \begin{bmatrix} q_{i4} + 1/q_{i4} & q_{i4} - 1/q4 \\ q_{i4} - 1/q_{i4} & q_{i4} + 1/q_{i4} \end{bmatrix}. \tag{101}$$

This representation exhibits high symmetry, with the additive terms being invariant with respect to their sequence. Furthermore, the scaled rotation is independent of the presence of shears. However, since the exponent of two matrices is the product of the two matrices only if they commute, that is

$$\exp(\mathtt{A} + \mathtt{B}) = \exp(\mathtt{A}) \exp(\mathtt{B}) \quad \text{only if} \quad \mathtt{AB} = \mathtt{BA}, \tag{102}$$

the interpretation of the elements in the exponent is not independent of the presence of the other elements. It is only possible to exchange a common scaling with the other components, as is the case for scaled rotation. Additionally, we can define the rotational component using (99) and obtain p_{i2} as

$$p_{i2} = (B_i(2,1) - B_i(1,2))/2 \quad \text{with} \quad \mathtt{B}_i = \log(\mathtt{A}_i) \tag{103}$$

where $\log(\mathtt{A}_i)$ is the matrix logarithm of $\mathtt{A}_i$. Therefore, we can identify the existence of shears by checking whether

$$d_s^2 = ||\begin{bmatrix} p_{i3} & p_{i4} \end{bmatrix}||= p_{i3}^2 + p_{i4}^2 = 0 \tag{104}$$

Additionally, since a scaled rotation has condition number equal one, we can use the condition number to identify the lack of shears, namely if $\text{cond}(\mathtt{A}_i) = 1$. For not too large shears, the condition number and the degree of shears $d_{s_i}^2$ are approximately the same

$$d_{s_i}^2 \approx \text{cond}(\mathtt{A}_i)\,. \tag{105}$$

7.4 Evaluation

We will demonstrate the steps to determine the bias and variance of angular, scale, and positional transformations on SIFT feature points. The positional uncertainty of SIFT keypoints is known to be approximately 1/3 pixel (see Förstner [60] p.681, and Laebe [134] Tab.6). However, in our experiments, we obtained an estimated standard deviation of $\hat{\sigma}_{u_i} \approx 0.67$ pixels, which is a factor of two larger than expected. This could be due to accepting small outliers. Currently, there is no published research on the uncertainty of the orientations and scales. However, the OpenCV implementation of the SIFT detector uses an orientation histogram with 36 bins, where the size of each bin is 10 degrees. Assuming an average standard deviation of less than three times the rounding error ($10^\circ/\sqrt{12} \approx 2.89^\circ$) – i.e., three times the standard deviation of a uniform distribution in the range $[-5^\circ, +5^\circ]$ – the average standard deviation of the angular difference αu_i is approximately $\hat{\sigma}_{\alpha_{u_i}} \approx 3 \cdot (2.89 \cdot \sqrt{2}) \approx 12^\circ$. While this is a quite large uncertainty, it can be used in cases where the rotation between keypoints is large. The augmenting factor 3 is meant to roughly take into account other effects than the rounding. In practice, the experiments we conducted on 4.3M correspondences with $\text{cond}(\mathtt{A}_i) < 1.2$ lead to an estimate of the standard deviation of the SIFT orientation, $\hat{\sigma}_{\alpha_{u_i}}$, of approximately 7.9°. This corresponds to

$$\hat{\sigma}_{\phi_i} = 7.9^\circ/\sqrt{2} = 5.5^\circ\,. \tag{106}$$

Empirically, the OpenCV implementation of the SIFT detector yields a standard deviation of $\hat{\sigma}_{\rho_i} = 0.51$ for scale estimates. Therefore the scales from the detector may deviate on average by a factor of $1.6 \approx \exp(0.51)$ in both directions.

7.4.1 Composition of reference transformations

To estimate the uncertainty of the feature point transformations between images, a large-scale dataset of ground truth (GT) transformations and corresponding

measured feature points is required. We assume that the GT transformations are given by the affinity matrices estimated from homographies. To calculate these homographies, we first identify all significant planes in 3D from the reconstructions of the 1DSfM dataset introduced by Wilson [135], which we reconstructed using COLMAP. To ensure the accuracy and coherency of the reconstructions, we manually checked their quality and filtered the degenerated ones. The scales of the reconstructions were obtained using the ruler tool of Google Maps [136].

To extract keypoints, we utilized SIFT features introduces in Lowe [61] as implemented in OpenCV [137] with RootSIFT [138] descriptors. The 3D planes were then segmented using the Progressive-X$^+$ algorithm from Barath [139]. We collected homographies that link two views of a real 3D planar surface, consistent with the camera motion, and estimable from their GT correspondences using the standard normalized DLT algorithm described in Hartley [33]. Correspondences are considered inliers if the reprojection error

$$\epsilon_{u_i} = ||\boldsymbol{u}'_i - \text{h2a}(\mathtt{H}_{kj}\,\text{a2h}(\boldsymbol{u}_i))^\top||_2 \tag{107}$$

is smaller than a δ-pixel threshold. The operators a2h and h2a convert vectors between their affine and homogeneous representation. We consider only homographies with more than 10 inliers and reject those estimated by the DLT algorithm $\mathtt{H}'_{kj}$ (decomposed by Malis [140]) with an error $\epsilon_{\mathtt{R}'_{kj}} > 3°$ or $\epsilon_{t'_{kj}} > 3°$, expressed as

$$\epsilon_{\mathtt{R}'_{kj}} = (180/\pi)\arccos\left(\left(\text{tr}\left(\mathtt{R}'_{kj}\mathtt{R}_{kj}^\top\right) - 1\right)/2\right) \tag{108}$$

$$\epsilon_{t'_{kj}} = (180/\pi)\arccos(\boldsymbol{t}_{kj}^\top\boldsymbol{t}'_{kj})(|\boldsymbol{t}_{kj}||\boldsymbol{t}'_{kj}|) \tag{109}$$

The GT dataset used for estimating the uncertainty of transformations includes 1k planes observed in 10k images, resulting in roughly 226k homographies and 6.1M correspondences. Further details on the composition of the dataset and the evaluation of state-of-the-art homography estimation methods can be found in **[68]**.

7.4.2 The positional transformation uncertainty

The positional transformation uncertainty is dependent on the symmetric positional residual of each keypoint pair, which is related to the mean squared reprojection error, as shown in equation eq. (75). Figure 13 shows the histogram of residuals for 6.1 million keypoint pairs. Additionally, Förstner [60] p.681, and Zeisl [141] eq. (15) have shown that the standard deviation of the keypoint depends on the detector scale. Therefore, we suggests that the positional transformation error ϵ_{u_i} is also dependent on keypoint scales s_i and s'_i. To further investigate this dependence, we clustered the symmetric positional residuals based on related s_i and s'_i scales, and measured the standard deviation for individual bins. The results are shown in Figure 14.

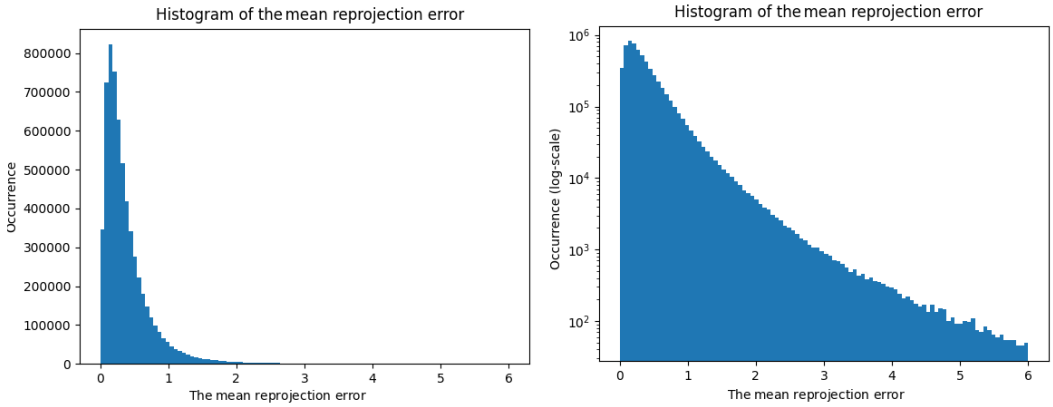


Figure 13: The residuals ϵ_{u_i} of 6.1M keypoint pairs. The right histogram shows the logarithmic scale of the occurrence to visualize the distribution of the residuals. Measured standard deviation $\hat{\sigma}_{u_i} \approx 0.67$ pixels. The STD is a factor two larger, than expected, which might result from accepting small outliers.

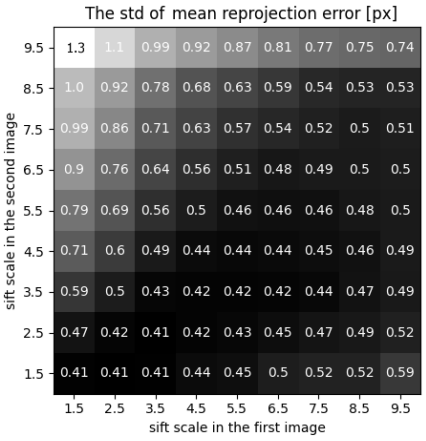


Figure 14: The figure shows the standard deviation of ϵ_{u_i} for individual combinations of keypoint scales s_i and s'_i. The plot clearly demonstrates the dependence of reprojection accuracy on the scale of the related keypoints.

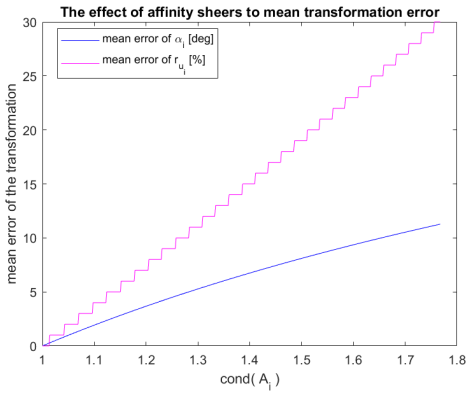


Figure 15: The plot shows the approximate effect of the condition number of the affinity matrix on the angular error and scale difference.

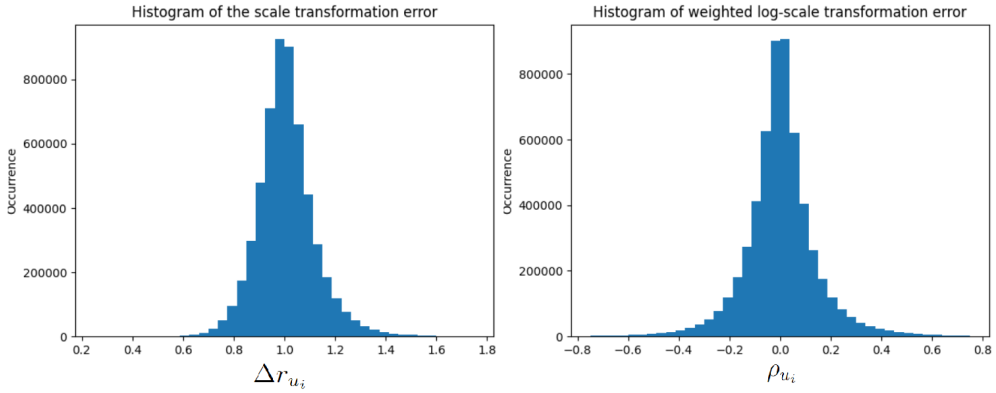


Figure 16: The histogram of the scale transformation ratio Δr_{u_i} and the weighted log-ratio ρ_{u_i} on 5.6M keypoint pairs.

7.4.3 The scale ratio uncertainty

We exclude cases where the affinity matrix has a condition number $\text{cond}(\mathtt{A}_i) > 1.5$, as the shears are assumed to have too large an impact on the scales. We focus on analyzing cases with small scale ratios, i.e., values $\tilde{r}_{u_i} \in [0.5, 2]$. This interval contains 99.62% of the keypoint pairs i.e., the scale statistics of the remaining 5.6M keypoint pairs are shown in Figure 16.

7.4.4 The angular transformation uncertainty

To evaluate the rotations $\Delta\alpha_{u_i}$ of the keypoint pairs, we restrict the samples to those with condition number $\text{cond}(\mathtt{A}_i) < \delta_{\text{cond}}$. The dataset contains approximately 4.3M of correspondences for $\delta_{\text{cond}} = 1.2$, and 5.8M correspondence for $\delta_{\text{cond}} = 1.5$. The total number of correspondences is 6.1M. In both examples, the number of samples allows a reliable estimate of the feature point's uncertainties. For small slopes of the 3D plane normal seen by stereo pair, the condition number of the affinity

$$\text{cond}\left(\mathbf{A}_{\text{shear}}\mathbf{A}_{\text{scale-diff}}\right) = \text{cond}\left(\begin{bmatrix}1 & a\\ 0 & 1\end{bmatrix}\begin{bmatrix}1+s & 0\\ 0 & 1\end{bmatrix}\right) \approx 1 + \sqrt{s^2 + a^2}. \tag{110}$$

The visualization of the mean error caused by the shears for individual condition numbers of the affinity matrix is in Fig. 15. The analysis of the shears describes their influence on estimated standard deviations of feature points. We expect approximately 3.7° of angular error, and 7% of scale difference at mean for $\delta_{\text{cond}} = 1.2$. For $\delta_{\text{cond}} = 1.5$, the mean deviations are 8° of angular error, and 18% of scale difference. Note that most of the correspondences in the dataset have an affinity matrix with a smaller condition number, see Fig. 17. Therefore the effect of sheers

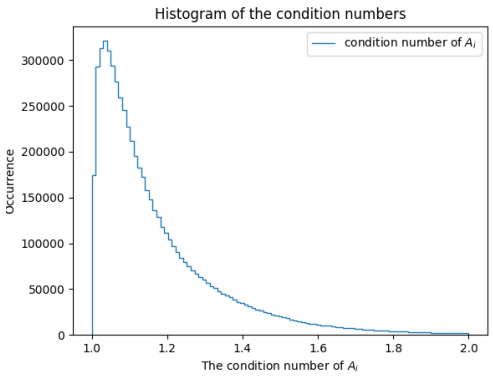


Figure 17: The histogram of condition numbers of $\mathtt{A}_i$. Approx. 70% of samples have condition number smaller than 1.2 and 95% of them have condition number smaller than 1.5.

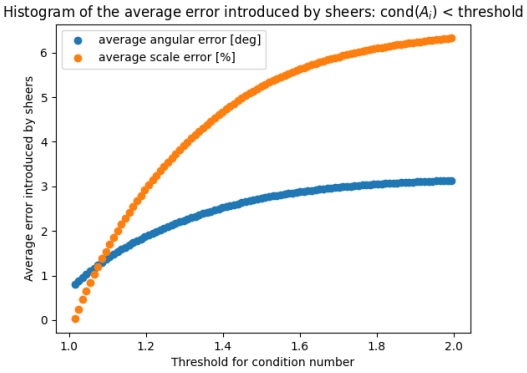


Figure 18: The average error introduced by sheers of $\mathtt{A}_i$ for correspondences with condition number smaller than the specified threshold.

will be smaller on average when we assume all correspondences up to selected threshold δ_{cond}. The mean average influence of sheers can be approximated by weighting individual errors by their relative number of occurrences. Therefore, we expect approximately 1.8° of angular transformation error and 2.9% of scale difference in mean for all samples that fulfill $\text{cond}(\mathtt{A}_i) < 1.2$. For $\text{cond}(\mathtt{A}_i) < 1.5$, the mean deviations are 2.8° of angular transformation error, and 5.1% of scale difference, see Figure 18. The effect of changing the condition number threshold on the angular transformation error $\Delta\alpha_i$ is the following. For $\delta_{\text{cond}} = 1.2$, the standard deviation of the angular transformation is 7.9°, which corresponds to the standard deviation 5.5°. For $\delta_{\text{cond}} = 1.5$, the angular transformation has a standard deviation of 8.3°, which corresponds to the directional uncertainty 5.9°.

Furthermore, the histogram of the angular transformation error $\Delta\alpha$ in Fig. 19 confirms that for $\text{cond}(\mathtt{A}_i) < 1.2$ leads the difference of individual decomposition methods to similar average residuals as the difference of directional vectors.

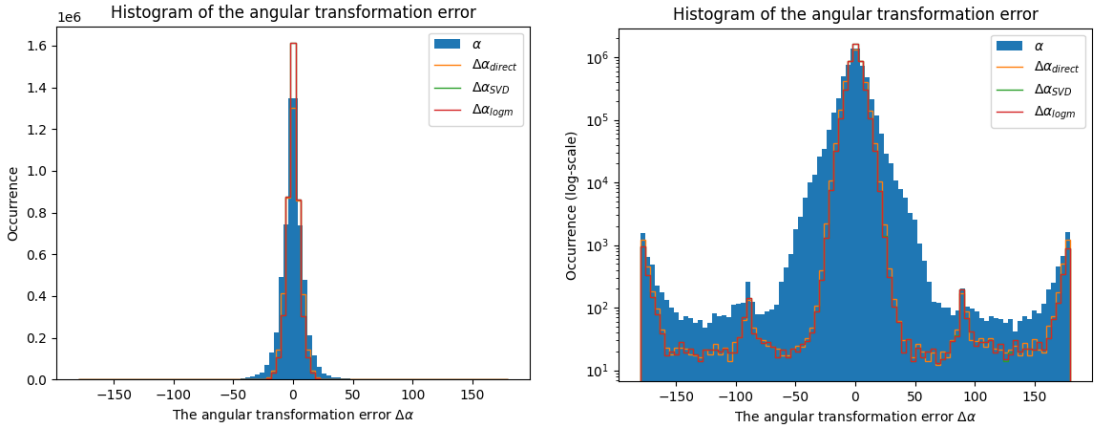


Figure 19: The histogram shows the angular transformation error $\Delta\alpha$ plotted on top of the angles α. The transformation was measured using three different methods: (1) as the angle between directional vectors, $\Delta\alpha_{\text{direct}}$ (eq. (87)), (2) by subtracting the reference angular transformation decomposed by SVD, $\Delta\alpha_{\text{SVD}}$ (eq. (93)), and (3) by subtracting the ground truth angular transformation obtained from the exponential analysis, $\Delta\alpha_{\text{logm}}$ (eq. (99)). We considered 4.3M correspondences with $\text{cond}(\widetilde{\mathtt{A}}_i) < 1.2$, and the standard deviation is $\hat{\sigma}_\alpha \approx 7.9°$, which is approximately two times the rounding error.

8 Uncertainty in SfM

SfM estimates the reconstruction $\hat{\theta}$ from feature points. In the scope of this section, we describe new techniques used to propagate the uncertainty of the feature points (described previously) into the uncertainty of estimated solutions of geometrical problems. The two main tasks are discussed: 1) the propagation of uncertainty into the geometric solutions for minimal problems (e.g., an estimate of the uncertainty of the homography matrix), and 2) the propagation of uncertainty into the reconstruction using the constraints of a projection equation. The uncertainty of estimated parameters allows for better analysis of degenerate solutions and the determination of inaccurate reconstruction parts. Moreover, we employ the uncertainty estimates in the applications described in the following Sec. 9.

8.1 Uncertainty of minimal problems

Estimating the geometric relationship between two cameras, such as homography matrix, is fundamental in computer vision. Each relation between cameras is described by a set of constraints that the geometrical model must fulfill. So-called minimal solvers estimate the solutions consistent with a related set of constraints from a minimal number of correspondences. Most minimal geometric problems in computer vision are in form $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{0}$ where $\boldsymbol{y}$ realize observations and $\boldsymbol{x}$ parameters. The classical variance propagation for implicit functions leads to

$$\Sigma_{x,x} = \mathtt{B}^{-1}\mathtt{A}\Sigma_{yy}\mathtt{A}^{\top}\mathtt{B}^{-\top} \qquad \text{with} \quad \mathtt{A} = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{y}} \quad \text{and} \quad \mathtt{B} = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{x}}\,. \tag{111}$$

The covariance matrix Σ_{yy} refers to the input measurements (e.g., keypoint coordinates or affinity correspondences), and the covariance matrix Σ_{xx} refers to the model parameters.

However, minimal problems usually have the number of constraints $\boldsymbol{f}$ smaller than the number of model parameters, and hence the matrix $\mathtt{B}$ cannot be inverted. We propose redefining the implicit function by adding constraints $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ between the model parameters to address this issue. In other words, we use the extended implicit function with their Jacobians, given by

$$\begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) \\ \boldsymbol{h}(\boldsymbol{x}) \end{bmatrix} = \mathbf{0} \qquad \text{with} \quad \mathtt{A} = \begin{bmatrix} \partial \boldsymbol{f}/\partial \boldsymbol{y} \\ \mathtt{0}^{\top} \end{bmatrix}, \quad \mathtt{B} = \begin{bmatrix} \partial \boldsymbol{f}/\partial \boldsymbol{x} \\ \partial \boldsymbol{h}/\partial \boldsymbol{x} \end{bmatrix}. \tag{112}$$

By adding constraints $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ between the model parameters, we obtain exactly the same number of constraints as parameters, which ensures that the matrix $\mathtt{B}$ is regular (except for critical geometric configurations).

However, one more step is needed to have a covariance matrix comparable to the one estimated by MC simulation. The theoretical covariance matrix $\bar{\Sigma}_{\mathrm{T}} := \hat{\Sigma}_{xx}$

cannot be directly compared with any empirically estimated covariance matrix $\bar{\Sigma}_{\text{E}}$, as $\bar{\Sigma}_{\text{T}}$ is singular while $\bar{\Sigma}_{\text{E}}$ is usually regular. The empirical covariance matrix can be obtained from the outputs of the minimal solver by repeatedly distorting the inputs based on an input covariance matrix of the observations. To obtain comparable covariance matrices $\Sigma_{\text{T}} \approx \Sigma_{\text{E}}$, we propose to regularize both by projecting them onto the orthonormal basis of the column space of the theoretical covariance matrix $\bar{\Sigma}_{\text{T}}$. Such transformation can be calculated by finding the nullspace of the transposed nullspace of $\bar{\Sigma}_{\text{T}}$, i.e.,

$$\mathtt{J}_{\text{reg}} = \text{null}(\text{null}(\bar{\Sigma}_{\text{T}})^{\mathsf{T}}) \,. \tag{113}$$

The matrix $\mathtt{J}_{\text{reg}}$ project both covariance matrices to the desired orthonormal basis as follows

$$\Sigma_{\text{T}} = \mathtt{J}_{\text{reg}}^{\top} \bar{\Sigma}_{\text{T}} \mathtt{J}_{\text{reg}} \tag{114}$$

$$\Sigma_{\text{E}} = \mathtt{J}_{\text{reg}}^{\top} \bar{\Sigma}_{\text{E}} \mathtt{J}_{\text{reg}} \,. \tag{115}$$

The following paragraphs provide a list of the minimal set of constraints $\boldsymbol{f}$, $\boldsymbol{h}$ used by minimal solvers utilized in experimental evaluation.

8.1.1 Homography estimation

Estimating planar homography $\mathtt{H} \in \mathbb{R}^{3\times3}$ is a well-studied problem, with simple linear solutions from point and affine correspondences. Here, we assume that affine correspondence is a region defined by its center $\boldsymbol{u}_i$ and shape $\mathtt{A}_{u_i} \in \mathbb{R}^{2\times2}$. Since each PC gives two linear constraints on $\mathtt{H}$, and each AC gives six linear constraints on $\mathtt{H}$, the minimal number of correspondences necessary to estimate the unknown homography is either 4PC or 1AC+1PC. Both the well-known 4PC [33] (the DLT algorithm) and the 1AC+1PC [120, 121] solvers solve a system of eight linear equations in nine unknowns and are therefore equivalent in terms of efficiency. Assuming two ACs, one can choose another subset of constraints used for the uncertainty propagation, i.e., 1AC + rotation and scale of the second AC instead of one PC.

To propagate the uncertainty, we add a constraint on the parameters $\mathtt{H}$ in the form $\boldsymbol{h}_{\text{H}} = ||\text{vec}(\mathtt{H})||_2 - 1 = 0$ for both approaches. This constraint avoids the trivial all-zeros solution.

8.1.2 Fundamental matrix estimation

The problem of estimating the relative pose of two uncalibrated cameras, i.e., estimating the fundamental matrix $\mathtt{F} \in \mathbb{R}^{3\times3}$ can be computed by a 7PC solver, see Hartley [33]. The fundamental matrix $\mathtt{F}$ has seven degrees of freedom, since it is a 3×3 singular matrix defined up to scale. Each epipolar constraint, eq. (35), for $m \in$

$\{1, \ldots, 7\}$ gives one linear constraint on $\mathtt{F}$. Therefore, we have 7 constraints from 7PC and, at the same time, 9 parameters of $\mathtt{F}$. The second approach we employ is using 2AC + 1PC correspondences. Each AC gives three linear constraints on the epipolar geometry as in Barath [142]. The solver itself rewrites the constraints in eq. (116), and eq. (117) in a matrix form as $\mathtt{M}_{\mathrm{F}}\mathrm{vec}(\mathtt{F}) = \mathbf{0}$, to find a 2-dimensional null-space of the matrix $\mathtt{M}_{\mathrm{F}}$. The unknown fundamental matrix is parameterized as $\mathtt{F} = \lambda\mathtt{F}_1 + \mathtt{F}_2$, where $\mathtt{F}_1$ and $\mathtt{F}_2$ are matrices created from the 2-dimensional null-space of $\mathtt{M}_{\mathrm{F}}$. The λ is found to fulfill the constraint $\det(\mathtt{F}) = 0$. In other words, the solver finds a solution of a polynomial of degree three in one unknown λ.

The uncertainty propagation employs seven linear constraints, i.e., six from 2AC and one from a PC, and two constraints on the fundamental matrix. The constraints are (1) for three-point correspondences

$$f_{\mathrm{m}} = \begin{bmatrix} \boldsymbol{u}_{2,m}^{\top} & 1 \end{bmatrix} \mathtt{F} \begin{bmatrix} \boldsymbol{u}_{1,m} \\ 1 \end{bmatrix} = 0\,, \tag{116}$$

(2) two pairs of affine constraints ([142] eq. (8))

$$\boldsymbol{f}_{\mathrm{i}} = [\mathtt{E}_2|\mathbf{0}_2]\,\mathtt{F} \begin{bmatrix} \boldsymbol{u}_{1,i} \\ 1 \end{bmatrix} + [\mathtt{A}_{u_i}^{-\top}|\mathbf{0}_2]\,\mathtt{F}^{\top} \begin{bmatrix} \boldsymbol{u}_{2,i} \\ 1 \end{bmatrix} = \mathbf{0}\,, \tag{117}$$

(3) the two constraints $h_{\mathrm{F}} := [h_{\mathrm{F},1}, h_{\mathrm{F},2}]^{\top}$ of the parameters

$$h_{\mathrm{F},1}(\mathtt{F}) = \det(\mathtt{F}) = 0 \tag{118}$$

$$h_{\mathrm{F},2}(\mathtt{F}) = ||\mathrm{vec}(\mathtt{F})||_2 - 1 = 0\,. \tag{119}$$

8.1.3 Essential matrix estimation

The problem of estimating the unknown essential matrix $\mathtt{E} \in \mathbb{R}^{3\times3}$, which describes the relative pose of two calibrated cameras, has five degrees of freedom and nine parameters. There are two main approaches to estimate $\mathtt{E}$: the 5PC solver introduced by Nister [34] and the 2AC solver from Brath [122] or Eichhardt [74]. The 5PC solver uses five point correspondences, each providing one linear constraint on $\mathtt{E}$. The 2AC solver, on the other hand, utilizes two affine correspondences, each giving three linear constraints on $\mathtt{E}$ as in [122]. Thus, two ACs provide more constraints than degrees of freedom, resulting in an over-constrained system of equations. One approach is to use just five out of six constraints, while another approach utilized in experimental evaluation is to calculate an over-constrained system of equations [122].

We describe two possible methods to propagate the uncertainty when using PCs and one for ACs. One straightforward solution is to use the five PC constraints

described in eq. (37) and add four constraints $\boldsymbol{h}_{\mathrm{E}} = [h_{\mathrm{E},1}, h_{\mathrm{E},2}, \boldsymbol{h}_{\mathrm{E},(3,4)}]^{\mathsf{T}}$ on $\mathrm{vec}(\mathtt{E})$:

$$h_{\mathrm{E},1}(\mathtt{E}) = \det(\mathtt{E}) = 0 \tag{120}$$
$$h_{\mathrm{E},2}(\mathtt{E}) = ||\mathrm{vec}(\mathtt{E})||_2 - 1 = 0 \tag{121}$$
$$\boldsymbol{h}_{\mathrm{E},(3,4)}(\mathtt{E}) = 2\mathtt{E}\mathtt{E}^{\top}\mathtt{E} - \mathrm{tr}(\mathtt{E}\mathtt{E}^{\top})\mathtt{E} = \mathtt{0}\,. \tag{122}$$

The vector $\boldsymbol{h}_{\mathrm{E},(3,4)}(\mathtt{E})$ represents the nine trace constraints of $\mathtt{E}$, of which only two are independent. In general, we can choose any two of these nine constraints (with the exception of singular cases like $\mathtt{E} = [1;0;0]_{\times}$) so that the constraints in $\boldsymbol{h}_{\mathrm{E}}$ are independent.

The second approach for estimating the relative pose of two calibrated cameras involves using a minimal set of parameters, such as a unit translation (baseline) vector $\boldsymbol{t}_{21} \in \mathbb{R}^3$ and the Euler vector $\boldsymbol{e}_{\mathrm{vec}} \in \mathrm{SO}(3)$. For this parametrization, we assume the essential matrix as defined in eq. (38), and the rotation matrix $\mathtt{R}_{21} = \mathcal{R}_e(\boldsymbol{e}_{\mathrm{vec}})$ is composed from the Euler vector $\boldsymbol{e}_{\mathrm{vec}}$ as follows:

$$\mathcal{R}_e(\boldsymbol{e}_{\mathrm{vec}}) = \mathtt{E}_3 + (1-\cos\alpha_e)\,[\boldsymbol{e}_{\mathrm{vec}}]_{\times}^2 + \sin\alpha_e\,[\boldsymbol{e}_{\mathrm{vec}}]_{\times} \qquad \text{where} \quad \alpha_e = \sqrt{\boldsymbol{e}_{\mathrm{vec}}^{\top}\boldsymbol{e}_{\mathrm{vec}}}\,. \tag{123}$$

This minimal representation leads to five points constraints and one constraint of the baseline

$$h_t(\boldsymbol{t}_{21}) = ||\boldsymbol{t}_{21}||_2 - 1 = 0\,. \tag{124}$$

The minimal problem using 2AC has three constraints for each correspondence. The PCs are realized as in eq. (37), and two constraints of AC are provided by eq. (9) and (10) in [142]. Next we assume the constraint eq. (124) of $\boldsymbol{t}_{21}$. In sum, this leads to six unknowns (i.e., $\boldsymbol{t}_{21} \in \mathbb{R}^3, \boldsymbol{e}_{\mathrm{vec}} \in \mathbb{R}^3$) and seven constraints. Therefore, we suppress one equation of the form eq. (10) in [142] to have the same number of constraints and parameters.

8.1.4 Essential matrix + focal length estimation

The essential matrix with a focal length has six degrees of freedom and nine parameters. It can be described as $\bar{\mathtt{E}} = \bar{\mathtt{K}}_2^{-\top}\,\mathtt{E}\,\bar{\mathtt{K}}_1^{-1}$, where

$$\bar{\mathtt{K}}_l = \begin{bmatrix} f_l & 0 & 0 \\ 0 & f_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{125}$$

is the l-th camera calibration matrix assuming focal length f_l and principal point $\boldsymbol{u}_{\mathrm{pp,l}} = \mathbf{0}_2$. We assume either 6PC or 2AC as input. The point constraints are the same as for the essential matrix eq. (37). Further, we used the constraints $h_{\mathrm{F},1}$, $h_{\mathrm{F},2}$ that are the same as for the fundamental matrix and h_{Ef} that corresponds to eq. (18) from [143].

8.2 Uncertainty of reconstruction

Estimating the uncertainty of a reconstruction (e.g., camera poses and 3D points) is a critical tool for evaluating its quality, guiding the reconstruction process, and comparing the optimal setup of hyper-parameters. To estimate the uncertainty of the reconstruction θ, we can propagate the uncertainty from the feature points to the estimated solution of the relative pose problem. Then, we can propagate it further to the 3D points using triangulation constraints and, finally, from the 3D points to the camera poses using the absolute pose solver and its geometric restrictions. However, we can simplify this process by employing the geometric constraints between 3D points, camera poses, and keypoints, as in bundle adjustment (41). This section builds on the camera model notation and bundle adjustment derivation covered in the key concepts. We provide a guide for propagating uncertainty from keypoints to the parameters of the large-scale reconstruction.

Let us start from the normal equation, eq. (47), in a form

$$\mathtt{M}\,\Delta\boldsymbol{\theta} = \boldsymbol{m} \tag{126}$$

$$\mathtt{M} = \mathtt{J}^\top \Sigma_{\hat{\epsilon},\hat{\epsilon}}^{-1} \mathtt{J} \qquad \boldsymbol{m} = \mathtt{J}^\top \Sigma_{\hat{\epsilon},\hat{\epsilon}}^{-1} (\hat{\boldsymbol{u}} - \boldsymbol{u})\,. \tag{127}$$

The coordinate system of the reconstruction is defined up to a similarity transformation with seven degrees of freedom, which means that the rank of the Jacobian matrix $\mathtt{J}$ is $K - 7$. As a result, the Fisher information matrix $\mathtt{M} \in \mathbb{R}^{K \times K}$ is rank-deficient. Following the backward uncertainty propagation of the non-linear function eq. (20), we can see that the pseudo-inversion of M realizes the covariance matrix of the inner geometry of the reconstruction θ. The next subsections show two approaches to estimate the pseudo-inversion of $\mathtt{M}$ even for large-scale reconstructions.

8.2.1 Taylor expansion algorithm

The first approach is based on the Taylor expansion (TE) algorithm, which partially corresponds to the Levenberg–Marquardt (LM) algorithm. This approach solves the inversion of $\mathtt{M}$ by adding a damping term, i.e., by assuming $\mathtt{M} + \gamma \mathtt{E}_K$. We can denote this as a function of γ, i.e.,

$$g(\gamma) = (\mathtt{M} + \gamma \mathtt{E}_K)^{-1}. \tag{128}$$

Since the damping term introduces some error, the idea is to estimate the inversion of $\mathtt{M}$ using the Taylor series of $g(\gamma)$ around $\gamma = 0$. To achieve this, we need to compute the i-th derivative of g with respect to γ, which is given by

$$\frac{\partial^i g(\gamma)}{\partial \gamma^i} = (-1)^i\, i (\mathtt{M} + \gamma E_K)^{-(i+1)}\,. \tag{129}$$

Using the derivative of $g(\gamma)$, the Taylor series expansion around $\gamma = 0$ equal

$$\sum_{i=0}^{\infty} \left(\frac{(-\gamma)^i}{i!} \frac{\partial^i g(\gamma)}{\partial \gamma^i} \right) \tag{130}$$

which allows us to express the inversion of $\mathtt{M}$ as a sum of recursive functions

$$g(0) = (\mathtt{M} + \gamma \mathtt{E}_K)^{-1} + \sum_{t=1}^{\infty} \left(\frac{\gamma^t}{(t-1)!} (\mathtt{M} + \gamma \mathtt{E}_K)^{-(t+1)} \right) . \tag{131}$$

Thus, we can substitute $M_{\text{inv}} = (\mathtt{M} + \gamma \mathtt{E}_K)^{-1}$ and calculate $M_{\text{inv}}^{(t+1)} = M_{\text{inv}}^t M_{\text{inv}}$, which involves only matrix multiplications.

8.2.2 Nullspace bounding method

This method assumes additional constrains on the reconstruction, i.e., $\boldsymbol{h}_\theta(\boldsymbol{\theta}) = 0$. In general, we can assume any constraints that fix the gauge of the covariance matrix. However, to have the uncertainty of inner geometry, we would need to calculate the S-transformation eq. (26) which is computationally challenging. Therefore, an ideal approach is to fix the whole scene as done in MP inversion. In that case, the derivative of h_θ also equal the nullspace of the column space of $\mathtt{J}$, i.e.,

$$\mathtt{J}\mathtt{H}_\theta = \mathbf{0} \tag{132}$$

Using Lagrange multipliers $\boldsymbol{\lambda}$, we are minimising the function

$$g(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{2} (\mathtt{J}\Delta\boldsymbol{\theta} + \hat{\boldsymbol{u}} - \boldsymbol{u})^\top \Sigma_{uu}^{-1} (\mathtt{J}\Delta\boldsymbol{\theta} + \hat{\boldsymbol{u}} - \boldsymbol{u}) + \boldsymbol{\lambda}^\top (\mathtt{H}_\theta^\top \hat{\boldsymbol{\theta}}) \tag{133}$$

which has partial derivative with respect $\boldsymbol{\lambda}$ equal to zero in the optimum

$$\frac{\partial g(\boldsymbol{\theta}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathtt{H}_\theta^\top \hat{\theta} = 0. \tag{134}$$

Therefore, the constraints can be integrated into the *extended normal equation*

$$\begin{bmatrix} \mathtt{M} & \mathtt{H}_\theta \\ \mathtt{H}_\theta^\top & 0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\theta}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathtt{J}^\top \Sigma_{uu}^{-1} (\hat{\boldsymbol{u}} - \boldsymbol{u}) \\ 0 \end{bmatrix} \tag{135}$$

and allow us to express the M-P inversion as the inversion of an extended information matrix about its nullspace, i.e.

$$\begin{bmatrix} \Sigma_{\theta\theta} & - \\ - & - \end{bmatrix} = \begin{bmatrix} \mathtt{M} & \mathtt{H}_\theta \\ \mathtt{H}_\theta^\top & \mathtt{0} \end{bmatrix}^{-1} \tag{136}$$

The remaining question is how to calculate the nullspace $\mathtt{H}_\theta$, which has asymptotic complexity is $\mathcal{O}(K^3)$ for general matrix $\mathtt{M}$. The following text shows how to utilize the geometrical constraints of the similarity transformation to estimate the nullspace in $\mathcal{O}(L)$ (where $L \ll K$). Let us assume a similarity transformation ($\boldsymbol{\theta} \to \boldsymbol{\theta}_s$) that do not change of the projection function, i.e.

$$\boldsymbol{p}(\boldsymbol{\theta}) - \boldsymbol{p}(\boldsymbol{\theta}_s) = 0 \tag{137}$$

If we assume a small change $\boldsymbol{\theta} \to \boldsymbol{\theta}_s$ and linearize the function $p(\boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$, the difference $\boldsymbol{p}(\boldsymbol{\theta}) - \boldsymbol{p}(\boldsymbol{\theta}_s)$ remain equal zero. Any such function has its total derivative multiplied by $\Delta\boldsymbol{\theta}$ equal zero. Therefore, we obtain the formula

$$\mathtt{J}\boldsymbol{\theta} - \mathtt{J}\boldsymbol{\theta}_s = \mathtt{J}\Delta\boldsymbol{\theta} = \mathtt{J}\mathtt{H}_\theta = \mathtt{0}\,. \tag{138}$$

Therefore $\Delta\boldsymbol{\theta}$ equals the nullspace $\mathtt{H}_\theta$. In other words, if we find such $\Delta\boldsymbol{\theta}$ that does not change eq. (137), we will have the basis of the nullspace $\mathtt{H}_\theta$. The order of parameters in the nullspace directly depends on the ordering of parameters in θ. We utilize the ordering from eq. (29). Every single camera has parameters in the order $\boldsymbol{P}_l = \left[\boldsymbol{e}_{\text{vec,l}}, \boldsymbol{C}_l, f_l, \boldsymbol{u}_{\text{pp,l}}, \boldsymbol{\theta}_{\text{rd,l}}\right]$, i.e., the Euler vector, camera center, focal length, principal point, and radial distortion parameters. The translation can be expressed form camera center using $\boldsymbol{t}_l = -\mathcal{R}_e(\boldsymbol{e}_{\text{vec,l}})\boldsymbol{C}_l$. Further we denote the l-th camera rotation as $\mathtt{R}_l = \mathcal{R}_e(\boldsymbol{e}_{\text{vec,l}})$ for brevity. We can express the parameters after similarity transformation realized by rotation $\mathtt{R}_s$, translation $\boldsymbol{t}_s$, and scale λ_s as

$$\begin{aligned}
\mathtt{R}_{l,s} &= \mathtt{R}_l\mathtt{R}_s^{-1} && (139)\\
\boldsymbol{C}_{l,s} &= \lambda_s\mathtt{R}_s\boldsymbol{C}_l + \boldsymbol{t}_s && (140)\\
\boldsymbol{X}_{m,s} &= \lambda_s\mathtt{R}_s\boldsymbol{X}_m + \boldsymbol{t}_s. && (141)
\end{aligned}$$

The similarity transformation does not change camera intrinsic $f_l, \boldsymbol{u}_{\text{pp,l}}, \boldsymbol{\theta}_{\text{rd,l}}$. Therefore, we can simplify the eq. (137) by assuming $f_l = 1$, $\boldsymbol{u}_{\text{pp,l}} = \mathbf{0}_2$, $\boldsymbol{\theta}_{\text{rd,l}} = \emptyset$, i.e., it holds that

$$\text{h2a}(\mathtt{R}_l(\boldsymbol{X}_m - \boldsymbol{C}_l)) - \text{h2a}(\mathtt{R}_{l,s}(\boldsymbol{X}_{m,s} - \boldsymbol{C}_{l,s})) = \mathbf{0}_2. \tag{142}$$

This equation is linear in $\boldsymbol{t}_s$ and λ_s and we can write the change $\Delta\boldsymbol{X}_m$, $\Delta\boldsymbol{C}_l$ as

$$\begin{aligned}
\Delta\boldsymbol{C}_l &= \boldsymbol{C}_l - \boldsymbol{C}_{l,s} = \boldsymbol{C}_l - \lambda_s\mathtt{R}_s\boldsymbol{C}_l + \boldsymbol{t}_s && (143)\\
\Delta\boldsymbol{X}_m &= \boldsymbol{X}_m - \boldsymbol{X}_{m,s} = \boldsymbol{X}_m - \lambda_s\mathtt{R}_s\boldsymbol{X}_m + \boldsymbol{t}_s && (144)
\end{aligned}$$

The $\Delta\boldsymbol{C}_l$, $\Delta\boldsymbol{X}_m$ are submatrices of the basis vectors of $\mathtt{H}_\theta$, see eq. (138). Next, we write the Jacobian $\mathtt{J}$ and the nullspace $\mathtt{H}_\theta$, both estimated in $\hat{\boldsymbol{\theta}}$, to have better insight on their structure

$$\mathtt{J}(\hat{\boldsymbol{\theta}}) = \frac{\partial\boldsymbol{p}(\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}} = \begin{bmatrix}
\frac{\partial\boldsymbol{p}_1}{\partial\boldsymbol{P}_1} & \cdots & \frac{\partial\boldsymbol{p}_1}{\partial\boldsymbol{P}_L} & \frac{\partial\boldsymbol{p}_1}{\partial\boldsymbol{X}_1} & \cdots & \frac{\partial\boldsymbol{p}_1}{\partial\boldsymbol{X}_M} \\
\vdots & & \vdots & \vdots & & \vdots \\
\frac{\partial\boldsymbol{p}_N}{\partial\boldsymbol{P}_1} & \cdots & \frac{\partial\boldsymbol{p}_N}{\partial\boldsymbol{P}_L} & \frac{\partial\boldsymbol{p}_N}{\partial\boldsymbol{X}_1} & \cdots & \frac{\partial\boldsymbol{p}_N}{\partial\boldsymbol{X}_M}
\end{bmatrix} \tag{145}$$

and

$$\mathtt{H}_{\theta}(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} \mathtt{H}_{P_1}^{t_s} & \mathtt{H}_{P_1}^{\mathtt{R}_s} & \mathtt{H}_{P_1}^{\lambda_s} \\ \vdots & \vdots & \vdots \\ \mathtt{H}_{P_L}^{t_s} & \mathtt{H}_{P_L}^{\mathtt{R}_s} & \mathtt{H}_{P_L}^{\lambda_s} \\ \mathtt{H}_{X_1}^{t_s} & \mathtt{H}_{X_1}^{\mathtt{R}_s} & \mathtt{H}_{X_1}^{\lambda_s} \\ \vdots & \vdots & \vdots \\ \mathtt{H}_{X_M}^{t_s} & \mathtt{H}_{X_M}^{\mathtt{R}_s} & \mathtt{H}_{X_M}^{\lambda_s} \end{bmatrix}. \tag{146}$$

The $\boldsymbol{p}_n$ realize the n-th projection equation related to a pair of $(l, m) \in \boldsymbol{S}$. There is $2N$ rows of $\mathtt{H}_{\theta}$ related to the parameters $\Delta\boldsymbol{\theta} = \{\Delta\boldsymbol{P}_1, \ldots, \Delta\boldsymbol{P}_L, \Delta\boldsymbol{X}_1, \ldots, \Delta\boldsymbol{X}_M\}$ that are changed by seven parameters of similarity transformation, i.e. $\boldsymbol{t}_s$, $\mathtt{R}_s$, and λ_s. Thus, one possible way of $\mathtt{H}_{\theta}$ calculation is to randomly chose $\boldsymbol{t}_s$, $\mathtt{R}_s$, λ_s and utilise the change of $\boldsymbol{\theta}$ with respect to individual similarity transformation parameters, e.g., $\mathtt{H}_{X_1}^{t_s} \in \mathbb{R}^{3\times 3}$ can be found as the change of $\boldsymbol{X}_1$ when changing $t_{s,1}$, $t_{s,2}$, and $t_{s,3}$ using eq. (144). To simplify this approach, we can linearize the known pieces of the function $\Delta\boldsymbol{\theta}$ according to the $\boldsymbol{t}_s$, $\mathtt{R}_s$, and λ_s parameters. The differential of $\Delta\boldsymbol{\theta}$ indicates the direction of the change in $\boldsymbol{\theta}$ evaluated at $\hat{\boldsymbol{\theta}}$ that does not affect the projection equation. Therefore, any step in this direction approximates $\Delta\boldsymbol{\theta}$, and we can assume a unit step, equivalent to directly using the partial derivatives

$$\mathtt{H}_{\theta} = \begin{bmatrix} \frac{\partial\Delta\boldsymbol{e}_{\text{vec},1}}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta\boldsymbol{e}_{\text{vec},1}}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta\boldsymbol{e}_{\text{vec},1}}{\partial\lambda_s} \\ \frac{\partial\Delta\boldsymbol{C}_1}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta\boldsymbol{C}_1}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta\boldsymbol{C}_1}{\partial\lambda_s} \\ \frac{\partial\Delta f_1}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta f_1}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta f_1}{\partial\lambda_s} \\ \frac{\partial\Delta\boldsymbol{u}_{p,1}}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta\boldsymbol{u}_{p,1}}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta\boldsymbol{u}_{p,1}}{\partial\lambda_s} \\ \frac{\partial\Delta\boldsymbol{\theta}_{\text{rd},1}}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta\boldsymbol{\theta}_{\text{rd},1}}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta\boldsymbol{\theta}_{\text{rd},1}}{\partial\lambda_s} \\ \vdots & \vdots & \vdots \\ \frac{\partial\Delta X_1}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta X_1}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta X_1}{\partial\lambda_s} \\ \vdots & \vdots & \vdots \\ \frac{\partial\Delta X_m}{\partial\boldsymbol{t}_s} & \frac{\partial\Delta X_m}{\partial\boldsymbol{e}_{\text{vec,s}}} & \frac{\partial\Delta X_m}{\partial\lambda_s} \end{bmatrix} = \begin{bmatrix} \mathtt{0}_3 & \mathtt{H}_{\mathtt{R}_1}^{\mathtt{R}_s} & \mathbf{0}_3 \\ \mathtt{E}_3 & [\boldsymbol{C}_1]_{\times} & \boldsymbol{C}_1 \\ \mathbf{0}_3^{\top} & \mathbf{0}_3^{\top} & 0 \\ \mathtt{0}_{2\times 3} & \mathtt{0}_{2\times 3} & 0 \\ \mathtt{0}_{(B+D)\times 3} & \mathtt{0}_{(B+D)\times 3} & \mathbf{0}_{B+D} \\ \vdots & \vdots & \vdots \\ \mathtt{E}_3 & [\boldsymbol{X}_1]_{\times} & \boldsymbol{X}_1 \\ \vdots & \vdots & \vdots \\ \mathtt{E}_3 & [\boldsymbol{X}_M]_{\times} & \boldsymbol{X}_M \end{bmatrix}. \tag{147}$$

Note that $\mathtt{0}_3$ notation is used for an zero matrix $\mathtt{0}_3 \in \mathbb{R}^{3\times 3}$, the $\mathbf{0}_3$ is the vector $\mathbf{0}_3 \in \mathbb{R}^3$, and $(B + D)$ is the number of radial distortion parameter. In general, all the intrinsic have the partial derivative w.r.t. similarity transformation equal zeros.

The expression eq. (147) allow us to write the nullspace directly from parameters $\boldsymbol{\theta}$ without any calculation, except for the blocks $\mathtt{H}_{\mathtt{R}_l}^{\mathtt{R}_s}$. A skew-symmetric matrix can approximate the rotation change after the linearization. However, we found it more numerically stable to calculate it from the Jacobian $\mathtt{J}$ and the known part of the nullspace $\mathtt{H}_\theta$ according to the eq. (138). The columns of $\mathtt{H}_{\mathtt{R}_l}^{\mathtt{R}_s}$ blocks are orthogonal to the rest of the nullspace $\mathtt{H}_\theta$ and also to the Jacobian $\mathtt{J}$. Therefore, the multiple of red parts of $\mathtt{J}$ and $\mathtt{H}_\theta$ in Fig. 20 minus the multiple of related green parts should equal zero matrices. This can be written as

$$\mathtt{J}_\mathtt{R}\mathtt{H}_\mathtt{R} = \mathtt{B} \tag{148}$$

where all the unknown blocks $\mathtt{H}_{\mathtt{R}_l}^{\mathtt{R}_s}$ are stacked into a matrix $\mathtt{H}_\mathtt{R} \in \mathbb{R}^{3L\times 3}$, i.e.

$$\mathtt{H}_\mathtt{R} = \begin{bmatrix} \mathtt{H}_{\mathtt{R}_1}^{\mathtt{R}_s} \\ \vdots \\ \mathtt{H}_{\mathtt{R}_L}^{\mathtt{R}_s} \end{bmatrix} . \tag{149}$$

The red sub-blocks in Fig. 20 (a), i.e.

$$\mathtt{J}_{1,l} = \frac{\partial p_1(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{e}_{\mathrm{vec,l}}} \quad \mathtt{J}_{2,l} = \frac{\partial p_2(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{e}_{\mathrm{vec,l}}} \quad \mathtt{J}_{3,l} = \frac{\partial p_3(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{e}_{\mathrm{vec,l}}} \tag{150}$$

form a block-diagonal matrix $\mathtt{J}_\mathtt{R} \in \mathbb{R}^{3L\times 3L}$

$$\mathtt{J}_\mathtt{R} = \mathrm{diag}\left(\begin{bmatrix} \mathtt{J}_{1,1} \\ \mathtt{J}_{2,1} \\ \mathtt{J}_{3,1} \end{bmatrix}, \ldots, \begin{bmatrix} \mathtt{J}_{1,L} \\ \mathtt{J}_{2,L} \\ \mathtt{J}_{3,L} \end{bmatrix} \right) , \tag{151}$$

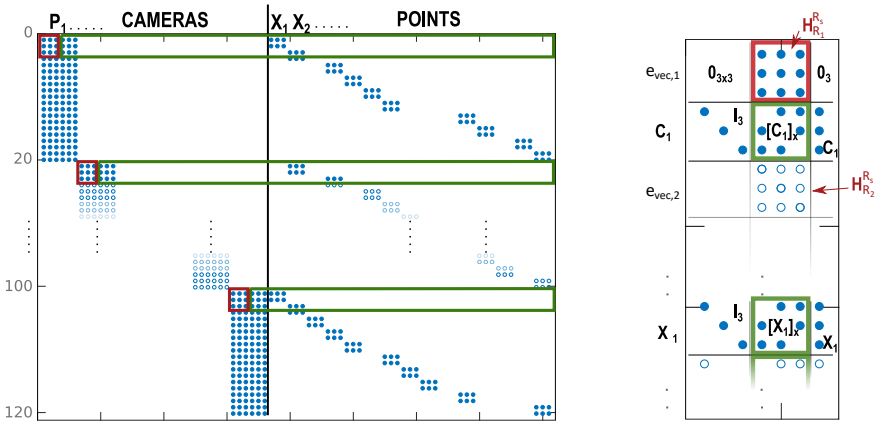
and the multiple of the green blocks in Fig. 20 (a), (b) equals $-\mathtt{B} \in \mathbb{R}^{3L\times 3}$ matrix. Finding the solution of eq. (148) can be expressed as

$$\mathtt{H}_\mathtt{R} = \mathtt{J}_\mathtt{R}^{-1}\mathtt{B} , \tag{152}$$

and requires only an inversion of L matrices of dimension 3×3 on the diagonal that are multiplied with $\mathbb{R}^{3L\times 3}$ matrix $\mathtt{B}$, i.e., the asymptotic complexity is a fixed multiple of number of cameras L instead of K^3.

8.2.3 Schur complement method

The damping term $\gamma\mathtt{E}$ in eq. (131) and the nullspace bounding in eq. (136) enable the use of the Schur complement method to compute the inversion of $\mathtt{M}$. In general,



(a) The Jacobian matrix $\mathtt{J}$ (b) The nullspace matrix $\mathtt{H}_\theta$

Figure 20: The internal structure of the matrices $\mathtt{J}$,$\mathtt{H}_\theta$ using six parameters for each camera $\boldsymbol{P}_i$, i.e., neglecting the intrinsic parameters. The matrices $\mathtt{J}_\mathtt{R}$ and $\mathtt{H}_\mathtt{R}$ are composed of the red submatrices of $\mathtt{J}$ and red submatrices of $\mathtt{H}_\theta$. The multiplication of green submatrices of $\mathtt{J}$ and the green submatrices of $\mathtt{H}_\theta$ equals $-\mathtt{B}$, leading to eq. (152).

following method is not applicable when the inverted matrix is rank deficient. As the steps are the same for both regular matrices that are inverted

$$(\mathtt{M} + \gamma \mathtt{E}_K)^{-1} \qquad \text{and} \qquad \begin{bmatrix} \mathtt{M} & \mathtt{H}_\theta \\ \mathtt{H}_\theta^\top & \mathtt{0} \end{bmatrix}^{-1} \tag{153}$$

let us demonstrate the steps for the superior method, namely the nullspace bounding method, which does not require any matrix additions or multiplications after the inversion to express the covariance matrix of the reconstruction.

The first step in the superior method is to scale the values in $\mathtt{M}$ to be in approximately the same range, which corresponds to choosing an appropriate coordinate system, such as expressing the focal length in multiples of image width instead of pixels. Since we are inverting instead of MP inverting, we can extend eq. (136) to include diagonal matrices $\mathtt{S}_a$ and $\mathtt{S}_b$, which scale the values in $\mathtt{M}$, i.e.

$$\begin{bmatrix} \Sigma_{\theta\theta} & - \\ - & - \end{bmatrix} = \begin{bmatrix} \mathtt{S}_a & 0 \\ 0 & \mathtt{S}_b \end{bmatrix} \left(\begin{bmatrix} \mathtt{S}_a & 0 \\ 0 & \mathtt{S}_b \end{bmatrix} \begin{bmatrix} \mathtt{M} & \mathtt{H}_\theta \\ \mathtt{H}_\theta^\top & 0 \end{bmatrix} \begin{bmatrix} \mathtt{S}_a & 0 \\ 0 & \mathtt{S}_b \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathtt{S}_a & 0 \\ 0 & \mathtt{S}_b \end{bmatrix} \tag{154}$$

$$\begin{bmatrix} \Sigma_{\theta\theta} & - \\ - & - \end{bmatrix} = \begin{bmatrix} \mathtt{S}_a & 0 \\ 0 & \mathtt{S}_b \end{bmatrix} \begin{bmatrix} \mathtt{M}_s & \mathtt{H}_s \\ \mathtt{H}_s^\top & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathtt{S}_a & 0 \\ 0 & \mathtt{S}_b \end{bmatrix} \tag{155}$$

$$\begin{bmatrix} \Sigma_{\theta\theta} & - \\ - & - \end{bmatrix} = \mathtt{S}\mathtt{Q}^{-1}\mathtt{S}\,. \tag{156}$$

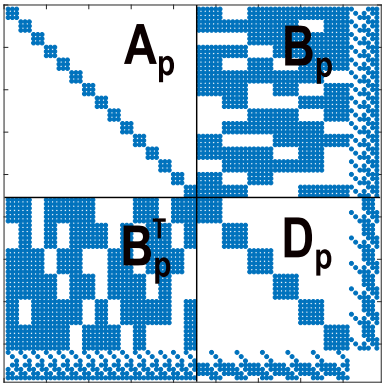


Figure 21: The internal structure of the matrix $\mathtt{Q}_\mathtt{P}$. The blue dots realize non-zero values. Each camera is realized by vector $\boldsymbol{P}_l \in \mathbb{R}^6$.

The next step is to permute the columns and rows (i.e. the order of parameters $\boldsymbol{\theta}$) in the matrix $\mathtt{Q}$ to have point parameters followed by camera parameters and the nullspace

$$\begin{bmatrix} \Sigma_{\theta\theta} & - \\ - & - \end{bmatrix} = \mathtt{SP}(\mathtt{PQP})^{-1}\mathtt{PS} = \mathtt{SP}\,\mathtt{Q}_\mathtt{P}^{-1}\,\mathtt{PS}\,. \tag{157}$$

The matrix $\mathtt{P}$ realizes the appropriate permutation. The matrix $\mathtt{Q}_\mathtt{P}$ can be decomposed into three submatrices $\mathtt{A}_\mathtt{P}$, $\mathtt{B}_\mathtt{P}$, and $\mathtt{D}_\mathtt{P}$ as vizualized on Fig. 21. These submatrices allow us to express the inversion as the block matrix inversion in from

$$\mathtt{Q}_\mathtt{P}^{-1} = \begin{bmatrix} \mathtt{A}_\mathtt{P} & \mathtt{B}_\mathtt{P} \\ \mathtt{B}_\mathtt{P}^\top & \mathtt{D}_\mathtt{P} \end{bmatrix}^{-1} = \begin{bmatrix} \mathtt{A}_\mathtt{P}^{-1} + \mathtt{A}_\mathtt{P}^{-1}\mathtt{B}_\mathtt{P}\mathtt{Z}_\mathtt{P}^{-1}\mathtt{B}_\mathtt{P}^\top\mathtt{A}_\mathtt{P}^{-1} & -\mathtt{A}_\mathtt{P}^{-1}\mathtt{B}_\mathtt{P}\mathtt{Z}_\mathtt{P}^{-1} \\ -\mathtt{Z}_\mathtt{P}^{-1}\mathtt{B}_\mathtt{P}^\top\mathtt{A}_\mathtt{P}^{-1} & \mathtt{Z}_\mathtt{P}^{-1} \end{bmatrix} \tag{158}$$

where $\mathtt{Z}_\mathtt{P}$ matrix is the symmetric Schur complement matrix of the block $\mathtt{A}_\mathtt{P}$

$$\mathtt{Z}_\mathtt{P}^{-1} = (\mathtt{D}_\mathtt{P} - \mathtt{B}_\mathtt{P}^\top\mathtt{A}_\mathtt{P}^{-1}\mathtt{B}_\mathtt{P})^{-1}\,. \tag{159}$$

Note that the symmetric block diagonal matrix $\mathtt{A}_\mathtt{P} \in \mathbb{R}^{3M\times 3M}$ is composed of $\mathbb{R}^{3\times 3}$ blocks on the diagonal, which allows for parallel inversion of these blocks. Each $\mathbb{R}^{3\times 3}$ block is related to one 3D points. Much smaller matrix $\mathtt{Z}_\mathtt{P} \in \mathbb{R}^{(K-3M)\times(K-3M)}$ is related to camera parameters. The submatrix of $\mathtt{Q}_\mathtt{P}^{-1}$ related to the points is calculated as

$$\mathtt{A}_\mathtt{P}^{-1} + \mathtt{Y}_\mathtt{P}\mathtt{Z}_\mathtt{P}^{-1}\mathtt{Y}_\mathtt{P}^\top \qquad \text{where} \quad \mathtt{Y}_\mathtt{P} = \mathtt{A}_\mathtt{P}^{-1}\mathtt{B}_\mathtt{P}\,. \tag{160}$$

Because of the symmetric matrix $\Sigma_{\theta,\theta}$, we can calculate only the upper (or lower) triangle.

The uncertainty propagation using the nullspace bounding is feasible for up to a few thousand cameras in the reconstruction. If the reconstruction is larger, the

inversion of the Schur complement matrix $\mathtt{Z}_\mathtt{P}$ appears to be numerically unstable. Let us assume a partial reconstruction with Fisher information matrix $\mathtt{M}_i$. If a new camera is registered, we can extend $\mathtt{M}_i$ about zeros for new camera parameters and new registered observations $\boldsymbol{u}_\Delta$, i.e.

$$\bar{\mathtt{M}}_i = \begin{bmatrix} \mathtt{M}_i & \mathtt{0} \\ \mathtt{0} & \mathtt{0} \end{bmatrix} . \tag{161}$$

The size of zero matrices depends on the camera dimension $\dim(\mathtt{P}_l)$ and the number of registered observations. Next we can add the Fisher information matrix $\mathtt{M}_\Delta$ realized by new camera

$$\mathtt{M}_{i+1} = \bar{\mathtt{M}}_i + \mathtt{M}_\Delta . \tag{162}$$

This expression leads to an update of the Schur complement matrix in form

$$\mathtt{Z}_{i+1} = \mathtt{Z}_i + \mathtt{Z}_\Delta . \tag{163}$$

Let us simplify the derivation by not assuming the scaling as in eq. (154) and permutation in eq. (157). The Woodbury matrix identity show that adding a new camera

$$\begin{aligned} \mathtt{Z}_{i+1}^{-1} &= (\mathtt{Z}_i + \mathtt{J}_{u_\Delta}^\top \Sigma_{u_\Delta,u_\Delta} \mathtt{J}_{u_\Delta})^{-1} && (164) \\ &= \mathtt{Z}_i^{-1} - \mathtt{Z}_i^{-1} \mathtt{J}_{u_\Delta}^\top (\Sigma_{u_\Delta,u_\Delta} + \mathtt{J}_{u_\Delta} \mathtt{Z}_i \mathtt{J}_{u_\Delta}^\top)^{-1} \mathtt{J}_{u_\Delta} \mathtt{Z}_i^{-1} && (165) \\ &= \Sigma_{P,P} - \mathtt{W}_\Delta && (166) \end{aligned}$$

leads to the subtraction of positive semi-definite matrix $\mathtt{W}_\Delta$ from the original covariance matrix of cameras, called here $\Sigma_{P,P}$ for brevity. In other words, the uncertainty of the reconstruction decrease with an increasing number of cameras. Suppose we have $\Sigma_{P,P}$ large enough (i.e., more than approximately 150 cameras) the change of $\mathtt{W}_\Delta$ becomes neglectable, i.e., we can approximate the uncertainty propagation by using reasonably large sub-scenes.

8.3 Evaluation

We conduct two sets of experiments to assess the quality of our uncertainty propagation methods. The first set of experiments involves using the chi^2 test to evaluate the propagation of uncertainty for minimal solvers. The second set of experiments focuses on testing the propagation of uncertainty from keypoints to the sparse reconstruction. We compare the performance of six algorithms on nine scenes. To assess the accuracy of the propagated covariance matrices, we compare them with the ground truth using a metric based on the relative difference between the matrices related to camera parameters. Additionally, we test the speed of each

algorithm, examine the change in the covariance matrix when a new camera is registered, and investigate the effect of assuming a smaller scene on the accuracy of the covariance matrix. We present the results in various figures, including visualizations of standard ellipses and comparisons of the errors of covariance matrices of individual methods on small scenes with respect to the ground truth.

8.3.1 Uncertainty of minimal problems

To verify the accuracy and numerical stability of the provided implementations for uncertainty propagation in minimal solvers, we used a small covariance matrix of input measurements. We propagated this covariance matrix through the solvers in two ways: (1) using uncertainty propagation to obtain the theoretical covariance matrix, and (2) using simulation to obtain the empirical covariance matrix.

In the simulation, we repeatedly distorted the input measurements according to the input covariance and ran the minimal solver to obtain outputs. These outputs were then used to estimate the empirical covariance matrix. To assess the quality of both the theoretical and empirical covariance matrices after projection onto the orthonormal basis of the column space of $\bar{\Sigma}_{\mathrm{T}}$ (i.e. the theoretical one Σ_{T}, and empirical one Σ_{E}), we use the chi^2 test statistic

$$\lambda_t = k \left(\log \left(\frac{\det(\Sigma_{\mathrm{T}})}{\det(\Sigma_{\mathrm{E}})} \right) - p + \mathrm{tr}(\Sigma_{\mathrm{E}} \Sigma_{\mathrm{T}}^{-1}) \right) \tag{167}$$

where k is the number of samples used to estimate Σ_{E}, and p is the minimum number of output parameters, i.e. $\mathrm{rank}(\Sigma_{\mathrm{E}})$. The hypothesis that Σ_{T}, and Σ_{E} follow the same distribution can be rejected if λ_t is outside the bounds defined by the cumulative distribution function of the chi^2 distribution with $(p+p^2)/2$ degrees of freedom and chosen significance level. We use a significance level of $\alpha_{\min} = 0.999$. Figure 22 shows the mean and standard deviation of λ_t estimated from 500 trials of the statistical test, where each trial estimated the empirical covariance matrix using $p = 100$ samples and $\sigma_{\mathrm{in}} = 10^{-13}$ variance of input measurements.

8.3.2 Uncertainty of reconstruction

In this section, we compare the methods developed in the theoretical part with two previous approaches: Kanatani [75] and Lhuillier [58]. In addition, we apply MP inversion (as in [75]) in several environments with different numerical precision. The comparison of the mentioned methods is from our paper **[99]**. The Taylor expansion approach was published in **[97]**, and the ground truth datasets were composed in the scope of **[98]**. A list of the compared methods can be found in Table 2. The comparison of a few preceding approaches, such as fixing one camera and scale or fixing three points to fix the gauge of the covariance matrix, can be

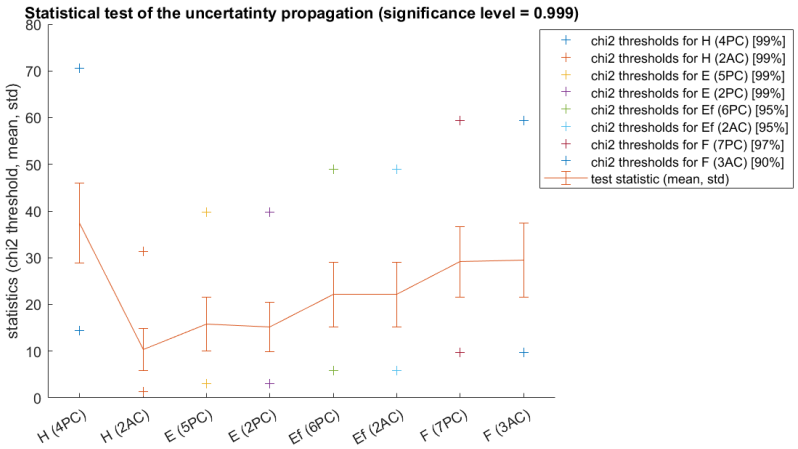


Figure 22: The figure shows the results of the statistical test evaluating the accuracy of covariance propagation methods derived for minimal solvers. The markers represent the bounds of the test statistic λ_t with the significance level $\alpha_{\min} = 0.999$. Each test was evaluated 500 times, resulting in the mean, standard deviation, and percentage of passed tests displayed in brackets. The small percentage of failed tests suggests that the linearization of the constraints is susceptible to numerical errors. For each trial, we assumed $p = 100$ samples and $\sigma_{\text{in}} = 10^{-13}$ variance of input measurements.

found in our papers [**97, 98**].

We begin by describing the dataset used to evaluate the compared methods. We assume the uncertainty propagation method described by Kanatani [75] as the ground truth for our evaluation. This method involves the MP inversion of matrix `M` and has limitations due to the numerical instability, critical even for small reconstructions, and computation speed. We observed that because of this numerical instability, MP inversion often assumes more degrees of freedom than seven. If the number of degrees of freedom is fixed to seven, it can lead to much larger uncertainty in some parameters than expected. This occurs when we apply the SVD decomposition to invert `M`, and the smallest singular values are either assumed as zeros or inverted. As a result, the most uncertain parameters are treated as either the most accurate or much larger than expected. To avoid these problems, we calculated the ground truth using Maple with 100 significant digits. We verified this result using a unit test, which showed that the covariance matrices calculated with half the number of significant digits led to similar results up to the rounding error. However, this calculation is computationally demanding and can only be used for small reconstructions, such as those with $L < 70$, $M < 250$, and $N < 6000$. We estimated the ground truth covariance matrices for five small

#	Algorithm
1	$\mathtt{M}^+$ using Maple (K. Kanatani [75])
2	$\mathtt{M}^+$ using Ceres (K. Kanatani [75])
3	$\mathtt{M}^+$ using Matlab (K. Kanatani [75])
4	$\mathtt{Z}^+$ with correction term (M. Lhuillier [58])
5	TE inversion of $\mathtt{Z}$ (M. Polic [**97**])
6	Nullspace bounding of $\mathtt{M}$ (M. Polic [**99**])

Table 2: Uncertainty propagation methods compared in this section. Method (1) is evaluated in Maple using 100 significant digits. Method (1) is only evaluated on datasets (1)-(4) and is assumed to be the ground truth.

#	Dataset name	L	M	N
1	Cube	6	15	60
2	Toy	10	60	200
3	Flat	30	100	1k
4	Daliborka	64	200	5.2k

Table 3: Summary of datasets with known ground truth. Datasets (1), (3) are synthetic, (2) is reconstructed by COLMAP, and (4) is reconstructed by Bundler.

#	Dataset name	L	M	N
5	Marianska	118	81k	249k
6	Dolnoslaskie	360	530k	226k
7	Tower of London	530	66k	509k
8	Notre Dame	715	127k	748k
9	Seychelles	1.4k	407k	2.1M

Table 4: Summary of the datasets without known ground truth. Dataset (9) is reconstructed by COLMAP and (5)-(8) by Bundler.

scenes, including two synthetic scenes (Cube and Flat), and three datasets reconstructed using publicly available pipelines (COLMAP [8] and Bundler [13]) using a limited number of cameras and reduced number of registered observations and 3D points. Two real datasets were used for evaluation, and two were used for visualizing the uncertainties. The rest of the listed datasets were built using publicly available pipelines without any restrictions on the number of cameras, points, or observations registered. See Tab.3 and4 for more details.

The covariance matrices contain a large range of values because of different units of individual parameters. For example, the mean-variance of focal length is 2×10^3 while the mean-variance of all Euler vector variables is 8×10^{-3} for ground truth datasets. To compare the algorithms concerning the ground truth, we need to specify a suitable metric. To simplify our metric, we compare the camera

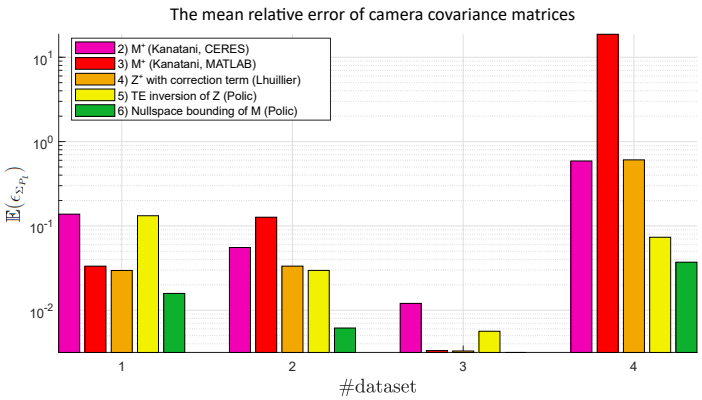


Figure 23: The mean error $\mathbb{E}(\epsilon_{\Sigma_{P_l}})$ for algorithms (2)-(6) from Tab. 2 using the datasets with known ground truth covariance matrices, Tab. 3.

parameters only, i.e., calculate the magnitude of individual parameters

$$\mathtt{Q}_{\mathrm{PP}} = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\mathtt{P}_l\mathtt{P}_l^\top} \tag{168}$$

and use it to normalize the differences between covariance matrices of individual cameras, i.e.

$$\epsilon_{\Sigma_{P_l}} = \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8}\left(\frac{\left(\sqrt{\mathrm{abs}(\widetilde{\Sigma}_{\hat{P}_l\hat{P}_l} - \widehat{\Sigma}_{\hat{P}_l\hat{P}_l})}\right)_{i,j}}{\left(\mathtt{Q}_{\mathrm{PP}}\right)_{i,j}}\right). \tag{169}$$

The $\widetilde{\Sigma}_{\hat{P}_l\hat{P}_l}$ realize the ground truth covariance matrix, and $\widehat{\Sigma}_{\hat{P}_l\hat{P}_l}$ the estimated one. The comparison of mean error $\epsilon_{\Sigma_{P_l}}$ of cameras $l = 1, \ldots, L$ in the datasets (1)-(4) is in the Fig. 23.

Another important metric to consider is the speed of the algorithm. We performed the experiments on a single computer with a 2.6GHz Intel Core i7-6700HQ and 32GB of RAM. Some of the implementations are not practical due to their long run-time or large memory requirements. For example, calculating the ground truth covariance matrix for a small dataset (4) took approximately 22 hours. The same algorithm implemented in Ceres [47] (using 15 significant digits and the Eigen library [144]) took 25.9 minutes, while the Matlab implementation (using 15 significant digits and LAPACK library [145]) took 0.45 seconds. This example shows that the ground truth cannot be found for datasets (5)-(9). Additionally, algorithms (2) and (3) cannot be used on the same datasets due to the memory

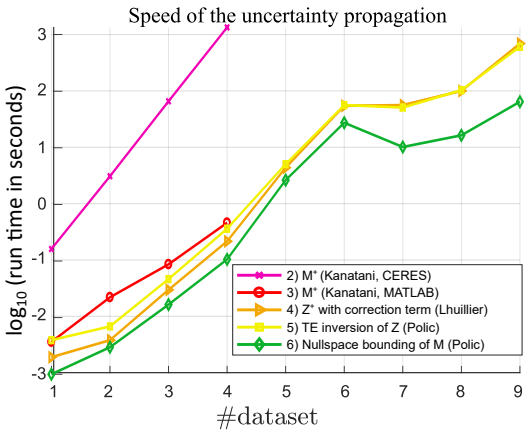


Figure 24: Comparison of the run time of individual algorithms on synthetic and real datasets. Algorithms (2) and (3) were not evaluated on datasets (5)-(9) because of memory requirements.

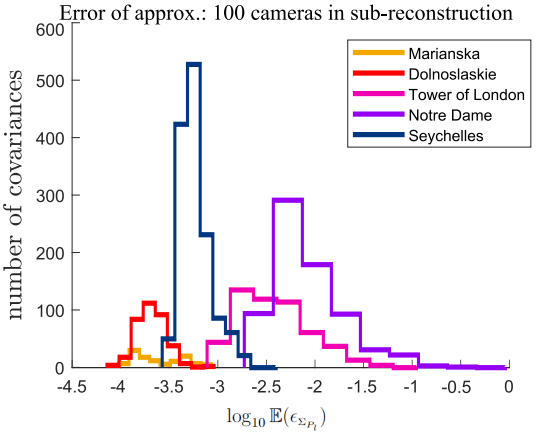


Figure 25: The relative error of camera covariance matrices estimated on datasets (5)-(9) using one hundred neighboring views for the repeatedly randomly chosen sub-reconstruction.

requirements for storing $\mathtt{M}^+$. For instance, dataset (5) would require 470GB to calculate the covariance matrix for $\Sigma_{\theta\theta}$. The comparison of the run-time of individual algorithms is shown in Fig. 24.

In the theoretical section, we showed that extending a sub-reconstruction decreases the uncertainty. Therefore, the uncertainty of a sub-reconstruction is also an upper bound of the reconstruction uncertainty. Figure 25 shows the relative distance between the covariance matrix calculated from all the cameras and that calculated from 100 neighbouring cameras only. We used algorithm (6) and a randomly chosen camera for the propagation. The figure indicates that the relative error of the camera covariance matrix is small even when using a small set of neighbouring views to estimate it. The decreasing trend of the relative and absolute error of the estimated covariance matrices with increasing size of the sub-reconstruction, i.e., assuming $5, 10, 20, 40, 80, 160, 320$ cameras, is shown in Figures 26 and 27.

To gain better insight into the behavior of the different methods, we conducted a final experiment to visualize the estimated and ground truth covariance matrices of camera centers as standard ellipses. As an example, we used the Buddha dataset consisting of 67 images, 1553 points, and 4263 image observations. The number of cameras and points was reduced to 67 and 8, respectively, to allow for the calculation of ground truth covariance matrices. Two example images from the Buddha dataset are shown in Fig. 28. To visualize the relative errors in camera

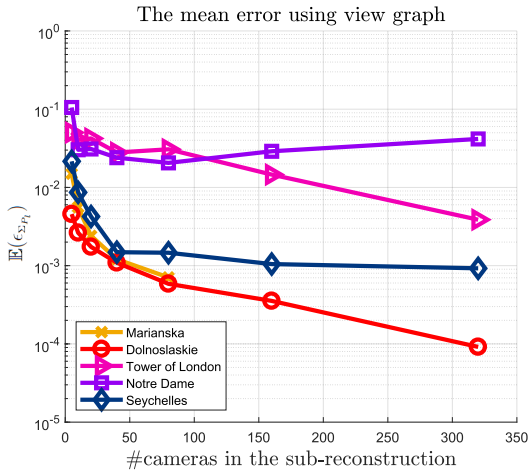


Figure 26: Mean of the relative error $\log_{10}\mathbb{E}(\epsilon_{\Sigma_{P_l}})$ for increasing the size of the sub-reconstruction utilized to estimate single camera uncertainty matrix.

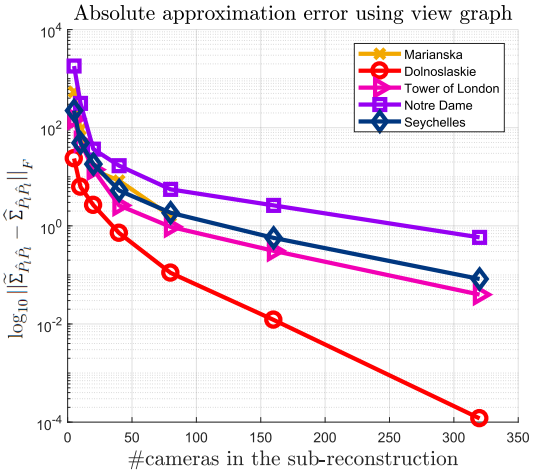


Figure 27: The median of the absolute error $\log_{10}||\widetilde{\Sigma}_{\hat{P}_l\hat{P}_l} - \widehat{\Sigma}_{\hat{P}_l\hat{P}_l}||_F$ for increasing size of the sub-reconstruction utilized to estimate single camera uncertainty matrix.

covariance, we generated a histogram with corresponding colors (see Fig. 29). The main visualization is shown in Fig. 30 and 31, which utilize the color coding of the errors from the histogram 29. The standard ellipses indicate the standard deviation of the camera center position. Additional visualizations of datasets (1)-(4) and the Cereal dataset can be found in [**98, 99**].

(a) 3D reconstruction (b) One of input images

Figure 28: The Buddha dataset [146] contains 67 images, 1553 points and 4263 image observations.

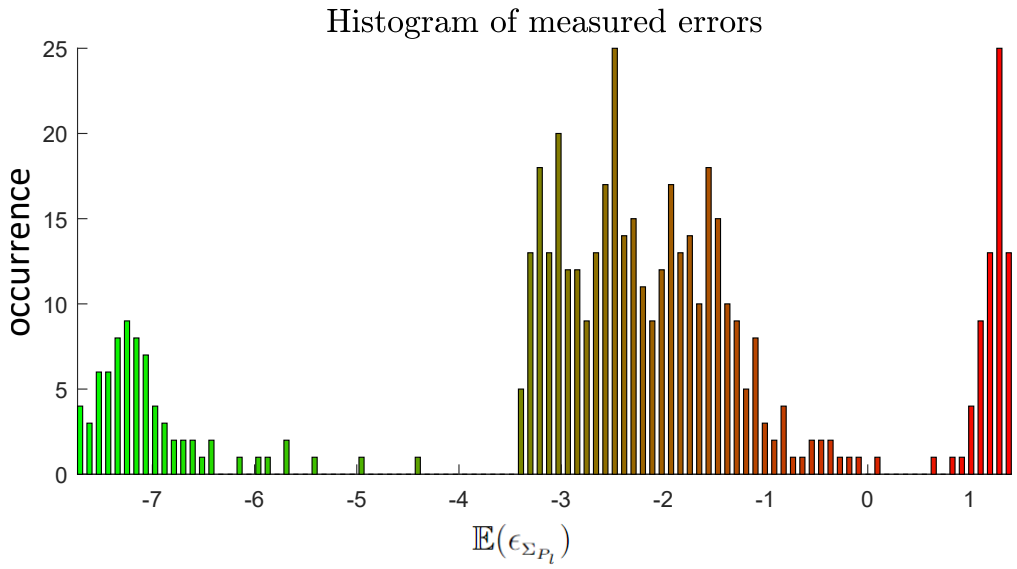
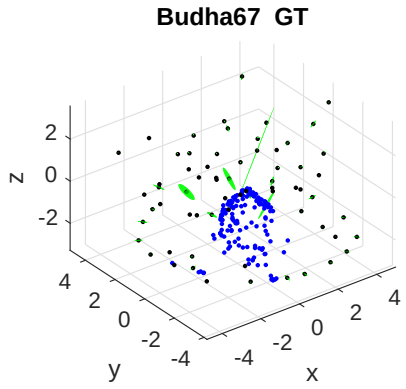
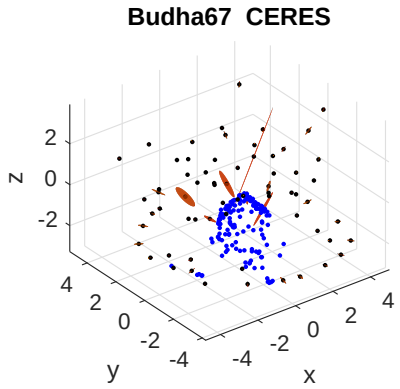


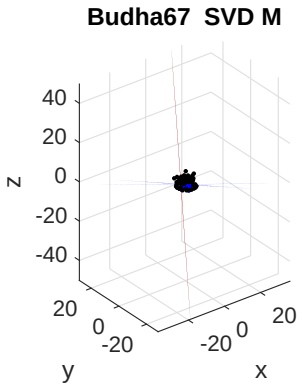
Figure 29: The distribution of relative camera covariance matrices errors with corresponding color coding utilized in the visualization of standard ellipsoids, in Fig. 30, and 31.



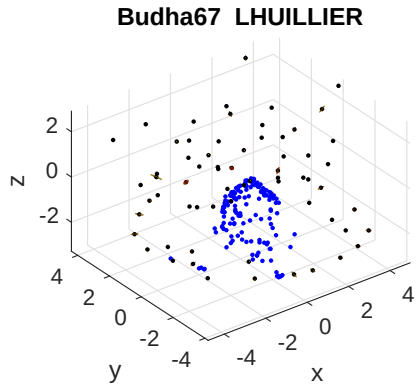
(a) GT covariance ellipsoids (alg. (1))



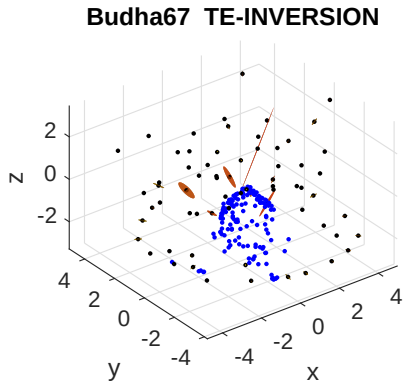
(b) 2) $\mathtt{M}^{+}$ (Kanatani, CERES)



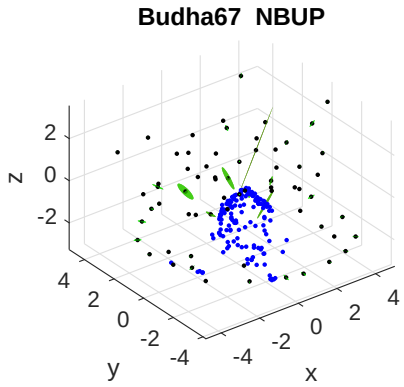
(c) 3) $\mathtt{M}^{+}$ (Kanatani, Matlab)



(d) 4) $\mathtt{Z}^{+}$ with correction term (Lhuillier)

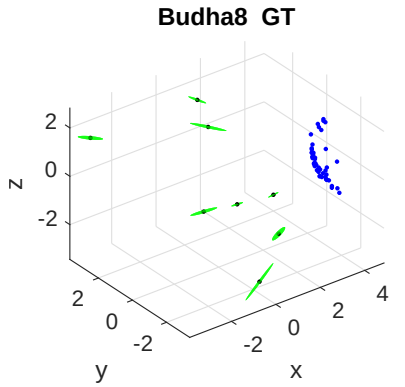


(e) 5) TE inversion of $\mathtt{Z}$ (Polic)

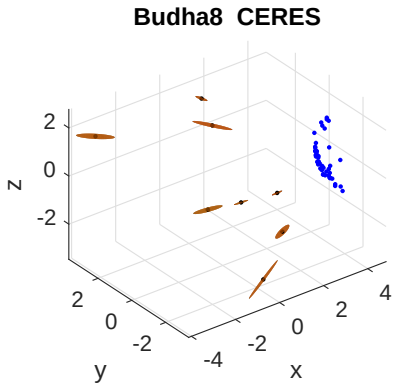


(f) 6) Nullspace bounding of $\mathtt{M}$ (Polic)

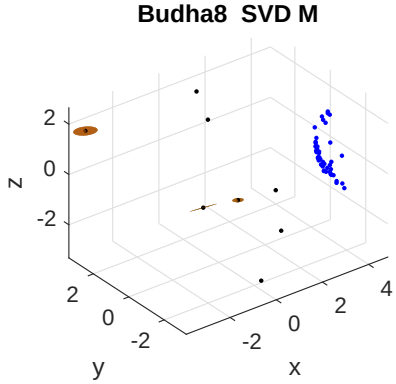
Figure 30: The visualization of the camera centers uncertainty on sub-reconstruction with all 67 cameras and 150 points in 3D for the Buddha dataset. Each ellipsoid shows the most unconstrained directions of the camera position and green to red color mapping the covariance matrix error. The blue dots realize the points in 3D.



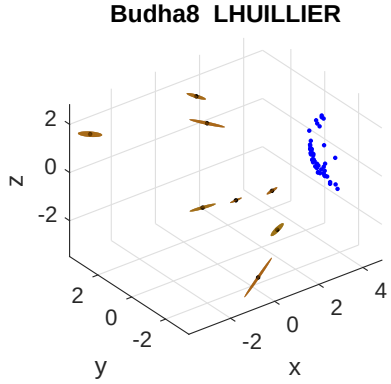
(a) GT covariance ellipsoids (alg. (1))



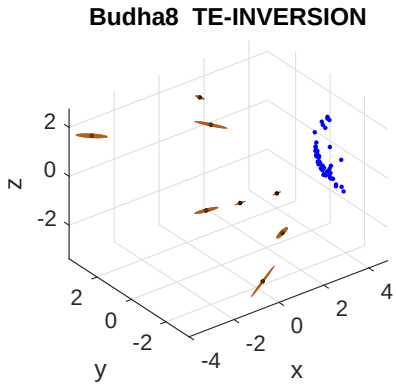
(b) 2) $\mathtt{M}^{+}$ (Kanatani, CERES)



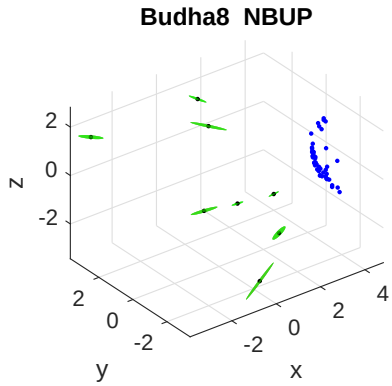
(c) 3) $\mathtt{M}^{+}$ (Kanatani, Matlab)



(d) 4) $\mathtt{Z}^{+}$ with correction term (Lhuillier)



(e) 5) TE inversion of $\mathtt{Z}$ (Polic)



(f) 6) Nullspace bounding of $\mathtt{M}$ (Polic)

Figure 31: The visualization of the camera centers uncertainty on sub-reconstruction with 8 cameras and 50 points in 3D for the Buddha dataset. Each ellipsoid shows the most unconstrained directions of the camera position and green to red color mapping the covariance matrix error. The blue dots realize the points in 3D.

9 Applications of the uncertainty modelling

The first step required for uncertainty propagation is always the estimation of model parameters, such as the essential or fundamental matrix. Without estimated model parameters, we do not have any linearization point for expressing the derivative of geometric constraints. This is a significant limitation of uncertainty applications in practice. However, we can use uncertainty to avoid subsequent extensive processing and improve the accuracy of the model estimate. For example, we may skip the verification of the fundamental matrix on all correspondences. Another example is skipping the reconstruction optimization when using an unsuitable mathematical model of the projection function. Using a suitable camera model can lead to an order of magnitude faster convergence of the bundle adjustment. This section shows how to use the information about the uncertainty of estimated parameters to make SfM fast, accurate, and robust.

9.1 Uncertainty-based robust model estimator

The uncertainty of the inputs can be naturally utilized by employing Maximum Likelihood (ML) estimation, which minimizes the influence of input measurement noise on the estimated parameters. A single iteration of ML estimation can reduce the error of algebraic solutions below 10%-40% of the parameters' standard deviations [147, 148]. We show that the output covariance matrix can be used to filter out very uncertain or too accurate algebraic solutions. The uncertainty of correctly estimated parameters follow a reference distribution, where too uncertain solutions are not usable in practice. At the same time, too accurate ones usually correspond to perfectly fitted outliers (the solution fits only the data used at the input of a minimal solver). The magnitude of the uncertainty can be described by several statistics. The most accurate one is the condition number, which expresses the squared ratio of the most and least uncertain axis of the standard ellipse. A weaker, much more computationally efficient statistic is $\text{tr}(\Sigma_{x_i x_i})$, which expresses the average variance of the i-th estimated model, expressed by vector $\boldsymbol{x}_i$. This can still identify very uncertain or too accurate solutions and can be employed as the initialization of the probability of having the correct model in the preemptive model verification by the Sequential Probability Ratio Test (SPRT) [104, 149].

9.2 Camera model selection

The utilization of reconstruction uncertainty for camera model selection is presented in **[15]**. SfM pipelines have numerous hyperparameters that are set by users, often without detailed knowledge of the underlying algorithms and their dependence on input data. As all minimal solvers and bundle adjustments assume

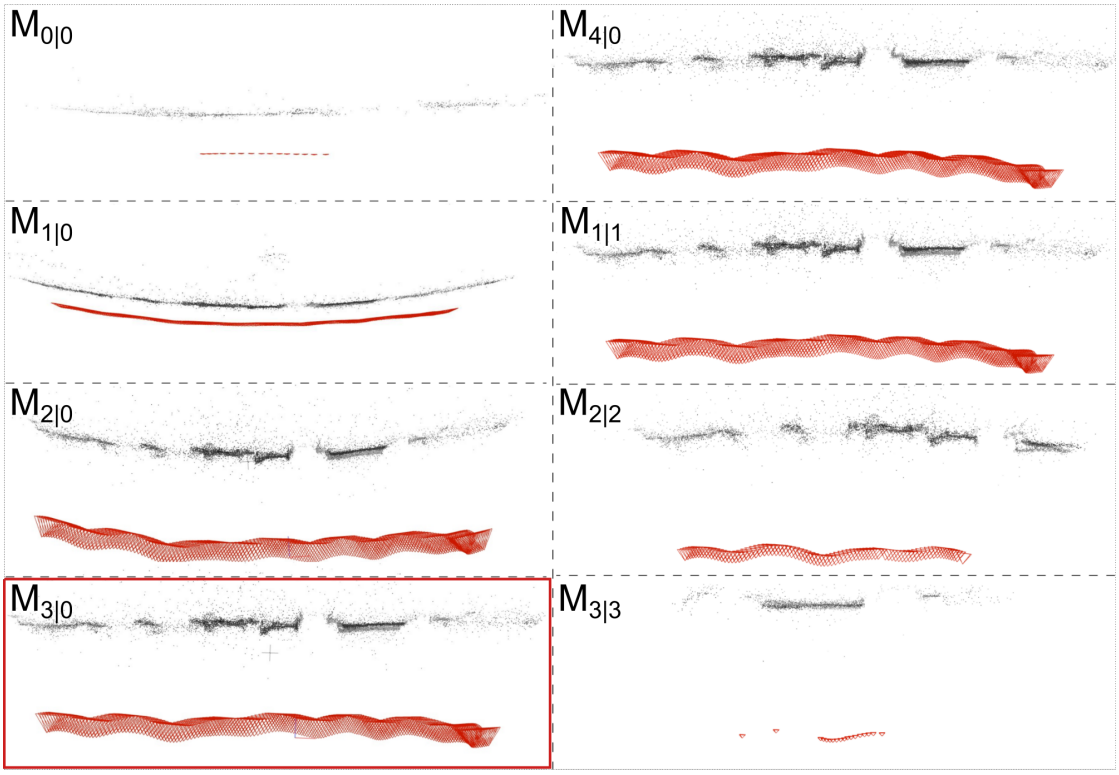


Figure 32: COLMAP reconstructed cameras (red) and 3D points (black) using eight different radial distortion models. The best camera model selected using the LACS method (framed in red) gives the most planar result for 3D points of a flat wall (using the terrains rig dataset from [151]).

a single set of geometrical constraints, i.e., one camera model, it is crucial to select the appropriate one. A model that is too simple may filter out inliers, while a model that is too complex may lead to over-fitting and result in degeneracies as shown in Albl [150]. An example of the reconstruction of a flat wall using different camera models is shown in Fig. 32. The ultimate goal of camera model selection is to choose a model that registers all cameras, has the largest number of inliers, and has the smallest reprojection error. However, as this is not feasible in practice, we need a criterion that expresses how well a reconstruction (with a selected camera model) fits the observed feature points. To avoid calculating a large-scale reconstruction using all camera models, we propose a method for choosing a camera model for a small subset of input images. Specifically, we suggest reconstructing a small subset of images using all camera models and selecting the most suitable model for large-scale reconstruction.

Notation for camera models and projection functions is described in Sec. 5.5,

and the basics of Information Criterion (IC) are introduced in Sec. 5.4 and Sec. 3.3. However, existing ICs do not perform well on this task. Therefore, we propose a new Accuracy-based Criterion (AC) that selects the model leading to the most accurate reconstruction. Note that we use the abbreviation "AC" for Accuracy-based Criterion instead of "Affine Correspondences" in this section, to stay consistent with both previous publications [**15, 17**].

The idea behind Accuracy-based Camera Selection (ACS) approach is that a well-fitting camera model will result in as many feature points as possible, with each observation contributing to the Fisher information matrix and decreasing the covariance matrix of estimated parameters. In an over-parameterized case, the number of feature points does not increase significantly as more camera parameters are employed, leading to larger uncertainty of the common parameters for all sub-reconstructions (assuming different camera models to calculate the reconstructions) when proper selection of the gauge of the covariance matrix is utilized.

9.2.1 Accuracy-based criterion (AC)

We assume that there are n estimated reconstructions $\hat{\boldsymbol{\theta}}^{(i)}$ from L images, calculated using different camera models from the set $\mathcal{M} = \mathbf{M}_1, \ldots, \mathbf{M}_n$, as input for the ACS. The AC (part of ACS) estimates the "goodness" of the fit for one camera model $\mathbf{M}_i$ in a way that is comparable with the other camera models in $\mathcal{M}$. Let us redefine the parameters of the reconstruction (e.g., $\boldsymbol{\theta}^{(i)}$, $\boldsymbol{P}^{(i)}$, $\boldsymbol{X}^{(i)}$, etc.) to correspond to the reconstruction from the subset of cameras directly. This simplifies the following notation, and if required, we call the whole reconstruction with a tilde (e.g., $\widetilde{\boldsymbol{\theta}}^{(i)}$, $\widetilde{\boldsymbol{P}}^{(i)}$, $\widetilde{\boldsymbol{X}}^{(i)}$, etc.). Using the sub-reconstruction $\boldsymbol{\theta}^{(i)}$ as the linearization point, the accuracy of observations $\mathtt{W}_{uu} = \Sigma_{uu}^{-1}$ can be propagated into the reconstruction parameters according to Sec. 8.2. In practice, each reconstruction $\boldsymbol{\theta}^{(i)}$ is in a different coordinate system with a different gauge of the covariance matrix. To obtain comparable covariance matrices, we need to:

1. Fix the gauge of the coordinate systems (K-transformation in Sec. 5.3).
2. Fix the gauge of the information matrix (S-transformation in Sec. 5.3).
3. Define the metric (i.e., AC) that expresses how well $\boldsymbol{\theta}^{(i)}$ (i.e., $\mathbf{M}_i$) fits the detected feature points.

One of the challenges for IC is a different number of reconstruction parameters $K^{(i)}$ for different reprojection error thresholds δ, different camera model $\mathbf{M}_i$, and different parameters setup. In the case of AC, we select a common part of the reconstruction further used to compare its uncertainty. Let us denote the common part of the reconstruction $\boldsymbol{\theta}_A$ and the remaining parameters $\boldsymbol{\theta}_B^{(i)} = \{\boldsymbol{\theta}^{(i)} \setminus \boldsymbol{\theta}_A\}$.

Note that we assume that these parts are not overlapping and form the reconstruction, i.e., $\boldsymbol{\theta}^{(i)} = \{\boldsymbol{\theta}_A^{(i)}, \boldsymbol{\theta}_B^{(i)}\}$. To fix the gauge of the coordinate system, we chose one reference reconstruction $\boldsymbol{\theta}^{(r)}$, i.e., one reference camera model $\boldsymbol{M}_r \in \mathcal{M}$. Next, all the reconstructions are transformed (step number (1)) to this reference coordinate system such that the similarity transformation minimizes the distances of the camera centers and angles between the optical axes w.r.t. the camera poses in the reference reconstruction. To simplify the following text, let us assume that all reconstructions $\boldsymbol{\theta}^{(i)}$, $\forall i \in \{1, \ldots, n\}$ are already aligned to the reference coordinate system. Step number (2) is to fix the gauge of the coordinate system following eq. (26). The difference is that we assume two sets of parameters, i.e., $\boldsymbol{\theta}^{(i)} = \{\boldsymbol{\theta}_A^{(i)}, \boldsymbol{\theta}_B^{(i)}\}$. Therefore the Jacobian is also composed of $\mathtt{J}_A^{(i)}$, $\mathtt{J}_B^{(i)}$ where $\mathtt{J}_A^{(i)}$ denotes the Jacobian of $\boldsymbol{p}^{(i)}$ w.r.t. $\boldsymbol{\theta}_A$ and $\mathtt{J}_B^{(i)}$ denotes the Jacobian of $\boldsymbol{p}^{(i)}$ w.r.t. $\boldsymbol{\theta}_B$. Then, we can write the propagation as

$$\mathtt{W}_{\theta\theta}^{(i)} = \begin{bmatrix} \mathtt{W}_{AA}^{(i)} & \left(\mathtt{W}_{AB}^{(i)}\right)^{\mathsf{T}} \\ \mathtt{W}_{AB}^{(i)} & \mathtt{W}_{BB}^{(i)} \end{bmatrix} = \begin{bmatrix} \left(\mathtt{J}_A^{(i)}\right)^{\mathsf{T}} \\ \left(\mathtt{J}_B^{(i)}\right)^{\mathsf{T}} \end{bmatrix} \mathtt{W}_{uu} \begin{bmatrix} \mathtt{J}_A^{(i)} & \mathtt{J}_B^{(i)} \end{bmatrix}, \tag{170}$$

where $\mathtt{W}_{\theta\theta}^{(i)}$ is a symmetric positive semi-definite matrix with 7 degrees of freedom, and $\mathtt{W}_{AA}^{(i)}, \mathtt{W}_{BB}^{(i)}, \mathtt{W}_{AB}^{(i)}$ are blocks of $\mathtt{W}_{\theta\theta}^{(i)}$ corresponding to $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B^{(i)}$. Note that $\boldsymbol{\theta}_A$ is independent of $\mathbf{M}_i$ while $\boldsymbol{\theta}_B^{(i)}$ consist of the remaining parameters which size may be different. To have a comparable information matrix (i.e., we do not have to calculate the MP inversion leading to numerical instabilities), we define such an S-transformation matrix $\mathtt{S}^{(i)}$ that ensures that common parameters $\boldsymbol{\theta}_A$ are independent of $\boldsymbol{\theta}_B^{(i)}$, i.e.

$$\begin{bmatrix} \mathtt{W}_A^{(i)} & 0 \\ 0 & \mathtt{W}_B^{(i)} \end{bmatrix} = \mathtt{S}^{(i)} \begin{bmatrix} \mathtt{W}_{AA}^{(i)} & \left(\mathtt{W}_{AB}^{(i)}\right)^{\mathsf{T}} \\ \mathtt{W}_{AB}^{(i)} & \mathtt{W}_{BB}^{(i)} \end{bmatrix} \left(\mathtt{S}^{(i)}\right)^{\mathsf{T}}, \tag{171}$$

where the matrix $\mathtt{S}^{(i)}$ can be written as

$$\mathtt{S}^{(i)} = \begin{bmatrix} \mathtt{E}_B & -\left(\mathtt{W}_{AB}^{(i)}\right)^{\mathsf{T}} \left(\mathtt{W}_{BB}^{(i)}\right)^{-1} \\ \mathtt{0}_A & \mathtt{E}_B \end{bmatrix}. \tag{172}$$

The submatrix $\mathtt{W}_A^{(i)}$ of the information matrix has the same dimension $K_A = \dim(\mathtt{W}_A^{(i)})$ $\forall i \in \{1, \ldots, n\}$, i.e., for all camera models $\boldsymbol{M_i}$. The matrix $\mathtt{E}_B$ is unit matrix of size $\dim(\boldsymbol{\theta}_B^{(i)})$, and $\mathtt{0}_A$ is zero matrix of size $\dim(\boldsymbol{\theta}_A)$. Moreover, this block can be expressed from the Schur complement of a block matrix

$$\mathtt{W}_A^{(i)} = \mathtt{W}_{AA}^{(i)} - \left(\mathtt{W}_{AB}^{(i)}\right)^{\mathsf{T}} \left(\mathtt{W}_{BB}^{(i)}\right)^{-1} \mathtt{W}_{AB}^{(i)}, \tag{173}$$

and express the accuracy of common parameters $\boldsymbol{\theta}_A$ with fixed gauge of the information matrix. The MP inversion of $\mathtt{W}_A^{(i)}$ realize the covariance matrix

$$\Sigma_{\theta_A\theta_A}^{(i)} = \left(\mathtt{W}_A^{(i)}\right)^+ \tag{174}$$

describing the uncertainty of common parameters. For example, the largest eigenvalue $\lambda_{\max}(\Sigma_{\theta_A\theta_A}^{(i)})$ equal the squared magnitude of the main diagonal of the standard ellipse, i.e., the magnitude of the most uncertain parameter. As the computation of MP inversion is computationally demanding, we rather analyze the eigenvalues of the information matrix $\mathtt{W}_A^{(i)}$. Suppose we assume the eigenvalues $\lambda_A = \mathrm{eig}(\mathtt{W}_A^{(i)})$ in the ascending order. In that case, the first seven eigenvalues will equal zeros as so as the seven smallest eigenvalues of the related covariance matrix. The other eigenvalues are the inversion of the eigenvalues of the covariance matrix, i.e.

$$\left(\mathrm{eig}(\Sigma_{\theta_A\theta_A}^{(i)})\right)_{(K_A^{(i)}-j)} = \frac{1}{\lambda_{A,(8+j)}} \qquad \forall j \in \{0,\ldots,K_A^{(i)}-8\}\,. \tag{175}$$

Therefore, the variance of the most uncertain parameter is realized $1/\lambda_{A,8}$, and we can use it as the IC. We empirically found that a better indicator of the scene accuracy is the trace

$$\mathrm{AC} = \mathrm{tr}(\mathtt{W}_A^{(i)}) \tag{176}$$

corresponding to the sum of eigenvalues. As the eigenvalues of the information matrix are inverted variances, their sum is large if all the common parameters are accurate. Moreover, the trace can be calculated efficiently without the eigenvalue decomposition.

9.2.2 Camera model selection method (ACS)

Accuracy-based Camera model Selection (ACS) selects the camera model that yields the highest score according to the AC criterion. We describe the observations by their covariance matrix Σ_{uu}, select a subset of $5 \leq L \leq 15$ cameras, and calculate the reconstruction parameters $\boldsymbol{\theta}^{(i)}$ for each camera model $\mathbf{M}_i$. The number of cameras L required for a reliable estimate of the camera model was found empirically. Increasing the number of cameras improves the estimated 3D model's accuracy and computational time. Our approach is to run all the reconstructions in parallel for all the camera models and wait until some register L cameras. This time is denoted as T_1. If the camera model fits the observations well, the slowest part of the reconstruction process, the bundle adjustment, is much faster than for a wrong camera model. Moreover, the cameras are registered in approximately the same order for suitable camera models. Therefore, our heuristic is to start all the reconstructions in parallel and stop all the SfM instances that are not able to

register L cameras in $T_d = \gamma T_1$. Here, γ is an empirically set factor that determines how many times longer to wait before stopping the SfM executions. All the sub-reconstructions that contain L cameras are used to find the common set of parameters. We focused on the camera parameters only, as selecting common 3D points is computationally demanding. The camera parameters common to all the sub-reconstructions are denoted by $\boldsymbol{\theta}_A \subseteq \boldsymbol{\theta}^{(i)}, \forall i$. The ACS method is summarized in Algorithm 1.

9.2.3 Learned threshold (LACS)

The ACS selects the most accurate reconstruction and related camera model for a fixed reprojection threshold δ. Therefore, different reprojection thresholds may lead to different camera models being selected. However, the reprojection error threshold can be easily adjusted. For instance, assuming $\delta_{\max} = 2$ px, we can filter out correspondences with reprojection error larger than 0.5, 1, 1.5 px and update the reconstructions by bundle adjustment. Let us assume a number of thresholds n_{thr}. As the AC is fast to calculate, we obtain the AC for each camera model (up to n) and each reprojection threshold, i.e., a matrix $\mathbb{R}^{n \times n_{\text{thr}}}$ where each unknown AC criterion is replaced by zero. These values are further sent to a shallow neural network consisting of four hidden fully connected layers (with dimensions: $d_0 = n \times n_{\text{thr}}$, $d_{1,2} = (n \times n_{\text{thr}})/2$, $d_{3,4} = n$), each followed by leaky ReLU activation [152]. This neural network is trained on synthetically created projections with additional positional noise using a known camera model and real scenes. The process is described in the empirical evaluation. The LACS benefits from multiple reprojection thresholds and leads to a superior estimate of the camera model for input images.

Input: A finite set of images $\mathcal{I} = \{\mathtt{I}_1, \mathtt{I}_2, \ldots, \mathtt{I}_{\widetilde{L}}\}$; reprojection threshold δ; the number of registered cameras in the sub-reconstruction L; time factor γ; a finite set of camera models $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$

Output: the selected camera model $\mathbf{M}_b$; calibration parameters for the output camera model $\hat{\boldsymbol{\theta}}_b$

$T_d \leftarrow \infty$, $\mathcal{S}_{\text{sub}} \leftarrow \emptyset$, $\mathcal{S}_{\text{AC}} \leftarrow \emptyset$

`// run in parallel until` T_d `elapses`

for $i \leftarrow 1$ ***to*** n **do**

  $[\hat{\boldsymbol{\theta}}^{(i)}, T_1] \leftarrow \text{SfM}(\mathcal{I}, \mathbf{M}_i, \delta, L)$

  if $T_d = \infty$ **then**

    $T_d \leftarrow \gamma\, T_1$

  end

  $\mathcal{S}_{\text{sub}} \leftarrow \{\mathcal{S}_{\text{sub}}, \hat{\boldsymbol{\theta}}^{(i)}\}$

end

`// finished sub-reconstructions` $\mathcal{S}_{\text{sub}}$

$\boldsymbol{S_A} \leftarrow$ find_common_parameters($\mathcal{S}_{\text{sub}}$)

for $\hat{\boldsymbol{\theta}}^{(i)} \in \mathcal{S}_{sub}$ **do**

  $\bar{\boldsymbol{\theta}} \leftarrow$align_coordinates($\hat{\boldsymbol{\theta}}^{(i)}$, $\mathcal{S}_{\text{sub},1}$)

  $[\boldsymbol{\theta}_A, \boldsymbol{\theta}_B] \leftarrow$ split_parameters($\bar{\boldsymbol{\theta}}$, $\boldsymbol{S_A}$)

  $[\mathtt{J}_A, \mathtt{J}_B] \leftarrow$ get_derivatives($\mathbf{M}_i$, $\boldsymbol{\theta}_A$, $\boldsymbol{\theta}_B$)

  $[\mathtt{W}_{AA}, \mathtt{W}_{AB}, \mathtt{W}_{BB}] \leftarrow$ get_inform_mat($[\mathtt{J}_A, \mathtt{J}_B]$)

  $\mathtt{W}_A^{(i)} \leftarrow$ get_schur_complement($[\mathtt{W}_{AA}, \mathtt{W}_{AB}, \mathtt{W}_{BB}]$)

  $\mathcal{S}_{\text{AC}} \leftarrow \{\mathcal{S}_{\text{AC}}, \text{tr}(\mathtt{W}_A^{(i)})\}$

end

$\mathbf{M}_b \leftarrow$ select_model($\mathcal{S}_{\text{AC}}$, $\mathcal{M}$)

Algorithm 1: The ACS method runs the SfM algorithm in parallel for each camera model $\mathbf{M}_i$ until the time limit T_d is exceeded. The set $\mathcal{S}_{\text{sub}}$ contains all the reconstructions that successfully registered L images. The function align_coordinates($\hat{\boldsymbol{\theta}}^{(i)}$, $\mathcal{S}_{\text{sub},1}$) fix the gauge of the coordinate system by aligning the camera poses of $\hat{\boldsymbol{\theta}}^{(i)}$ cameras to the cameras of the $\mathcal{S}_{\text{sub},1}$ reconstruction. The function split_parameters($\bar{\boldsymbol{\theta}}$, $\boldsymbol{S_A}$) split $\hat{\boldsymbol{\theta}}^{(i)}$ into $\boldsymbol{\theta}^{(i)} = \{\boldsymbol{\theta}_A^{(i)}, \boldsymbol{\theta}_B^{(i)}\}$ where $\boldsymbol{\theta}_A^{(i)}$ realize the set of common camera parameters for all the reconstructions in $\mathcal{S}_{\text{sub}}$. The function get_derivatives($\mathbf{M}_i$,$\hat{\boldsymbol{\theta}}_A$,$\hat{\boldsymbol{\theta}}_B$) computes partial derivatives for a given model $\mathbf{M}_i$. Next, the function get_inform_mat($[\mathtt{J}_A, \mathtt{J}_B]$) follow eq. (170). The Schur complement of a block matrix, i.e. the function get_schur_complement($[\mathtt{W}_{AA}, \mathtt{W}_{AB}, \mathtt{W}_{BB}]$) is calculated according eq. (173), and the function select_model($\mathcal{S}_{\text{AC}}$, $\mathcal{M}$) pick up the camera model $\mathbf{M}_i$ with the largest AC.

9.3 Evaluation

This section presents the experimental evaluation of two methods based on uncertainty propagation in the scope of SfM. The first method, uncertainty-based preemptive verification utilize the probability of having a high inlier sample of the minimal problem solution, estimated from the uncertatinty of the solution, to speed up the Sequential Probability Ratio Test (SPRT). Therefore, we demonstrate the speed improvement achieved by employing this method.

The second method is the Accuracy-based Camera Selection (ACS) and its extension, the Learning-based ACS (LACS). The ACS method selects the most accurate camera model for a given set of input images by calculating the reconstruction accuracy using the Accuracy-based Criterion (AC). We test the ACS method on polynomial degree estimation to check if it is competitive with the current state-of-the-art Information Criterion (IC). Then, we generate synthetic reconstructions with known camera models and test ACS and LACS compared to IC. We also present the mean success ratios of the ACS and LACS classifiers with an increasing number of registered cameras. Next, we evaluate the performance of ACS and LACS methods on synthetic sub-reconstructions and plot confusion matrices, i.e., a statistic of selected camera models for the reconstruction with a known ground truth camera model. We also visualize the AC evaluation for iteratively adding new cameras into the partial reconstruction. Finally, the outputs of real datasets, including the number of reconstructed cameras, points, observations, runtime, reprojection error, distance to ground truth (if available), and selected models by LACS, are shown.

In summary, this evaluation demonstrates the effectiveness of uncertainty propagation methods in improving the accuracy and speed of SfM pipelines. We show that the proposed methods can lead to automatic camera model selection and improve the overall quality of the reconstructions.

9.3.1 Uncertainty-based preemptive verification

Experimental results related to uncertainty modeling and utilization in preemptive verification are presented in [**17**]. The main idea is to train the distribution of $\text{tr}(\Sigma_{x_i x_i})$ of estimated models, leading to more than 95% of inliers. In other words, we stored the trace of the covariance matrix $\text{tr}(\Sigma_{x_i x_i})$ and inlier ratio for all the models generated in the RANSAC loop, for all the image pairs in all the datasets assumed in [**17**]. When running the RANSAC, we 1) estimate the model and its uncertainty, 2) calculate the likelihood of having a sample with a high inlier ratio, and 3) initialize the SPRT test by this likelihood of the model to be verified. The empirical results showed that the point solvers follow approximately the exponential distribution while the affine ones follow the

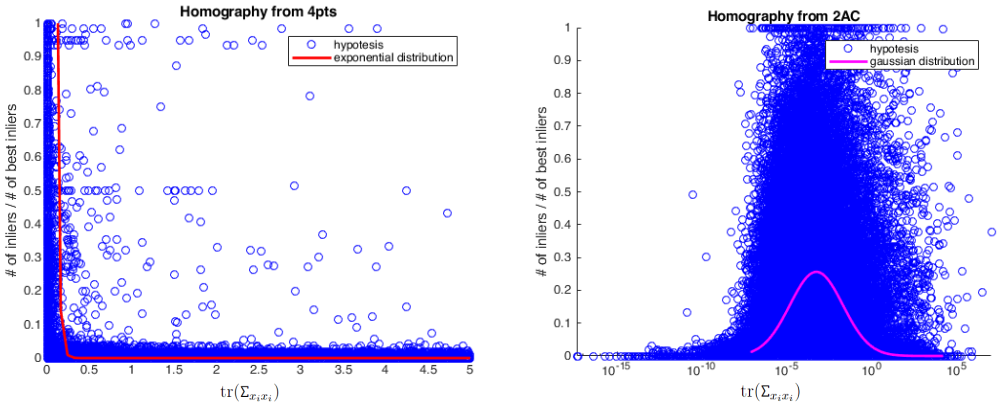
log-normal distribution. The exponential distribution is defined by the rate parameter $\hat{\lambda}_{\text{PC}} = (n_{0.95} - 2)/\sum_{i=1}^{n_{0.95}}(\text{tr}(\Sigma_{x_i x_i}))$, where $n_{0.95}$ denotes the number of models with an inlier ratio > 0.95. The affine correspondences are modeled by a log-normal distribution with the mean $\mu_{\text{AC}} = 1/n_{0.95} \sum_{i=1}^{n_{0.95}} \log_{10}(\text{tr}(\Sigma_{x_i x_i}))$ and variance $\sigma^2_{\text{AC}} = 1/(n_{0.95} - 1) \sum_{i=1}^{n_{0.95}} (\log_{10}(\text{tr}(\Sigma_{x_i x_i})) - \mu_{\text{AC}})^2)$. Fig. 33 shows the fitted distributions for PC and AC inlier ratios stored for homography and fundamental matrix estimation. These apriori-determined distributions can also be updated online during the first few cycles of RANSAC, improving the accuracy of the likelihood estimate of a high inlier ratio sample. Note that the uncertainty propagation for the essential matrix is computationally expensive and, therefore, not beneficial for speeding up the preemptive verification step. Thus, we skipped the uncertainty evaluation for the essential matrix. Further, we simplify the high inlier ratio modeling by assuming that all distributions are log-normal.

We evaluate the performance of proposed methods on both epipolar geometry and homography estimation tasks. The evaluation is conducted on the benchmark, introduced by Bian [153], using the image pairs from TUM [154], KITTI [155], and Tanks and Temples [156] datasets for epipolar geometry estimation, and on the scenes of the HPatches dataset [157] for homography estimation. The RANSAC's inlier-outlier threshold for the epipolar geometry estimation task is set to 1 px (`F`) and 5 px (`H`). Our primary focus is on the execution time of RANSAC for a given reprojection error (we refer the readers to **[17]**, for more details about utilized reprojection error metric). To speed up the RANSAC process, we incorporate the likelihood of having a high inlier ratio, estimated from the apriori determined uncertainty distribution of the models with $> 95\%$ inliers, into the SPRT test [104, 149]. This step avoids expensive validation of models that are likely to be worse than the current best model. Following experiments demonstrate that the cumulative distribution function for homography and fundamental matrix estimation with uncertainty check leads to the fastest robust estimation of both tested AC minimal solvers. We also observe that compared to PCs, the ACs lead to significant speed up when used according to the guideline in **[17]**.

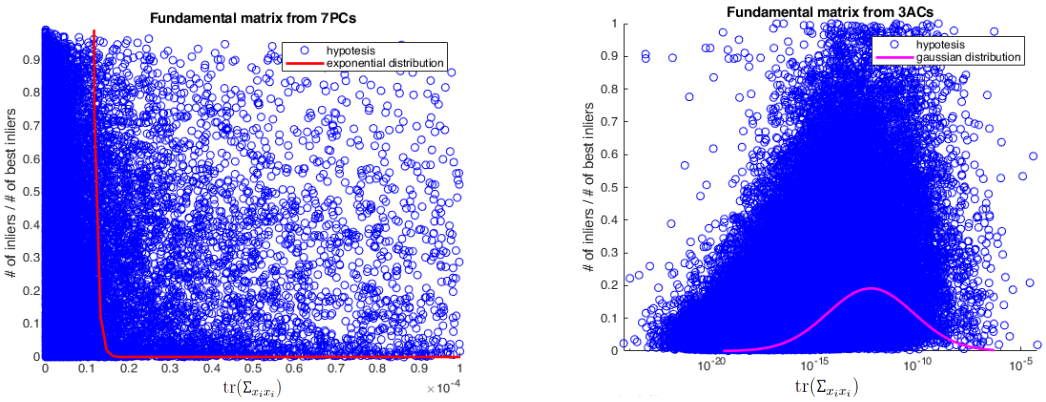
9.3.2 Camera model selection

The experimental evaluation is presented in **[15]**. The experiments can be divided into three parts: 1) comparing the ACS method on the problem of polynomial degree estimation, 2) evaluating standard IC, ACS, and LACS on synthetic reconstructions with known camera model, and 3) evaluation of the ACS and LACS methods on real images without known ground truth camera model.

Polynomial degree estimation. We first assess whether the ACS method produces competitive results compared to the well-known standard ICs on the problem



(a) Homography estimation

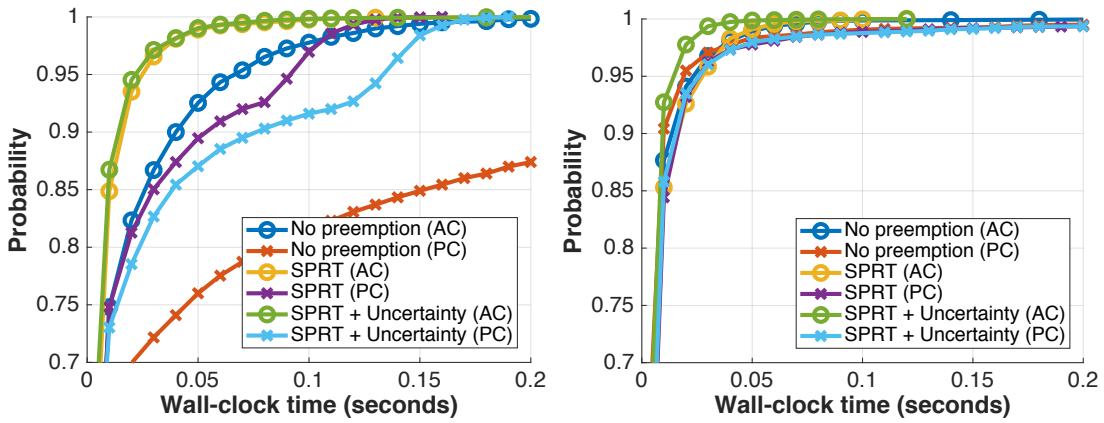


(b) Fundamental matrix estimation

Figure 33: The inlier ratio for all estimated models `H`, `F` from all RANSAC loops of all image pairs of all tested datasets as a function of the trace $\mathrm{tr}(\boldsymbol{\Sigma}_{\boldsymbol{x}_i \boldsymbol{x}_i})$, where $\boldsymbol{x}_i$ realize the i-th model parameters. This figure shows that some estimates are less likely to lead to a high inlier ratio.

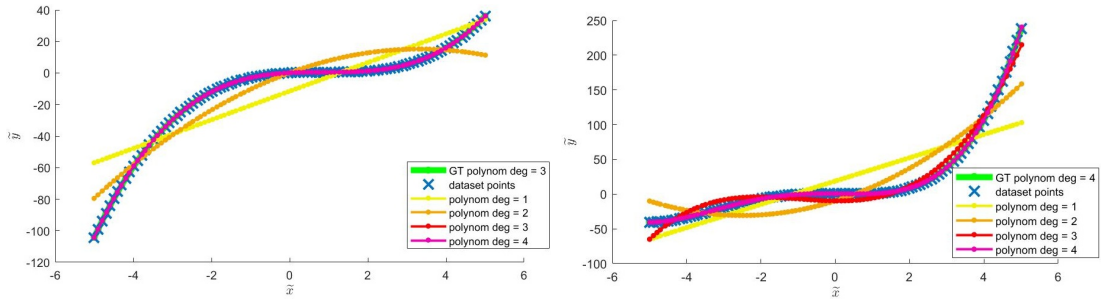
of polynomial degree estimation. We compare 13 standard ICs as summarized in Table 1 in [**15**]. Note that the standard ICs comprise the goodness of fit, realized by the log-likelihood L_{IC}, of having k_{IC} parameters and the bias correction term. The log-likelihood L_{IC} can be further decomposed into $L_{\mathrm{IC}} = T_{\mathrm{IC}} - R_{\mathrm{IC}}$, where the sum of squared weighted residuals is realized by R_{IC}, and the constant term T_{IC}, depending on the number of observations N_{IC}, is suppressed [54, 105–113, 158]. In our case, T_{IC} varies for camera models and reprojection error thresholds. Thus, we updated the definition of each IC. The description of standard ICs and the derivation of their update are presented in [**15**].

To evaluate the performance of polynomial degree estimation, we generated



(a) Homography matrices estimation (b) Fundamental matrices estimation

Figure 34: Evaluation of the cumulative distribution function of the execution times (in seconds) for pre-emptive model verification strategies. We tested the affine (AC) and point-based (PC) robust model estimation for homography and fundamental matrix.



(a) Example of fitted polynomial deg = 3. (b) Example of fitted polynomial deg = 4.

Figure 35: The visualisation of example polynomials $\widetilde{f}_{\text{pol}}$ that are used to compare the IC with ACS method on the well-studied problem of polynomial degree estimation.

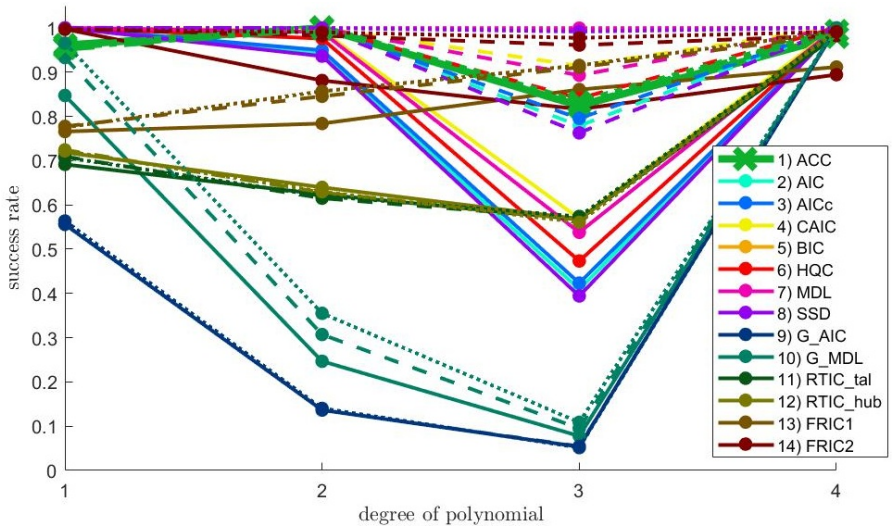


Figure 36: The success rate of the correctly estimated degree of the polynomial for 13 standard information criterion methods and the AC criterion. Each method was tested 10k times on each polynomial degree. The polynomial fitting was done statistically optimal from 100 equally distributed samples in the interval $[-5, 5]$. The [solid, dashed, dotted] lines correspond to the positional noise $\widetilde{\epsilon}_{\text{pol}}$ of the measurements, i.e., $\widetilde{\epsilon}_{\text{pol}} \in \mathcal{N}(0, \widetilde{\sigma}^2_{\text{pol}})$ where $\widetilde{\sigma}^2_{\text{pol}} = [10^{-2}, 10^{-3}, 10^{-4}]$.

10,000 polynomials $\widetilde{y}_{\text{pol}} = \widetilde{f}_{\text{pol}}(\widetilde{x}_{\text{pol}}) + \widetilde{\epsilon}_{\text{pol}}$ of each degree $1, 2, 3, 4$ without outliers and with different standard deviations $\widetilde{\epsilon}_{\text{pol}} \in \mathcal{N}(0, \widetilde{\sigma}^2_{\text{pol}})$ where $\widetilde{\sigma}^2_{\text{pol}} = \{10^{-2}, 10^{-3}, 10^{-4}\}$. Each polynomial has coefficients in the range $[-1.5, , 1.5]$, and is evaluated in the interval $\widetilde{x}_{\text{pol}} \in [-5, 5]$ to ensure a significant part of the polynomial is observed. The examples are shown in Fig. 35. We compare the ACS method with 13 standard ICs on these polynomials, resulting in an overall success rate of 94.1% over 120k trials, demonstrating its practicality. The success rate for individual polynomial degrees is shown in Fig. 36.

Evaluation on synthetic datasets. In this experiment, we compared the performance of standard IC, ACS, and LACS on synthetically created sub-reconstructions. To model the reconstructions from images realistically, we utilized eight real cameras consisting of low-cost web cameras, cellphones, fish-eye and DSLR cameras calibrated by checkerboard pattern and camera models $\mathbf{M}_{0|0}$, $\mathbf{M}_{1|0}$, $\mathbf{M}_{2|0}$, $\mathbf{M}_{3|0}$, $\mathbf{M}_{4|0}$, $\mathbf{M}_{1|1}$, $\mathbf{M}_{2|2}$, $\mathbf{M}_{3|3}$. We refer the reader to the paper **[15]** for the actual intrinsic parameters of each camera model. Next, to obtain realistic noise of the keypoints, we estimated $4,5$M the covariance matrices of the keypoints from 454 images of the ETH3D dataset introduced in Schoeps [151] as the scaled inversion of the structure tensor studied in Förstner [60]. These covariance matrices

were randomly assigned to projections of 3D points into the cameras to generate 104 new reconstructions out of 13 ETH3D reconstructions (i.e., 2839 images and $10,3$M keypoints).

These reconstructions were split into [training, validation, evaluation] parts in respective ratios [0.8, 0.1, 0.1] and utilized to train the LACS. The ACS outputs for thresholds $\delta \in \{0.5, 1, 1.5, 2\}$px, realized by a vector $\boldsymbol{x}_i$ for camera model $\mathbf{M}_i$, were normalized by

$$f_{\text{norm}}(\boldsymbol{x}_i) = \frac{4\,(\boldsymbol{x}_i - \min(\boldsymbol{x}_i))}{(\max(\boldsymbol{x}_i) - \min(\boldsymbol{x}_i))} + 1\,. \tag{177}$$

Therefore we obtain a matrix of camera models (columns) and reprojection thresholds (rows) where each value corresponds to ACS output. We used the Adam [159] optimizer with learning rate 10^{-4} and standard Cross Entropy Loss function. To avoid overfitting, we trained for 4k epochs and selected the model with the lowest validation loss.

To evaluate the success rate of camera model selection, we tested 1k synthetic reconstructions for each camera model and each reconstruction size $L \in \{5, 10, 15\}$. This resulted in 24k sub-reconstructions, each consisting of $L-1$ neighboring cameras around one randomly selected camera in one of the 104 synthetic reconstructions. We added up to 20% of outliers to simulate real reconstruction mismatches by systematically permuting 3D point IDs for each of the 24k sub-reconstructions.

These sub-reconstructions were employed for success rate evaluation of the ICs, ACS, and LACS methods. The Fig. 37 show the success rate of correctly estimated camera model on $\mathcal{M}_{\text{small}} = \{\mathbf{M}_{0|0}, \mathbf{M}_{1|0}, \mathbf{M}_{2|0}, \mathbf{M}_{3|0}, \mathbf{M}_{4|0}\}$ camera models. Fig. 38 shows the same statistic using all the models in the dataset.

The increasing success ratios for a growing number of registered cameras in the sub-reconstruction are in Fig. 5. We can see that the LACS method leads to a reliable estimate of the camera model for $L = 15$ images in the sub-reconstruction.

Another statistic evaluated on the synthetic datasets is the confusion matrix, which is shown in Fig. 6. This matrix presents the relative number of selected camera models for a given ground truth camera model. If we always select the correct camera model, the confusion matrix would equal the identity matrix.

Evaluation on real datasets. In this experiment, we evaluated the AC criterion as well as the ACS and LACS methods on real datasets, including the ETH3D and KITTI [160] datasets. The ETH3D datasets provide the camera poses, enabling us to measure the positional error of the reconstructed cameras. The reconstructions

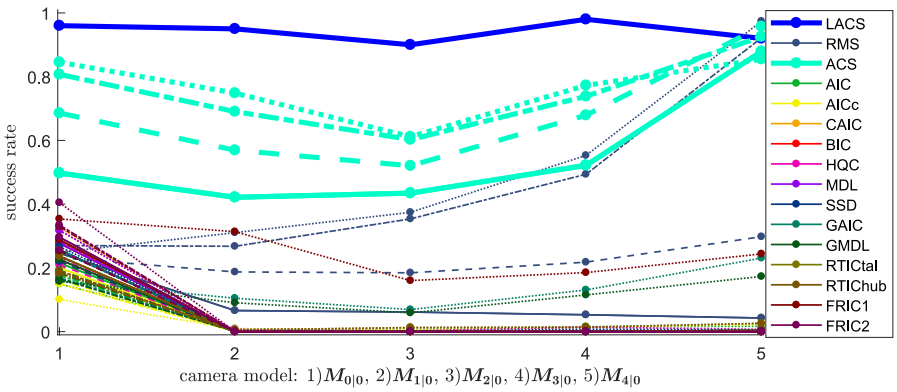


Figure 37: Success rates of the information criteria for camera model selection. The lines {solid, long dashed, short dashed, doted} correspond to a weighted reprojection error threshold of {0.5, 1, 1.5, 2}px. LACS uses all these thresholds to select the camera model.

Classifier / L	5	10	15
ACS [0.5px]	0.47	0.53	0.51
ACS [1.0px]	0.55	0.65	0.66
ACS [1.5px]	0.56	0.70	0.70
ACS [2.0px]	0.48	0.68	0.76
LACS	**0.68**	**0.83**	**0.93**

Table 5: The mean success ratios of the ACS and LACS classifiers w.r.t. an increasing number of registered cameras L. We assumed the $\mathcal{M}_{\text{small}}$ set of camera models.

were generated using the COLMAP pipeline, bundle adjustment using the Ceres solver, and derivatives of projection functions via developed USfM framework [**99**]. All execution time measurements were performed on a single computer with an AMD Ryzen 7 1700X processor.

The first experiment showcases the change in AC during the registration of the first 15 cameras by COLMAP on the *terrains rig* dataset from ETH3D, as illustrated in Fig. 39. The addition of a new camera is expected to increase the AC by providing more information and decreasing the covariance matrix, as shown ineq. (166). However, the small decrease in AC is due to additional filtering of inconsistent observations performed within COLMAP. The resulting reconstructions using all the images are visualized in Fig. 32.

Tables 7 and 8 compare the reconstructions obtained by the tested camera models on two datasets, one from ETH3D and the other from KITTI. A rectan-

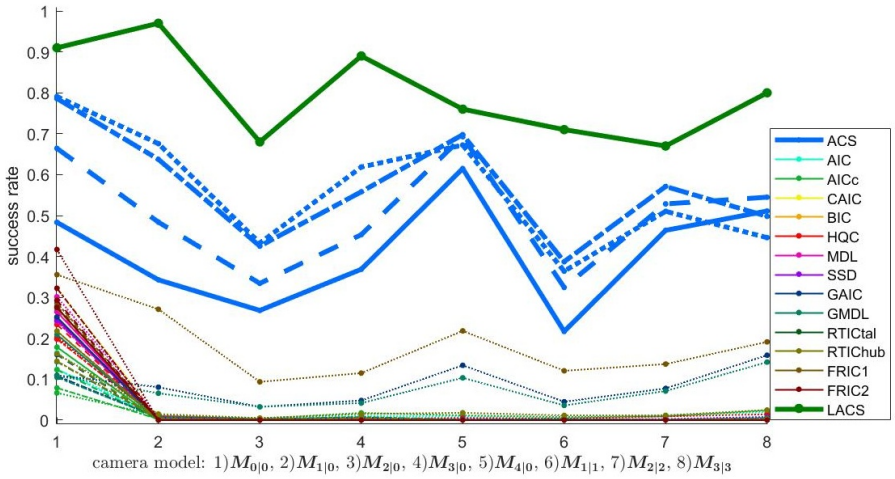


Figure 38: Success rates of the information criteria for camera model selection. The lines {solid, long dashed, short dashed, doted} correspond to a weighted reprojection error threshold of {0.5, 1, 1.5, 2}px. LACS uses all these thresholds to select the camera model.

gle highlights the camera model selected by LACS, and bold text highlights the best result achieved in each category, such as the number of registered cameras, number of points in 3D, or weighted reprojection error. Red rows indicate sub-reconstructions that could not be calculated in T_d time. In the case of ETH3D, we also measured the mean positional error of the estimated camera poses. We observed that the execution time increases if the camera model is over-parametrized or too restrictive. In the case of over-fitting unnecessary camera parameters, most of the time is spent on parameter optimization in bundle adjustment. We also observed that all 3D points are removed after the registration of a few cameras (e.g., < 15), and the reconstruction starts from scratch. For example, unsuccessful trials of the SfM initialization increased the reconstruction time from 175.7 sec to 1545 sec in the case of the *terrains_rig* dataset. If the camera model is too simple, repetitive cycles of adding, optimizing, and removing 3D points (that would be well explained by a suitable camera model) occur. The time overhead caused by the LACS method depends on γ. Still, it is negligible compared to the speedup caused by the utilization of a suitable camera model on large-scale reconstructions. Moreover, LACS builds the sub-reconstruction for all camera models, providing an approximate camera calibration that can improve the accuracy of the large-scale reconstruction. The visualization of the *terrains_rig* reconstructions using all camera models is in Fig. 39. The same visualization for the *2011_09_26_drive_0001* dataset from KITTI is in Fig. 40. More evaluation tables and visualization figures can be found in Polic [**15**].

ACS for $\delta = 2$px					LACS				
0.84	0.13	0.01	0.01	0.01	**0.96**	0.02	0.02	0.00	0.00
0.10	**0.72**	0.12	0.04	0.02	0.00	**0.95**	0.05	0.00	0.00
0.04	0.19	**0.63**	0.09	0.06	0.00	0.10	**0.90**	0.00	0.00
0.00	0.01	0.03	**0.81**	0.14	0.00	0.00	0.00	**0.98**	0.02
0.08	0.02	0.04	0.04	**0.83**	0.00	0.00	0.02	0.06	**0.92**

Table 6: The confusion matrices for the ACS and LACS methods were evaluated on synthetic sub-reconstructions. The ACS method used a threshold of $\delta = 2$px, while the LACS method benefited from all thresholds of reprojection errors $\delta \in 0.5, 1, 1.5, 2$px. The rows of the confusion matrices correspond to the ground truth camera models, and the columns correspond to the selected camera models from $\mathcal{M}_{\text{small}} = \{\mathbf{M}_{0|0}, \mathbf{M}_{1|0}, \mathbf{M}_{2|0}, \mathbf{M}_{3|0}, \mathbf{M}_{4|0}\}$ using the evaluated method. If the correct camera model is always selected, the confusion matrix would equal the identity matrix.

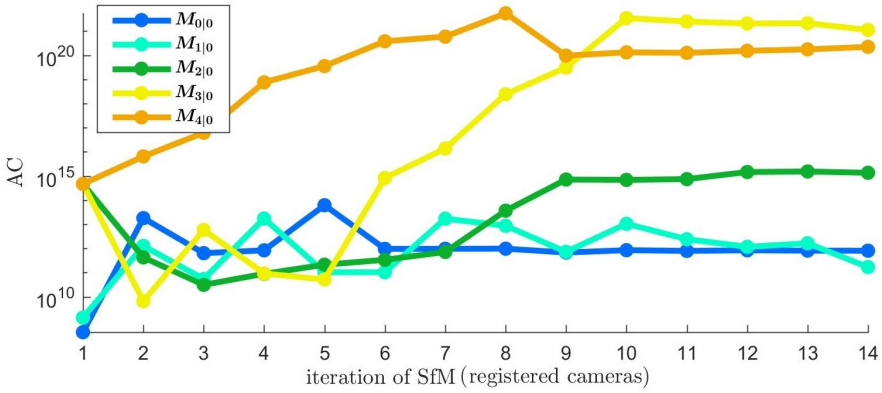


Figure 39: The dependence of AC criteria on iterations for the *terrains_rig* dataset [151] for 1-15 registered cameras.

$\mathcal{M}$	T_1	T_{all}	$\widetilde{L}$	$\widetilde{M}$	$\widetilde{N}$	$\sqrt{\Omega(\tilde{\boldsymbol{\theta}})}$	$\widetilde{Q}_C$
$\mathbf{M}_{0\|0}$	**7.0**	1070.2	34	6.7	64.1	0.7	2.5
$\mathbf{M}_{1\|0}$	32.0	626.8	**165**	**17.5**	210.5	0.9	7.0
$\mathbf{M}_{2\|0}$	16.9	212.7	**165**	**17.5**	**210.9**	0.8	6.4
$\mathbf{M}_{3\|0}$	17.6	175.7	**165**	17.3	210.2	0.7	3.1
$\mathbf{M}_{4\|0}$	29.5	215.1	**165**	17.2	209.6	0.7	3.7
$\mathbf{M}_{1\|1}$	12.0	**172.3**	**165**	17.2	209.6	0.7	3.5
$\mathbf{M}_{2\|2}$	94.7	1443.8	**165**	17.3	210.1	0.8	3.7
$\mathbf{M}_{3\|3}$	83.4	1545.0	18	4.3	27.0	**0.5**	**0.8**

Table 7: Evaluation of camera model selection on the *terrains_rig* [151] with known ground truth camera poses. The reprojection threshold used by COLMAP was $\delta = 2$px. The time T_1 realizes the time required to reconstruct $L = 15$ cameras, and T_{all} denotes the time of the overall reconstruction process. We assumed $\gamma = 5$, i.e., $T_d = 35$sec. The $\sqrt{\Omega(\tilde{\boldsymbol{\theta}})}$ realize the weighted reprojection error, see eq. (43). The mean distance $\widetilde{Q}_C$ [cm] realizes the distance between the estimated camera centers and GT camera centers after aligning them by Similarity transformation.

$\mathcal{M}$	T_1	T_{all}	$\widetilde{L}$	$\widetilde{M}$	$\widetilde{N}$	$\sqrt{\Omega(\tilde{\boldsymbol{\theta}})}$	$\widetilde{Q}_C$
$\mathbf{M}_{0\|0}$	113.5	1323.6	**114**	34.4	305.4	0.8	–
$\mathbf{M}_{1\|0}$	91.7	1401.3	**114**	52.3	424.2	0.6	–
$\mathbf{M}_{2\|0}$	**84.2**	1407.1	**114**	64.7	502.7	0.6	–
$\mathbf{M}_{3\|0}$	105.7	**1272.2**	**114**	**66.2**	**504.5**	0.6	–
$\mathbf{M}_{4\|0}$	-	2238.4	0	0	0	–	–
$\mathbf{M}_{1\|1}$	206.1	1628.0	**114**	64.7	496.7	0.6	–
$\mathbf{M}_{2\|2}$	-	**431.2**	12	7.6	104.8	**0.4**	–
$\mathbf{M}_{3\|3}$	-	1543.2	0	0	0	–	–

Table 8: Evaluation of camera model selection on the *2011_09_26_drive_0001* KITTI [160] without known ground truth camera poses. The reprojection threshold used by COLMAP was $\delta = 2$px. The time T_1 realizes the time required to reconstruct $L = 15$ cameras, and T_{all} denotes the time of the overall reconstruction process. We assumed $\gamma = 5$, i.e., $T_d = 35$sec. The $\sqrt{\Omega(\tilde{\boldsymbol{\theta}})}$ realize the weighted reprojection error, see eq. (43). The mean distance $\widetilde{Q}_C$ [cm] realizes the distance between the estimated camera centers and GT camera centers after aligning them by Similarity transformation.

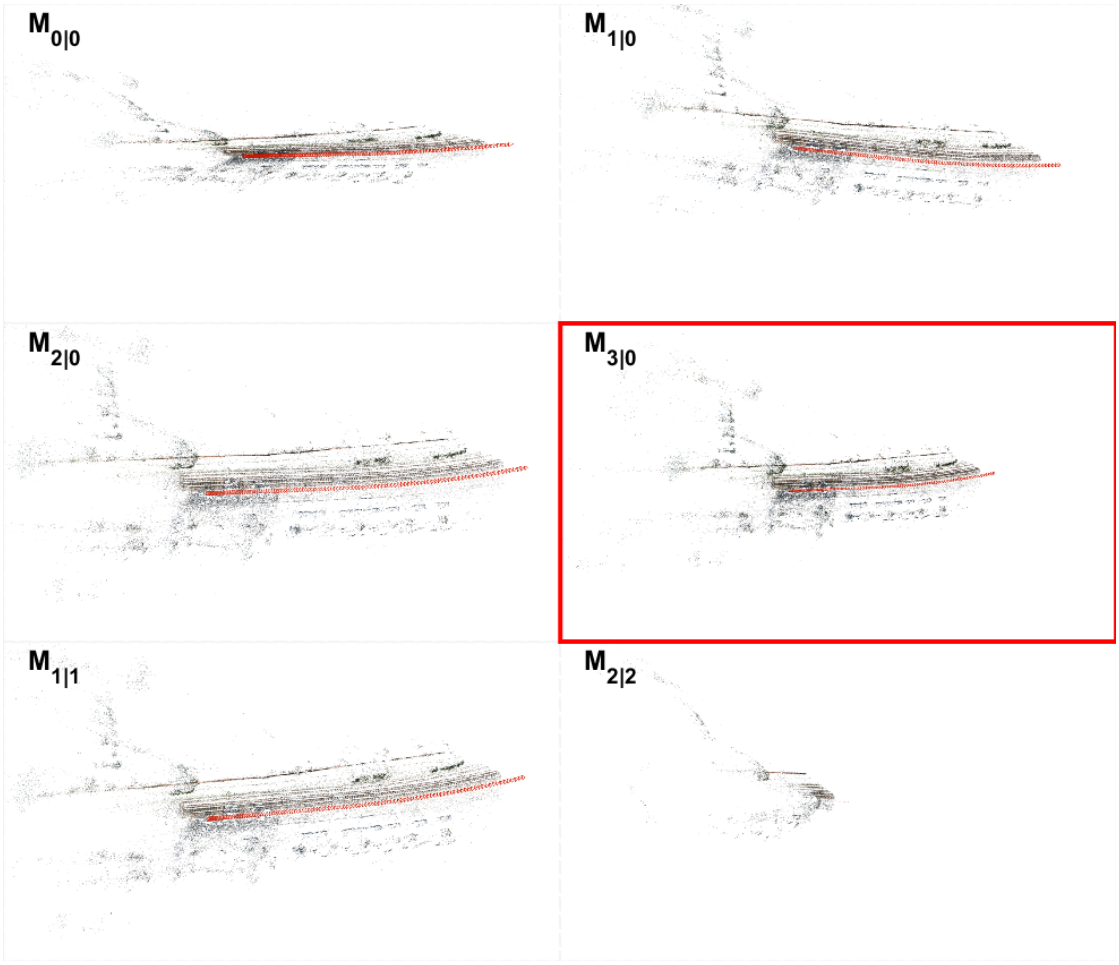


Figure 40: The 3D reconstructions of KITTI drive 0001 dataset by individual camera models. The camera model $\mathbf{M}_{3|3}$ failed to register more than three cameras.

10 Conclusion

This thesis offers an extensive overview of the uncertainty usage in Structure from Motion. The text describes key concepts such as uncertainty description, propagation, gauge specification, suitable model selection for observed data, the definition of camera geometry constraints, and reconstruction optimization. The comparison of state-of-the-art approaches for estimating the uncertainty of keypoints is covered and extended to obtain statistically consistent covariance matrices and positional uncertainty of affine regions. Deriving the uncertainty of other feature point parameters, such as scale and orientation, from template matching is more complex than for keypoint covariance matrices. To address this challenge, we created a large dataset of homographies and decomposed them to the feature point transformations between image pairs. By calculating the difference between reference and estimated transformations, we were able to calculate the variances of individual feature point transformations. We then propagated these uncertainties to the uncertainties of scale and orientation of feature points. This resulted in the first published estimate of the standard deviations of the scale and orientation of the SIFT detector.

In the second part, we utilize the uncertainties of feature points and propose new techniques for propagating these uncertainties. A new scheme, which employs constraints between parameters, is presented to simplify uncertainty propagation for minimal camera geometry problems, e.g., relative pose solvers. The method is empirically verified and utilized to derive a library of uncertainty propagation functions for common minimal problems. Uncertainty propagation from keypoints to sparse reconstruction using the projection function is a challenging problem due to the rank deficiency of the Fisher information matrix. To overcome this challenge, a damping term is added, and the inversion of the Fisher information matrix is expressed as the Taylor expansion at a point where the damping term equals zero. The second developed approach bounds the Fisher information matrix by its nullspace and allows the direct calculation of Moore-Pensore inversion as a simple inversion of the extended matrix.

The last part presents two applications that benefit from the estimated uncertainty. The first application speeds up robust model estimation by initializing the Sequential Probability Ratio Test with the probability of having a large number of inliers for the estimated model. This initialization avoids an unnecessary extensive verification for too uncertain or too accurate solutions. The second application derives a new statistical accuracy-based criterion that realizes a metric of the mathematical model's suitability for given observations. This is the first approach that works for automatic camera model selection from a general set of images. Furthermore, an extension that benefits from multiple reprojection error thresholds is trained and presented.

In summary, this thesis describes the basics about uncertainty propagation, extends uncertainty estimation for keypoints and affine regions, derives new schemes for uncertainty propagation, and shows the application of its estimate to speed up, robustify, and build more accurate reconstructions.

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] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. Hloc: Hints-based geolocation leveraging multiple measurement frameworks. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–9. IEEE, 2017.

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

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

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

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

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

[9] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[10] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[11] 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.

[12] 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.

[13] 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.

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

[15] Michal Polic, Stanislav Steidl, Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. Uncertainty based camera model selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5991–6000, 2020.

[16] 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.

[17] Daniel Barath, Michal Polic, Wolfgang Förstner, Torsten Sattler, Tomas Pajdla, and Zuzana Kukelova. Making affine correspondences work in camera geometry computation. In *European Conference on Computer Vision*, pages 723–740. Springer, 2020.

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[27] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 8092–8101, 2019.

[28] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32:12405–12415, 2019.

[29] 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.

[30] 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.

[31] 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.

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

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

[34] 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.

[35] Zuzana Kukelova, Jan Heller, Martin Bujnak, Andrew Fitzgibbon, and Tomas Pajdla. Efficient solution to the epipolar geometry for radially distorted cameras. In *Proceedings of the IEEE international conference on computer vision*, pages 2309–2317, 2015.

[36] 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.

[37] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[38] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020.

[39] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixé. Patch2pix: Epipolar-guided pixel-level correspondences supplementary material. *Training*, 480:320.

[40] 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.

[41] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *Proceedings of the European conference on computer vision (ECCV)*, pages 318–332, 2018.

[42] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1799–1808, 2016.

[43] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Camera pose estimation with unknown principal point. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2984–2992, 2018.

[44] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2316–2324, 2017.

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

[46] 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.

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

[48] Realitycapture: Mapping and 3d modeling photogrammetry software.

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

[50] 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.

[51] 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.

[52] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[53] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[54] Ken-ichi Kanatani. Uncertainty modeling and model selection for geometric inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1307–1319, 2004.

[55] J. C. Dainty and R. Shaw. *Image Science.* Academic Press, 1974.

[56] R. Szeliski. *Computer Vision: Algorithms and Applications.* Springer, 2010.

[57] Chris Sweeney. Theia multiview geometry library: Tutorial & reference. `http://theia-sfm.org`.

[58] 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.

[59] 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.

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

[61] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[62] Daniel D Morris and Takeo Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 696–702. IEEE, 1998.

[63] Ajit Singh. An estimation-theoretic framework for image-flow computation. 1989.

[64] Wolfgang Förstner. Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. *Computer Vision, Graphics, and Image Processing*, 40(3):273–310, 1987.

[65] Jianbo Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[66] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. Exploiting uncertainty in random sample consensus. In *International Conference on Computer Vision*, pages 2074–2081. IEEE, 2009.

[67] 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.

[68] Daniel Barath, Dmytro Mishkin, Michal Polic, Wolfgang Förstner, and Jiri Matas. A large scale homography benchmark. *arXiv preprint arXiv:2302.09997*, 2023.

[69] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI '81*, pages 674–679, 1981.

[70] W. Förstner. On the geometric precision of digital correlation. In J. Hakkarainen, E. Kilpelä, and A. Savolainen, editors, *Intl. Archives of Photogrammetry*, volume XXIV–3, pages 176–189. ISPRS Symposium, Comm. III, Helsinki, Jun 1982.

[71] F. Ackermann. Digital image correlation: Performance and potential application in photogrammetry. *Photogrammetric Record*, 11(64):429–439, 1984.

[72] R.M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume II. Addison-Wesley, Reading, MA, 1992.

[73] C. Raposo and J. P. Barreto. Theory and practice of structure-from-motion using affine correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5470–5478, 2016.

[74] Iván Eichhardt and Dmitry Chetverikov. Affine correspondences between central cameras for rapid relative pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 482–497, 2018.

[75] 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.

[76] A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer, 2001.

[77] Evgeni Begelfor and Michael Werman. How to put probabilities on homographies. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1666–1670, 2005.

[78] Frédéric Sur, Nicolas Noury, and Marie-Odile Berger. Computing the uncertainty of the 8 point algorithm for fundamental matrix estimation. 2008.

[79] Gabriella Csurka, Cyril Zeller, Zhengyou Zhang, and Olivier D Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer vision and image understanding*, 68(1):18–36, 1997.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] Daniel Barath, Michal Polic, Wolfgang Förstner, Torsten Sattler, Tomas Pajdla, and Zuzana Kukelova. Making affine correspondences work in camera geometry computation. In *European Conference on Computer Vision*, pages 723–740. Springer, 2020.

[85] Yasushi Kanazawa and Kenichi Kanatani. Do we really have to consider covariance matrices for image feature points? *Electronics and communications in Japan (part III: Fundamental electronic science)*, 86(1):1–10, 2003.

[86] Stephen M Smith and J Michael Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[87] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5987–5997, 2021.

[88] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4669–4678, 2021.

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

[90] Daniel D Morris. *Gauge freedoms and uncertainty modeling for 3D computer vision*. PhD thesis, Citeseer, 2001.

[91] Viorela Ila, Lukas Polok, Marek Solony, and Pavel Svoboda. Slam++-a highly efficient and temporally scalable incremental slam framework. *The International Journal of Robotics Research*, 36(2):210–230, 2017.

[92] Michael Kaess and Frank Dellaert. Covariance recovery from a square root information matrix for data association. *Robotics and autonomous systems*, 57(12):1198–1210, 2009.

[93] Viorela Ila, Lukas Polok, Marek Solony, and Klemen Istenic. Fast incremental bundle adjustment with covariance recovery. In *International Conference on 3D Vision (3DV)*, Oct 2017.

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

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

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

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

[98] Michal Polic and Tomas Pajdla. Camera uncertainty computation in large 3d reconstruction. In *International Conference on 3D Vision*, 2017.

[99] Michal Polic, Wolfgang Forstner, and Tomas Pajdla. Fast and accurate camera covariance computation for large 3d reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 679–694, 2018.

[100] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European conference on computer vision*, pages 128–142. Springer, 2002.

[101] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.

[102] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 141:81–93, 2015.

[103] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision*, pages 284–300, 2018.

[104] Ondřej Chum and Jiří Matas. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008.

[105] Hirotugu Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

[106] Clifford M Hurvich and Chih-Ling Tsai. A corrected akaike information criterion for vector autoregressive model selection. *Journal of time series analysis*, 14(3):271–279, 1993.

[107] Hamparsum Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[108] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[109] Edward J Hannan and Barry G Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):190–195, 1979.

[110] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[111] Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636, 1984.

[112] Elvezio Ronchetti. Robust model selection in regression. Technical report, PRINCETON UNIV NJ DEPT OF STATISTICS, 1984.

[113] Jose AF Machado. Robust model selection and m-estimation. *Econometric Theory*, 9(3):478–493, 1993.

[114] Patrick Bouthemy, Bertha Mayela Toledo Acosta, and Bernard Delyon. Robust model selection in 2d parametric motion estimation. *Journal of Mathematical Imaging and Vision*, pages 1–15, 2019.

[115] Kenneth P Burnham and David R Anderson. A practical information-theoretic approach. *Model selection and multimodel inference, 2nd ed. Springer, New York*, 2002.

[116] Sumio Watanabe. A widely applicable bayesian information criterion. *Journal of Machine Learning Research*, 14(Mar):867–897, 2013.

[117] Keisuke Kinoshita and L Lindenbaum. Camera model selection based on geometric aic. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 514–519. IEEE, 2000.

[118] Rihab K Hamad, Baidaa Hamed, and HA Hassonny. The automatic selection of radial distortion models. *International Journal of Computer Applications*, 975:8887.

[119] Vitaliy Orekhov, Besma Abidi, Chris Broaddus, and Mongi Abidi. Universal camera calibration with automatic distortion model selection. In *2007 IEEE International Conference on Image Processing*, volume 6, pages VI–397. IEEE, 2007.

[120] Kevin Köser. *Geometric Estimation with Local Affine Frames and Free-form Surfaces*. Shaker, 2009.

[121] Daniel Barath and Levente Hajder. A theory of point-wise homography estimation. *Pattern Recognition Letters*, 94:7–14, 2017.

[122] Daniel Barath and Levente Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 27(11):5328–5337, 2018.

[123] Athanasios Papoulis. *Random variables and stochastic processes*. McGraw Hill, 1965.

[124] W. Baarda. S-transformations and criterion matrices. *Publication on Geodesy*, 1973.

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

[126] Viktor Larsson, Torsten Sattler, Zuzana Kukelova, and Marc Pollefeys. Revisiting radial distortion absolute pose. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1062–1071, 2019.

[127] AliceVision. Meshroom: A 3D reconstruction software., 2018.

[128] J Bentolila and J. M. Francos. Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding*, 122:105–114, 2014.

[129] M. Fischler and R. Bolles. Random Sampling Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communications of the ACM*, 24:381–395, 1981.

[130] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[131] Thomas Schops, Viktor Larsson, Marc Pollefeys, and Torsten Sattler. Why having 10,000 parameters in your camera model is better than twelve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2535–2544, 2020.

[132] Timo Dickscheid, Falko Schindler, and Wolfgang Förstner. Coding images with local features. *International journal of computer vision*, 94(2):154–174, 2011.

[133] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022.

[134] Thomas Läbe, Timo Dickscheid, and Wolfgang Förstner. On the Quality of Automatic Relative Orientation Procedures. In *ISPRS Archives*, volume XXXVII Part B3b, pages 37–42, 2008.

[135] K. Wilson and N. Snavely. Robust Global Translations with 1DSfM. In *European Conference on Computer Vision*, pages 61–75, 2014.

[136] Google maps. `http://maps.google.com/`.

[137] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[138] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012.

[139] Daniel Barath, Denys Rozumny, Ivan Eichhardt, Levente Hajder, and Jiri Matas. Progressive-X+: Clustering in the consensus space. *arXiv preprint arXiv:2103.13875*, 2021.

[140] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA, 2007.

[141] Bernhard Zeisl, Pierre Fite Georgel, Florian Schweiger, Eckehard Steinbach, and Nassir Navab. Estimation of Location Uncertainty for Scale Invariant Feature Points. In *Proc. BMVC*, pages 57.1–57.12, 2009. doi:10.5244/C.23.57.

[142] D. Baráth, T. Tóth, and L. Hajder. A minimal solution for two-view focal-length estimation using two affine correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[143] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4912–4921, 2017.

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

[145] Edward Anderson, Zhaojun Bai, Jack Dongarra, Anne Greenbaum, Alan McKenney, Jeremy Du Croz, Sven Hammarling, James Demmel, C Bischof, and Danny Sorensen. Lapack: A portable linear algebra library for high-performance computers. In *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, pages 2–11. IEEE Computer Society Press, 1990.

[146] Simone Gasparini, Fabien Castan, and Yann Lanthony. Buddha dataset, January 2017. https://github.com/alicevision/dataset_buddha.

[147] Wolfgang Förstner and Kourosh Khoshelham. Efficient and accurate registration of point clouds with plane to plane correspondences. In *3rd International Workshop on Recovering 6D Object Pose*, 2017.

[148] J. Schneider, C. Stachniss, and W. Förstner. On the quality and efficiency of approximate solutions to bundle adjustment with epipolar and trifocal constraints. In *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-2/W3, pages 81–88, 2017.

[149] Jiri Matas and Ondrej Chum. Randomized RANSAC with sequential probability ratio test. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1727–1732. IEEE, 2005.

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

[151] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[152] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[153] Jia-Wang Bian, Yu-Huan Wu, Ji Zhao, Yun Liu, Le Zhang, Ming-Ming Cheng, and Ian Reid. An evaluation of feature matchers for-fundamental matrix estimation. *arXiv preprint arXiv:1908.09474*, 2019. https://jwbian.net/fm-bench.

[154] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.

[155] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[156] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78, 2017.

[157] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5173–5182, 2017.

[158] Colin L Mallows. Some comments on c p. *Technometrics*, 15(4):661–675, 1973.

[159] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[160] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.