All rights reserved to the author

This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for “fair use” of brief quotations for academic instruction, criticism, or research purposes only. Commercial use of this material is completely prohibited.

כל הזכויות שמורות למחבר/ת

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיץ באינטרנט, חיבור זה או כל חלק ממנו, למעט "שימוש הוגן" בקטעים קצרים מן החיבור למטרות לימוד, הוראה, בקורת או מחקר. שימוש מסחרי בחומר הכלול בחיבור זה אסור בחללים.
Robust Epipolar Geometry Estimation Using Noisy Pose Priors

Yehonatan Goldman
Robust Epipolar Geometry Estimation Using Noisy Pose Priors

Research Thesis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Yehonatan Goldman

Submitted to the Senate of the Technion — Israel Institute of Technology
Sivan 5774 Haifa June 2014
This research was carried out under the supervision of Prof. Ehud Rivlin, in the Faculty of Computer Science, and Prof. Ilan Shimshoni from Haifa University.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author’s master research period, the most up-to-date versions of which being:

# Contents

List of Figures

Abstract

Abbreviations and Notations

1 Introduction

2 Mathematical Background
  2.1 Epipolar Geometry
  2.2 Relative Pose and the Vector of Unknowns

3 Related Work
  3.1 Introduction
  3.2 Image Based Only Methods
    3.2.1 RANSAC
    3.2.2 Non-RANSAC Algorithms
  3.3 Model Generation Methods that Exploit Pose Priors
    3.3.1 Limited Correspondences Search Area
    3.3.2 Known Rotations
  3.4 Bundle Adjustment

4 Relative Pose Measurements
  4.1 Introduction
  4.2 Pose Measurements in Smartphones
  4.3 Pose Noise Propagation
  4.4 Pose Measurements and IMU Errors
  4.5 Summary

5 SOREPP
  5.1 Introduction
  5.2 The Concept
  5.3 Understanding SOREPP
    5.3.1 Core Principle
5.3.2 The Robust Cost Function ........................................ 29
5.3.3 The Regularization Term ........................................ 30
5.3.4 Other Pose Prior Sources ...................................... 31
5.4 SOREPP Components ............................................... 31
   5.4.1 Correspondence Weights .................................... 31
   5.4.2 Pose Region Search ......................................... 33
   5.4.3 Solution Refinement ....................................... 33
   5.4.4 Inverse Translation ....................................... 34
   5.4.5 Solution Grade ........................................... 34
6 Experiments ............................................................. 35
   6.1 Introduction .................................................. 35
   6.2 Experiments on Synthetic Data ................................ 35
      6.2.1 Experimental Setup .................................... 35
      6.2.2 Distant Cameras ....................................... 38
      6.2.3 Close Cameras .......................................... 40
   6.3 Experiments on Real Images ................................... 41
      6.3.1 Experimental Setup .................................... 41
      6.3.2 Analyzing SOREPP Components ........................ 43
      6.3.3 Performance Comparison ............................... 44
7 Conclusion and Further Research .................................. 51
A Rotation Matrix Calculations .................................... 53
List of Figures

1.1 Wide baseline images for which the epipolar geometry could not be estimated using current algorithms but was successfully estimated by SOREPP. ...................................................... 8
1.2 An illustration of epipolar geometry. .............................. 12
2.1 The coordinates system used in our work. ................. 13
2.2 The coordinates system used in our work. ................. 13
4.1 Samson errors in simulated analysis. ....................... 24
5.1 Illustration of the main concept behind SOREPP. ........... 30
5.2 Illustration of the Pose Region Search component. .......... 32
5.3 SOREPP probability for success as function of $v$. ............... 34
6.1 Results on synthetic data, distant cameras, 80% inliers. .... 37
6.2 Results on synthetic data, distant cameras, 10% inliers. .... 38
6.3 Results on synthetic data, close cameras with unknown translation vector, 80% inliers. .......................... 39
6.4 Results on synthetic data, close cameras with unknown translation vector, 10% inliers. .......................... 40
6.5 Algorithm performance compared over all the sets. ......... 46
6.6 Images from Open1 dataset matched using some of the algorithms. . 47
6.7 More images from Open1 dataset. ............................ 48
6.8 Images from Open2 dataset. ............................... 49
6.9 Images from Urban dataset. ............................... 50
Abstract

Epipolar geometry estimation is fundamental to many computer vision algorithms. It has therefore attracted a lot of interest in recent years, yielding high quality estimation algorithms for wide baseline image pairs. Currently many types of cameras (e.g., in smartphones and robot navigation systems) produce geo-tagged images containing pose and internal calibration data. Exploiting this information as part of an epipolar geometry estimation algorithm may be useful but not trivial, since the pose measurement may be quite noisy. We introduce SOREPP, a novel estimation algorithm designed to exploit pose priors naturally. It sparsely samples the pose space around the measured pose and for a few promising candidates applies a robust optimization procedure. It uses all the putative correspondences simultaneously, even though many of them are outliers, yielding a very efficient algorithm whose runtime is independent of the inlier fractions. SOREPP was extensively tested on synthetic data and on hundreds of real image pairs taken by a smartphone. Its ability to handle challenging scenarios with extremely low inlier fractions of less than 10% was demonstrated as was its ability to handle images taken by close cameras. It outperforms current state-of-the-art algorithms that do not use pose priors as well as other algorithms that do.
Abbreviations and Notations

\( a \) : Accelerometer measurement  
\( \text{BA} \) : Bundle Adjustment  
\( c \) : A weighting parameter in the energy function  
\( d(k) \) : The Sampson distance of the \( k \)-th correspondence  
\( dt \) : Small time interval  
\( E \) : The essential matrix  
\( \text{ENU} \) : East North Up coordinate system  
\( \text{EXIF} \) : Exchangeable Image File Format  
\( F \) : The fundamental matrix  
\( \text{FOV} \) : Field Of View  
\( f(k) \) : The reprojection of the \( k \)-th correspondence on an image  
\( \text{GPS} \) : Global Positioning System  
\( g(k) \) : A Gaussian weight  
\( \text{IMU} \) : Inertial Measurements Unit  
\( J \) : Jacobian matrix  
\( J_{1,2} \) : The Jacobian matrix of the pose prior of cameras \( q \) or 2, respectively  
\( J(k) \) : The Jacobian matrix of the Sampson distance of the \( k \)-th correspondence according to \( s \)  
\( K \) : Intrinsic calibration matrix  
\( K_{1,2} \) : The intrinsic calibration matrix of image 1 or image 2, respectively  
\( k \) : Index of a correspondence  
\( \tilde{k} \) : Another index of a correspondence
LOS : Line Of Sight
l(k) : The epipolar line of the k-th correspondence
l_{1,2}(k) : The epipolar line of the k-th correspondence on image 1 or image 2, respectively
MEMS : Micro Electro Mechanical Systems
m : A parameter controlling the number of optimization initialization additional to the prior value
N : Number of iterations
n :
p : Image pixel
p_{1,2} : Image pixel on image 1 or image 2, respectively
q : Correlation between descriptors, used by BLOGS
R : Rotation matrix
R_r : The relative rotation matrix from camera 1 to 2, defined in the coordinate system of camera 1
R_{1,2} : The rotation matrix from world global coordinate to camera 1 or camera 2, respectively
R_x(\theta) : The rotation matrix around the x axis by Euler angle \theta
R_y(\phi) : The rotation matrix around the y axis by Euler angle \phi
R_z(\psi) : The rotation matrix around the z axis by Euler angle \psi
s : The vector of unknowns
\delta s : The difference between the estimated vector of unknowns s and the prior one s_0
\hat{s} : The estimated vector of unknowns
s_0 : The vector of unknowns calculated by prior values
|s| : The number of elements in s
t : Translation vector
t_r : The relative translation vector from camera 1 to 2, defined in the coordinate system of camera 1
t_{1,2} : The position of camera 1 or camera 2, respectively, in global coordinates
v : The energy value over all the putative correspondences
v_{threshold} : A threshold energy, controlling if to reduce \sigma_h and optimize the energy again or not
w(k) : The weight (probability) of the k-th correspondence being an inlier
\hat{w}(k) : The normalized weight (probability) of the k-th correspondence being an inlier
x : The x axis
y : The y axis
z : The z axis
\( \alpha \) : The horizontal coordinate of the translation vector \( t_r \)
\( \beta \) : The vertical coordinate of the translation vector \( t_r \)
\( \epsilon \) : The inlier fraction
\( \theta \) : Pitch. The Euler angle representing rotation around the \( x \) axis
\( \lambda \) : Mahalanobis distance of the estimated vector of unknowns \( s \) and the prior one \( s_0 \) according to the covariance matrix \( \Sigma_s \)
\( \rho \) : The distance between cameras 1 and 2
\( \Sigma \) : Covariance matrix
\( \Sigma_{s0} \) : The \( 5 \times 5 \) covariance matrix of the input vector of unknowns \( s_0 \)
\( \Sigma_{1,2} \) : The \( 6 \times 6 \) covariance matrix of the pose of cameras 1 or 2, respectively
\( \sigma \) : Standard deviation
\( \sigma_{s0} \) : The standard deviation vector of the input vector of unknowns \( s_0 \)
\( \sigma_h \) : A parameter defining the standard deviation of the Sampson distances
\( \sigma_x \) : The standard deviation of a random variable \( x \)
\( \tau \) : The success threshold (in pixels) of epipolar geometry estimated for an image pair
\( \phi \) : Roll. The Euler angle representing rotation around the \( y \) axis
\( \psi \) : Azimuth. The Euler angle representing rotation around the \( z \) axis
\( \Omega \) : A general set
\( |\Omega| \) : A general set size
\( \Omega_{set} \) : A specific set size
\( \Omega_{all} \) : The set of all putative correspondences
Chapter 1

Introduction

Epipolar geometry is the intrinsic projective geometry between two views [HZ04], and it is encoded in the fundamental matrix $F$. Its estimation is one of the core problems in computer vision and is used as a basic component for stereo matching [SS02], structure from motion (SfM) [SS06], vision-based robot navigation [KAS11], and other applications. The epipolar geometry of two images is usually estimated by finding corresponding features in both. This is done first by detecting and matching invariant features using an algorithm such as SIFT [Low04], followed by the application of a robust estimation method from the RANSAC [FB81] family. Typically, such algorithms randomly sample a minimal set of putative correspondences, searching for a valid, outlier-free set. For each set a model is estimated using a model generation method such as the eight-point algorithm [LH81] and scored by assessing all the putative correspondences. The chosen model is the one with maximal grade, that is, the one supported by the most correspondences.

The main weakness of RANSAC is the necessity to sample a valid set. As the inlier fraction (the percentage of inliers out of all the putative correspondences) decreases, the probability to sample a valid set drops rapidly, increasing greatly the required number of iterations. For example, using the five-point algorithm [Nis04] for an inlier fraction of 10% requires about 300,000 RANSAC iterations for a confidence of 95%, making the matching in such conditions impractical. Moreover, even sampling a valid set does not ensure accurate estimation [CMK03].

In recent years considerable progress has been made in developing estimation algorithms that tackle these problems. Such algorithms include LO-RANSAC [CMK03], PROSAC [CM05], BEEM [GS08], BLOGS [BS09], and recently USAC [RCP+13]. However, scenarios with wide baseline images or small overlapping regions between the images still challenge even the current state-of-the-art algorithms due to the low inlier fractions. Figure 1.1 shows several such challenging image pairs.

The fundamental matrix $F$ which we want to estimate is constructed from the camera parameters only, independent of the scene. The intrinsic parameters are mainly the focal lengths of the cameras and their principal points, and the extrinsic parameters
describe their relative pose. Focal length data has appeared for a number of years in the metadata of standard cameras under the EXIF format, and algorithms that use this data exist [SSS06, Nis04, HK07]. Nowadays the pose data is also becoming available as smartphones with built-in sensors such as a compass, accelerometers, gyros and a GPS receiver become popular. For vision-based robot navigation, low-cost inertial measurement units (IMUs) based on micro-electro-mechanical systems (MEMS) devices can be used to estimate the relative pose. This evolution makes images with attached camera parameters increasingly popular.

In principle, knowing the pose and the internal calibration parameters would make vision-based epipolar geometry estimation methods unnecessary, but this is not the case in practice since the accuracy of such sensors is low. For example, typical compass azimuth errors can exceed 7°. Therefore epipolar geometry estimation contaminated with such errors can be considered as a search problem in the parametric space of the relative pose. Yet, the noisy pose data is valuable and can be used in order to constrain the search and focus it.

This thesis tackles the following problem: How to effectively exploit pose priors for epipolar geometry estimation, while we take into account the fact that the pose priors are noisy or even partly incorrect. The paper focuses on two sources for pose priors: Smartphones and IMUs. Smartphone images may differ greatly from one another, leading to low inlier fractions. The pose measurements in both cameras participating in the estimation are relatively noisy and independent of each other. Therefore the noise is combined. In the vision-based robot navigation context, the pose priors come from an IMU. The images are usually close in time, resulting in a high inlier fraction. The IMU provides an accurate relative orientation prior but the translation vector may be
completely incorrect, due to the two integration steps done by the IMU while calculating it.

To handle these difficulties, we propose a novel estimation algorithm called SOREPP (Soft Optimization method for Robust Estimation based on Pose Priors). SOREPP optimizes an M-estimator cost function designed for robustness to putative outlier correspondences, while the pose priors are used to initialize the optimization and regularize the solution. Knowledge of the expected amount of noise is used to limit the search to a part of the parameter space. Because, in contrast to RANSAC, explicit detection of inliers is not required, a solution can be found even in extreme conditions, typically where the inlier fraction is around 10%. SOREPP is insensitive to significant pose noise, and may be able to deal with even a completely incorrect translation vector, by decreasing its robustness to outlier correspondences. The algorithm is simple yet effective. It exhibits fast runtime and does not depend on the inlier fractions, making it attractive for real-time applications. SOREPP achieves a notable improvement over the other methods both in runtime and accuracy for very challenging image pairs, such as those in Figure 1.1.

The thesis has three main contributions. First, it addresses the problem of epipolar geometry estimation when pose prior data is available. Unlike previous works we take into account all available pose information and especially the expected noise level based on a physical model. In the thesis we apply the method to two common types of sensors but it can be easily generalized to other sensor types. The second contribution is the expansion of the robust optimization framework that is usually used to refine a solution, enabling it to be directly applied to the raw data with low inlier fractions and high levels of pose uncertainty. As far as we know this is the first time that this approach has been taken in low inlier fraction cases, instead of RANSAC. Finally, we introduce an extensive experimental setup for testing the algorithm and comparing it to the state-of-the-art. With a relatively low amount of work on single images, a large number of image pairs are generated. This enables us to test and compare different algorithms on hundreds of image pairs.
Chapter 2

Mathematical Background

2.1 Epipolar Geometry

This section is based on Hartley and Zisserman’s book [HZ04] and reviews the main concepts used in this work. Let us assume that two images observe the same scene. More specifically, that the corresponding pixels $p_1$ and $p_2$ are the projection of the same world point $P$, as illustrated in Figure 2.1. In general, any three points in space lie on a plane, and in our case these points are $p_1$, $p_2$, and $P$. Enforcing this coplanarity property leads to the epipolar constraint, which is defined in the following way:

$$ p_2^T F p_1 = 0, \quad (2.1) $$

where $F$ is the fundamental matrix. It is constructed from the parameters of the cameras only, independent of the scene, as follows:

$$ F = (K_2^{-1})^T R_r [t_r]_x K_1^{-1}, \quad (2.2) $$

where $R_r$ is the relative rotation matrix, $t_r$ is the translation vector, and the symbol $[x]_x$ represents the cross-product matrix. The matrices $K_1$ and $K_2$ are the intrinsic calibration matrices of the image 1 and image 2 respectively. They transform a pixel on the image to the corresponding line-of-sight (LOS) in a coordinate system whose origin is in the center of projection of that camera. The calibration matrices are constructed mainly from the focal length of the camera and the principal point, which is commonly approximated as the image center. The relative pose $R_r$, $t_r$ defines the pose of camera 2 in the coordinate system of camera 1, whose origin is located at its center of projection. The relative rotation matrix $R_r$ can be represented as composed of three relative Euler angles. The unscaled translation vector $t_r$ represents the direction between the cameras’ centers of projection and has only two degrees of freedom. Thus, the epipolar geometry does not describe the distance between the cameras. For calibrated images, i.e., when the calibration matrices are known, the only unknowns are these five relative pose...
parameters, which form the essential matrix $E$:

$$E = R_r t_r = (K_2)^T F K_1.$$  \hfill (2.3)

The epipolar line is the projection of the line-of-sight defined by a point on one image, on the other image. For a static scene, the epipolar line passes through the corresponding point on the other image. The lines can be calculated from the fundamental matrix and image points in the following way:

$$l_2 = F p_1,$$
$$l_1 = F^T p_2,$$  \hfill (2.4)

where $l_1$ and $l_2$ are the epipolar lines on images 1 and image 2 respectively. Thus, the epipolar constraint may be interpreted as a point lying on its corresponding epipolar line.

Most epipolar geometry estimation algorithms seek to reduce the distances between points and their corresponding epipolar lines. A common distance function is the Sampson distance [HZ04], which approximates the geometric distance. The signed Sampson distance of the $k$-th correspondence is defined as follows:

$$d(k) = \frac{p_2(k)^T F p_1(k)}{\sqrt{l_{1,x}(k)^2 + l_{1,y}(k)^2 + l_{2,x}(k)^2 + l_{2,y}(k)^2}}.$$  \hfill (2.5)

where $l_{1,x}$, $l_{1,y}$ and $l_{2,x}$, $l_{2,y}$ are the first two components of the corresponding epipolar lines. Note that $d(k)$ is a function of the intrinsic parameters of both cameras and their relative pose, through $F$ (2.2) and the epipolar lines (2.4).
2.2 Relative Pose and the Vector of Unknowns

The pose of a platform is composed of its orientation and position. We use East-North-Up (ENU) coordinates to define the position. The orientation is described uniquely by a rotation matrix that can be decomposed to Euler angles, which are the three angles representing rotation around the Cartesian axes: Azimuth $\psi \in [-\pi, \pi)$ for the $z$ axis, pitch $\theta \in [-\pi/2, \pi/2]$ for the $x$ axis and roll $\phi \in [-\pi, \pi)$ for the $y$ axis. The coordinates system that is used is illustrated in Figure 2.2. The rotation matrix can by constructed by three rotation matrices, each representing a rotation by a single Euler angle around one axis:

$$R = R_y(\phi)R_x(\theta)R_z(\psi).$$  \hspace{1cm} (2.6)

The relative pose is represented by $R_r$ and $t_r$, which in some cases are measured directly. In other cases, when the pose is measured independently for each camera, we define $R_1$, $R_2$ and $t_1$, $t_2$ as the pose of the cameras in a global coordinate system. The relative pose is calculated as follows:

$$R_r = R_2R_1^T,$$
$$t_r = R_1(t_2 - t_1).$$  \hspace{1cm} (2.7)

The Euler angles composing $R_r$ are the azimuth $\psi_r$, the pitch $\theta_r$, and the roll $\phi_r$. The representation of the Euler angles has in general singular points when $\theta_r = \pm90^\circ$. Since the relative pitch of two cameras that view the same scene is mostly smaller, they are safe to use, as indicated in [TMHF00]. The advantage of using Euler angles is that they are more intuitive than other representations, and that orientation measurements are commonly given in Euler angles.

We can decompose the rotation matrix $R_r$ into Euler angles by exploiting its
structure, in the following way (see Appendix A for details):

\[
\psi_r = \arctan 2 (R_r(2, 1), R_r(2, 2)) ,
\]
\[
\theta_r = \arctan 2 \left( R(2, 3), \sqrt{R(2, 1)^2 + R(2, 2)^2} \right) ,
\]
\[
\phi_r = \arctan 2 (-R_r(1, 3), R_r(3, 3)) .
\]

(2.8)

We describe the relative translation vector using polar coordinates, defining \( \alpha \) to be the relative horizontal angle and \( \beta \) to be the relative vertical angle:

\[
\alpha = \arctan 2 \left( t_r(x), t_r(y) \right) ,
\]
\[
\beta = \arcsin \left( \frac{t_r(z)}{\|t_r\|} \right) .
\]

(2.9)

This representation naturally preserves the unit norm property of the translation vector.

Finally, we define the vector of unknowns \( s \) to be composed of these five angles, ignoring the distance between the cameras, which is not represented by the epipolar geometry:

\[
s \triangleq \left( \psi_r \ \theta_r \ \phi_r \ \alpha \ \beta \right)^T .
\]

(2.10)

In general, the intrinsic calibration parameters are also needed, but in the context of this work they are usually known with sufficient accuracy. The input vector \( s_0 \) is built from the prior values. Note that the relative pose is asymmetric. Therefore, matching image 2 to image 1 is not identical to matching image 1 to image 2.

The covariance matrix of \( s_0 \) has a central role in SOREPP. We assume that the \( 6 \times 6 \) global pose covariance matrices of each camera are known and denote them as \( \Sigma_1 \) and \( \Sigma_2 \). One can approximate these matrices by creating diagonal matrices from the noise approximated in 4.2. Under the independent noise assumption, the \( 5 \times 5 \) relative pose covariance matrix \( \Sigma_{s_0} \) can be linearly approximated from the cameras’ covariance matrices:

\[
\Sigma_{s_0} \approx J_1 \Sigma_1 J_1^T + J_2 \Sigma_2 J_2^T .
\]

(2.11)

More details about the approximated noise levels of smartphones and IMUs, the noise propagation, and the relative pose covariance matrix \( \Sigma_{s_0} \), are given in Chapter 4.
Chapter 3

Related Work

3.1 Introduction

This section reviews the state-of-the-art methods for epipolar geometry estimation, dividing them to three families: methods which are based on image data only, RANSAC methods that exploit pose priors, and bundle adjustment.

3.2 Image Based Only Methods

3.2.1 RANSAC

Originally introduced in 1981 [FB81], RANSAC (RANdom Sample Consensus) is the basis for most modern robust algorithms for epipolar geometry estimation, and for many other geometric computer vision problems. RANSAC applies a general framework for model fitting where the data is contaminated with outliers. In RANSAC, the measurements space is sampled, searching for a valid (i.e., outlier-free) minimal set. From each sampled set, a model is estimated using a model generation method. Then, this model is verified against all the measurements, checking how much do the measurements “support” this model, for example lie in a distance smaller than a threshold from a model-based calculated “measurement”. RANSAC searches for the model which optimizes this target function, thus gets the highest support. Since RANSAC is random in nature, it returns the best model up to some probability. Its runtime, i.e., the number of iterations of sampling a minimal set, depends on the estimated amount of inliers in the data: as more inliers exist, the less samples are needed in order to ensure (with high enough probability) that a valid set has been sampled.

How many iteration are needed? Assuming that the inlier fraction is $\epsilon$, then the probability to sample an inlier is $\epsilon$. For a set size of $|\Omega_{set}|$, the probability to sample an outlier-free set is approximately $\epsilon^{|\Omega_{set}|}$, and the probability to sample an invalid set for $N$ iterations is $(1 - \epsilon^{|\Omega_{set}|})^N$. In order to ensure that the probability to sample a valid set is above $conf$, we need to demand that the probability to sample an invalid
set will decrease below $1 - \text{conf}$. Comparing these two conditions, we get:

$$N \geq \frac{\log (1 - \text{conf})}{\log (1 - \epsilon |\Omega_{\text{set}}|)}.$$  \hspace{1cm} (3.1)

Therefore it is clear why the minimal set size sampling is preferred: the smaller is $|\Omega_{\text{set}}|$, the less iterations are needed. For example, if the minimal set size is five measurements, and the measurements contain only 10% inliers, about 300,000 iterations are needed in order to ensure in a confidence of 95% that a valid set was sampled.

For epipolar geometry, the measurements are corresponding features on both images, and the model is the fundamental or the essential matrix. The eight-point [LHS1] or seven-point [TZ00] algorithms are used as model generation methods for uncalibrated cameras, and the five point algorithm such as [Nis04] is used in the calibrated case.

The basic RANSAC has some limitations. First, the assumption that sampling a valid minimal size set leads to accurate model is incorrect. Due to noise and to the measurements distribution in the measurements space, the estimated model might be inaccurate, therefore might not get the highest consensus. Second, the number of required iterations might be too big to be practical, therefore accelerations are needed. Modern algorithms tackle these issues, the most important in the context of this work are now reviewed.

**MLESAC and Guided MLESAC**

MLESAC [TZ00] stands for Maximum Likelihood Estimate for SAmpling Consensus. In MLESAC, the simple target function of inliers number is replaced by a probabilistic function. MLESAC tries to maximize the log likelihood of the residual errors of the putative correspondences, modeling it as a combination of two distributions, a normal distribution for the inliers and a uniform distribution for the outliers. The probability of each putative correspondence being an inlier is estimated by Expectation Maximization. Guided MLESAC [TM05] uses prior probabilities for each putative correspondence being an inlier instead of the Expectation Maximization step.

**PROSAC**

PROSAC [CM05] (PROgressive ranSAC) tries to accelerate RANSAC. It uses priors on the quality of the putative correspondences in order to guide the sampling and increase the chance to sample a valid set. It does so progressively, by separating the set of correspondences which are used to validate the hypothesized models, from the set of correspondences which are used to generate those models. While the verification set always contains all the given correspondences, the generation set is progressively expanded during the estimation. At the beginning, this set is very small, being assembled only from the top-ranked correspondences. Then a few RANSAC iterations are performed on this small set, searching to a good model. If no satisfying solution has been found,
the generation set is increased by adding lower ranked correspondences, then more RANSAC iterations are performed. After enough iterations the generation set would contain all the correspondences, meaning that the algorithm has degenerated to the standard RANSAC.

**LO-RANSAC**

LO-RANSAC [CMK03] is Locally Optimized RANSAC. It tackles the wrong assumption that sampling a valid minimal set leads to accurate model. It improves the so-far-the-best model by sampling bigger sets from the inliers of the model. Since it optimizes only the best model, the complicated optimization function is only rarely performed. The convergence is faster since less valid sets are missed up, therefore the overall runtime improves, in compare to RANSAC.

**USAC**

Recently USAC [RCP+13] (Universal ranSAC) was introduced. It integrates PROSAC and LO-RANSAC together with the SPRT [MC05] accelerated hypotheses test. SPRT test is based of the likelihood ratio test of the measurements being consistent with a bad and good models. If this ratio is too high, the model is rejected. This way bad models can be rejected even before all the correspondences are verified.

**BEEM**

BEEM [GS08] stands for Balanced Exploration and Exploitation Model. It tackles both the accuracy and the efficiency issues of RANSAC. The accuracy is improved by performing local optimization for promising models. The efficiency is mainly improved by using a minimal set size of only two correspondences, exploiting the fact that similarity invariant features such as the SIFT return, beside the feature location, also orientation and scale. Using that additional data, synthetic points are created, completing the number of measurements needed to estimate epipolar geometry. Additionally, priors on the quality of the putative correspondences are also used in order to guide the sampling.

**BLOGS**

BLOGS [BS09] (Balanced Local and Global Search) seeks to minimize a soft threshold cost function, and combines global search that is based on Joint Feature Distribution, and local search in order to further improve the current estimation.

### 3.2.2 Non-RANSAC Algorithms

Another category of algorithms are the iterative methods [AS03], which search for the solution in the parameter space. These methods minimize a cost function which measures the distance between points and their corresponding epipolar lines. Usually
these methods work in the fundamental matrix domain, but methods that work in
the relative pose domain also exist [HK07, DCP1I11]. In [HK07] a branch and bound
strategy is used in order to find the optimal solution. The main drawback of this method
is its computational cost, making it impractical for real time applications. In [DCPII11]
an entropy-based cost function is used to achieve robustness to outliers. This method is
built for vision-based robot navigation, where the images being matched are successive
video frames. Thus, they are similar and yield a high fraction of inliers. The assumption
of high inlier fractions is made in the initialization step of the optimization scheme.

3.3 Model Generation Methods that Exploit Pose Priors

Much work has been done on exploiting pose priors, in particular in the robotic and
navigation communities. Some papers integrate vision and an IMU in order to reduce
drift [TGL+10], use pose priors in order to choose images [IHBK11], or to constrain
a structure-from-motion solution [CKD06]. However, not much work has been done
regarding exploiting pose prior data in order to enhance the epipolar geometry estimation
itself. The existing methods are based on a RANSAC framework, and are inspired by
the existence of an IMU attached to the camera.

3.3.1 Limited Correspondences Search Area

The method described in [WHBS10] uses IMU measurements in order to limit the
correspondence search area to be only at a user defined distance around the corresponding
epipolar line. This accelerates the computation and reduces the number of incorrect
matches. The method works well in low noise scenarios. However, as the noise level
increases, so does the width of the region around the epipolar line where the correct
correspondence lies, covering large portions of the image. This makes filtering impractical
in noisy conditions.

3.3.2 Known Rotations

Other methods [FTP10, NZG+12] suggest using the known gravity vector, which is
equivalent to the two rotation angles pitch $\theta_r$ and roll $\phi_r$, reducing the degrees of freedom
of the solution from five unknowns to only three. They exploit the representation of the
rotation matrix from three Euler angles as shown in (2.6) in order to rotate a-priori
the points on one of the images. Thus, the epipolar constraint (2.1) and (2.2) gets the
following formulation:

$$
p_2^T F p_1 = p_2^T (K_2^{-1})^T (R_r[tr])_x K_1^{-1} p_1 =
$$

$$
p_2^T (K_2^{-1})^T (R_y(\phi_r) R_x(\theta_r) R_z(\psi_r)[tr])_x K_1^{-1} p_1 = 0
$$
Finally we get:

\[ p_2^T (K_{2}^{-1})^T R_y(\phi_r) R_z(\theta_r) \left( R_z(\psi)[t_r]_x \right) K_{1}^{-1} p_1 = 0. \]  

(3.2)

Therefore the essential matrix gets a simpler structure containing only three degrees of freedom: \( E = R_z(\psi)[t_r]_x \). RANSAC is applied with a correspondence set of size three. Reducing the minimal set size causes a rapid decrease in the number of iterations required by RANSAC, thus accelerating the runtime.

A similar method which assumes that the full rotation is known and uses a minimal set of two correspondences is described in [KCS11].

The main limitation of these methods is that they take the values of the measured orientation angles as hard constraints, and are therefore sensitive to noise in them. Moreover, any existing position priors might be ignored.

### 3.4 Bundle Adjustment

Bundle adjustment (BA) is the gold standard method for estimating camera poses and the scene from correspondences. It is based on optimizing a cost function that is built from reprojection errors, and is the Maximal A-Posteriori (MAP) estimator for the unknowns in case of normally distributed i.i.d. noise. Let us develop the MAP estimator for the unknowns vector \( s \):

\[ \hat{s} = \arg \max_s \{ \text{prob}(s|\text{measurements}) \}. \]  

(3.3)

Using Bayes’s theorem:

\[ \text{prob}(s|\text{measurements}) = \frac{\text{prob}(\text{measurements}|s)\text{prob}(s)}{\text{prob}(\text{measurements})}. \]  

(3.4)

Since we only want to optimize (3.3), we can neglect the denominator. Assuming statistically independent measurements, we get for a set \( \Omega_{\text{all}} \) of correspondences:

\[ \hat{s} = \arg \max_s \{ \text{prob}(s|\text{measurements}) \} = \]

\[ \arg \max_s \{ \text{prob}(\text{measurements}|s)\text{prob}(s) \} = \]

\[ \arg \max_s \left\{ \text{prob}(s) \prod_{k \in \Omega_{\text{all}}} \text{prob}(\text{measurements}(k)|s) \right\} = \]

\[ \arg \max_s \left\{ \log(\text{prob}(s)) + \sum_{k \in \Omega_{\text{all}}} \log(\text{prob}(\text{measurements}(k)|s)) \right\} = \]

\[ \arg \min_s \left\{ -\log(\text{prob}(s)) - \sum_{k \in \Omega_{\text{all}}} \log(\text{prob}(\text{measurements}(k)|s)) \right\}. \]
Assuming that the feature noise is i.i.d., normally distributed with a standard deviation $\sigma$ and that the model reprojection of the $k$-th correspondence is given by $f(k)$:

$$prob(measurements(k)|s) = const \cdot \exp \left( \frac{- (measurement(k) - f(k))^2}{2\sigma^2} \right).$$

Then:

$$\hat{s} = \arg \min_s \left\{ -\log(prob(s)) - const \cdot \sum_{k \in \Omega_{\text{all}}} \log \left( \exp \left( \frac{- (measurement(k) - f(k))^2}{2\sigma^2} \right) \right) \right\}.$$ 

Finally we get

$$\hat{s} = \arg \min_s \left\{ \frac{-const \cdot \log(prob(s))}{\sum_{k \in \Omega_{\text{all}}} (measurement(k) - f(k))^2} \right\}. \quad (3.5)$$

Bundle adjustment has many variants, the important ones in the context of the paper are variants that apply robustified error models, use epipolar geometry-based distances instead of reprojection errors, “structure-less” variants which estimate only the camera poses, and enforcement of prior constraints [TMHF00, IRBD12, TGBG11]. BA is commonly used as a final step, after most of the outliers have been removed and good values for the camera poses have to be given to initialize the process [HZ04, SSS06].

The core of SOREPP, the proposed algorithm, may be considered as a BA-variant, designed solely for two images, which is structure-less, based on a robustified cost function of epipolar constraints, applies pose prior regularization, and initialized by it. However, the general framework of SOREPP expands the BA framework and significantly improves the performance relative to the core optimization scheme. This, together with the robustness of the function to be optimized, enables to apply SOREPP on the raw input, without removing first the outliers nor improving the noisy pose priors.
Chapter 4

Relative Pose Measurements

4.1 Introduction

This section describes how relative pose is measured in smartphones and in IMUs, the physics behind it, and the estimated errors. It is based on Kayton and Fried’s book [KF97].

4.2 Pose Measurements in Smartphones

We use East-North-Up (ENU) coordinates to define the position of each platform, as illustrated in Figure 2.2. The orientation is described by three Euler angles $\psi$, $\theta$, and $\phi$. See Chapter 2.2 for more details.

A typical smartphone has several sensors used to determine these six pose parameters with the following qualitative accuracy:

The Position is usually calculated using a GPS receiver. Its typical accuracy under good reception conditions can be accurate to within a few meters. Under difficult GPS signal visibility conditions, such as in a dense urban environment, the errors may become significantly larger. Because the errors are affected by environmental conditions, they will usually be partially correlated for images taken in the same area and at the same time. Another way to roughly approximate position is by triangulation, using range estimation to cell phone towers. In the paper we ignore any possible dependency between the position noise of the two cameras and assume that any such noise is independent.

The Leveling Angles, also known as pitch and roll angles, define the platform’s orientation relative to the tangent plane of the earth at its current position. They can be easily calculated using accelerometers measuring the gravity vector. The force of gravity ensures quite accurate results in the order of $1^\circ$ even for cheap, insensitive accelerometers such as the ones used in smartphones. The calculation can be done using
the following equations:

\[
\begin{align*}
\theta &= \arctan 2(a_y', a_x'), \\
\phi &= \arctan 2 \left( a_x', -\sqrt{a_y'^2 + a_z'^2} \right). \quad (4.1)
\end{align*}
\]

where \(a_x', a_y', a_z'\) are the measured forces on the different axis of the platform. Note that these axis are rotated relative to the global coordinates.

The Azimuth is currently the most challenging parameter to measure. It is possible to calculate it by measuring the angular velocity vector of the earth, but very sensitive gyros are required. Azimuth is usually measured by means of a compass but this method has several drawbacks: The first is that the magnetic north is not the true north, but this is easily compensated for. A more severe problem is inaccuracy: the magnetic field itself is affected by high-tension electric wires and metals in its vicinity. In our experiments we experienced errors exceeding 7°.

It should be noted that instruments for pose measurement in smartphones are naturally built for static platforms. In the dynamic case, accelerometer measurements will be biased and no longer measure the gravity vector properly. However, in dynamic systems this problem is avoided because orientation is usually measured using built-in gyros as part of an IMU.

We used SitisMobile’s “GeoCam” application [Sit12] in order to record images with the attached pose. The measured pose is given in the same coordinate system as the image.

### 4.3 Pose Noise Propagation

The covariance matrix of the relative pose plays a central role in SOREPP, and thus we describe it here in detail. The influence of pose errors on the Sampson distances is also introduced.

The relative pose of two smartphone images is composed of statistically independent measurements associated with each of the two images and given in global coordinates. We assume that the \(6 \times 6\) global pose covariance matrices of each camera are known and denote them as \(\Sigma_1\) and \(\Sigma_2\). One can approximate these matrices by creating diagonal matrices from the squared noise approximated in Chapter 4.2. Under the independence assumption, the \(5 \times 5\) relative pose covariance matrix \(\Sigma_{s0}\) can be linearly approximated from the cameras’ covariance matrices (2.11):

\[
\Sigma_{s0} \approx J_1 \Sigma_1 J_1^T + J_2 \Sigma_2 J_2^T,
\]

where \(J_1, J_2\) are the Jacobians, composed of the derivatives of the relative pose angles.
by each camera’s pose parameters:

\[
J_i = \begin{bmatrix}
\frac{\partial \psi_r}{\partial \psi_i} & \frac{\partial \psi_r}{\partial \theta_i} & \frac{\partial \psi_r}{\partial \phi_i} & \frac{\partial \psi_r}{\partial x_i} & \frac{\partial \psi_r}{\partial y_i} & \frac{\partial \psi_r}{\partial z_i} \\
\frac{\partial \theta_r}{\partial \psi_i} & \frac{\partial \theta_r}{\partial \theta_i} & \frac{\partial \theta_r}{\partial \phi_i} & \frac{\partial \theta_r}{\partial x_i} & \frac{\partial \theta_r}{\partial y_i} & \frac{\partial \theta_r}{\partial z_i} \\
\frac{\partial \phi_r}{\partial \psi_i} & \frac{\partial \phi_r}{\partial \theta_i} & \frac{\partial \phi_r}{\partial \phi_i} & \frac{\partial \phi_r}{\partial x_i} & \frac{\partial \phi_r}{\partial y_i} & \frac{\partial \phi_r}{\partial z_i} \\
\frac{\partial x_r}{\partial \psi_i} & \frac{\partial x_r}{\partial \theta_i} & \frac{\partial x_r}{\partial \phi_i} & \frac{\partial x_r}{\partial x_i} & \frac{\partial x_r}{\partial y_i} & \frac{\partial x_r}{\partial z_i} \\
\frac{\partial y_r}{\partial \psi_i} & \frac{\partial y_r}{\partial \theta_i} & \frac{\partial y_r}{\partial \phi_i} & \frac{\partial y_r}{\partial x_i} & \frac{\partial y_r}{\partial y_i} & \frac{\partial y_r}{\partial z_i} \\
\frac{\partial z_r}{\partial \psi_i} & \frac{\partial z_r}{\partial \theta_i} & \frac{\partial z_r}{\partial \phi_i} & \frac{\partial z_r}{\partial x_i} & \frac{\partial z_r}{\partial y_i} & \frac{\partial z_r}{\partial z_i}
\end{bmatrix},
\]

where \( i = 1, 2 \) is the camera’s index. Although \( s_0 \) is not distributed normally but may have a complex multimodal distribution, we assume that the resulting covariance represents the uncertainty of the given relative pose. It is used to bound the solution in the relative pose space and for the Mahalanobis distance regularization term in (5.4). The relative covariance matrix also reveals the dependency between the pose parameters. For example, uncertainty in the azimuth in image 1 increases the uncertainty in \( \alpha \). We define \( \sigma_{s_0} \) as the vector of standard deviations of \( s_0 \), which is the square root of the diagonal of \( \Sigma_{s_0} \).

To gain some intuition about the relative pose errors, consider the following numerical examples. First, we assume that both cameras are leveled, facing north and located side by side at a distance of 100 meters. We assume that the global pose errors of the cameras have the following standard deviations: \( \sigma_\psi = 5^\circ \), \( \sigma_\theta \) and \( \sigma_\phi \) are \( 1^\circ \), and the position errors are 5 meters in each axis. We also assume that all the errors are uncorrelated. The resulting \( \sigma_{s_0} \) is as follows:

\[
\sigma_{s_0} = \begin{bmatrix}
\sigma_\psi_r & \sigma_\theta_r & \sigma_\phi_r & \sigma_\alpha & \sigma_\beta
\end{bmatrix}^T = \begin{bmatrix}
7.1^\circ & 1.4^\circ & 1.4^\circ & 6.4^\circ & 4.2^\circ
\end{bmatrix}^T.
\]

While it is easy to understand the values for \( \psi_r, \theta_r \) and \( \phi_r \) as the sum of variances of independent noises, the values of \( \alpha \) and \( \beta \) are more complicated and depend on the relative pose, the standard deviation values of the position of each camera, the distance between the cameras and the standard deviation values of the orientation of camera 1. If we reduce the distance between the cameras to be only 5 meters we get the following \( \sigma_{s_0} \) values:

\[
\sigma_{s_0} = \begin{bmatrix}
7.1^\circ & 1.4^\circ & 1.4^\circ & 81.2^\circ & 81^\circ
\end{bmatrix}^T.
\]

The values for the relative orientation are unchanged, but the uncertainty of the translation vector angles increases rapidly, as expected.

The way relative pose errors influence the Sampson distances is complicated. The standard deviation of the Sampson distance can be linearly approximated according to the relative pose covariance matrix. Using (2.5), the standard deviation of the \( k \)-th correspondence is approximated as follows:

\[
\sigma_{d(k)} \approx \sqrt{J(k)\Sigma_{s_0}J(k)^T},
\]
Figure 4.1: Samson errors in simulated analysis of a typical smartphone camera, FOV of $59.6^\circ \times 46.3^\circ$. Histogram of simulated Sampson distances, and calculated Normal and Laplacian distributions based on (4.3). See text for details. (a): Side by side, 100m distance. (b): Side by side, 5m distance. (c): One in front of the other, 100m distance. (d): One in front of the other, 5m distance.

where $J(k)$ is the Jacobian of the Sampson distance with respect to the relative pose $s_0$: 

\[
J(k) = \left( \frac{\partial d(k)}{\partial \psi} \quad \frac{\partial d(k)}{\partial r} \quad \frac{\partial d(k)}{\partial \theta} \quad \frac{\partial d(k)}{\partial \phi} \quad \frac{\partial d(k)}{\partial \alpha} \quad \frac{\partial d(k)}{\partial \beta} \right). \tag{4.4}
\]

For a typical smartphone such as Samsung Galaxy SII, the camera resolution is $2048 \times 1536$ pixels and the field of view (FOV) is $59.6^\circ \times 46.3^\circ$. Then the Sampson standard deviation values for the examples given before would be 40 and 190 pixels respectively. For cameras located one in front of the other or with smaller fields-of-view, the errors would be larger. For cameras located in front of the other, the standard deviation of the Sampson distance would be 100 and 190 pixels for the former examples. For a field-of-view of $10^\circ \times 7.5^\circ$, the standard deviation of the Sampson distance would be 230 and 1100 pixels for the former examples, cameras located side by side. Figure 4.1 shows a simulation of some of these cases, with Normal and Laplacian distributions that are calculated by the approximated standard deviation using (4.3). For the Laplacian distribution, which fits better the simulated Sampson distances for close cameras, we used the fact that assuming zero mean, the value of the remaining parameter is equal
to $\sigma_{d(k)}/\sqrt{2}$. Errors of that magnitude demonstrate why calculating $F$ directly from the pose priors is useless and an image-based epipolar geometry estimation algorithm is required to produce reasonable results.

To summarize, the relative pose and its uncertainty are calculated from the input global pose estimates of both cameras, and this representation describes well the physical constraints. We consider the noise to be statistically independent. Noise independence represents the more severe case. In practice sometimes this is not the case, and the relative pose noise is actually smaller than expected under the statistically independent assumption.

4.4 Pose Measurements and IMU Errors

An IMU is composed of three gyros and three accelerometers measuring the angular velocity and the acceleration in each of the axes, respectively. The relative pose is calculated by a numeric integration scheme called “strapdown” that takes into account the fact that the measurements are related to an inertial coordinate system but given in continuously changing, body-attached coordinates. Because the pose measurements consist of integration steps, noise is summed up and the pose estimate drifts over time.

The way an IMU measures pose is approximately described as follows:

**The Orientation** is calculated by one integration step. Therefore, for two frames in a sequence, the initial conditions are compensated for and the relative orientation error is composed from the gyro errors only. The main gyro noise is drift, which is considered constant. For low quality gyros, the drift is in the order of $100^\circ$/hour, or $0.03^\circ$/sec. Therefore, for two frames taken in rapid succession, the expected orientation error in all three axes is small and expected to be in the order of one-tenth of a degree or even lower.

**The Position** is calculated by two integration steps. Therefore the initial velocity is not compensated for and has a large effect on the calculated position. For example, for a platform traveling at a constant velocity, the accelerometer readout will be zero up to some noise, while the true translation depends on the initial velocity only and cannot be measured by the IMU at all. Therefore, for an IMU, the relative position may be completely incorrect even for two successive frames.

The fact that the pose measurements consist of integration steps makes the approximation of $\Sigma_{s0}$ easy, and is naturally given in relative coordinates. The covariance matrix is assumed to be diagonal. The errors in each of the five angles are approximated as independent of the others. The standard deviation of the three orientation angles can be approximated by the expected noise ($drift \cdot dt$), where $dt$ is the time interval between the two frames. The standard deviation of the translation vector angles should get infinite values, representing the fact that the translation vector is considered unknown.

One technical issue which has to be dealt with in the IMU case is the necessity to align the IMU to the camera axis. This issue is ignored in the paper. There are existing
toolboxes such as [LD07] that can be used for such alignment.

4.5 Summary

We now briefly summarize the topic of relative pose measurements. The estimation of $s_0$ and its covariance matrix $\Sigma_{s_0}$ are, along with the putative image correspondences, the main input for SOREPP. Similar derivations may be performed for applications other than smartphones and IMUs in a similar manner.
Chapter 5

SOREPP

5.1 Introduction

This chapter describes our proposed algorithm, called “SOREPP” (Soft Optimization method for Robust Estimation based on Pose Priors). First, the main concept is presented. Second, a deeper analysis of the behavior of SOREPP is given, in order to explain why it should work and what is the importance of the different parts of the target function to be minimized. Third and last, additional components are introduced, which expand the basic bundle adjustment framework and enable it to overcome significant pose priors noise and even missing data.

5.2 The Concept

SOREPP uses pose priors to focus the search in the relevant parameter space by initializing a standard optimization scheme. Thus, global search is avoided. The cost function being optimized is a differential and robust version of the standard, number-of-inliers cost function used by RANSAC. It works on all the putative correspondences, without detecting the inliers explicitly, which might be difficult in low inlier fraction cases. As a result, SOREPP is able to handle low inlier fractions. To do so, we have to define a robust cost function that can handle low inlier fractions while having a large basin of attraction to handle large pose errors.

SOREPP is given as input an estimate of the relative pose $s_0$ of the two cameras, and its uncertainty is described by a covariance matrix $\Sigma_{s_0}$. Chapter 4 explains how we can estimate $\Sigma_{s_0}$ for smartphones and for IMU-attached cameras. A set of putative correspondences $\Omega_{all}$ is given, together with a weight of each correspondence $w$, which is a rough approximation of the probability of being an inlier. Two of the many options that might be used to define such weights are described in Section 5.4.1.

We define the following Gaussian score for every putative correspondence $k$:

$$g(k) = \exp\left(\frac{-d(k)^2}{2\sigma_h^2}\right).$$

(5.1)
This function is a type of an M-estimator [Hub11], other types may be used as well. It applies a soft threshold with a parameter $\sigma_h$ on the Sampson distance associated with each correspondence. The value of $\sigma_h$ should be proportional to the feature localization noise, with some reserves needed to overcome inaccurate models. For the true relative pose $s_{true}$, inliers would receive high scores, i.e., close to 1, while outliers far away from the model would receive low scores, i.e., close to 0. These scores determine the preliminary target function that SOREPP will seek to minimize:

$$v = \sum_{k \in \Omega_{all}} \tilde{w}(k)(1 - g(k)),$$

(5.2)

where $\tilde{w}(k)$ are normalized weights:

$$\tilde{w}(k) = \frac{w(k)}{\sum_{k \in \Omega_{all}} w(\tilde{k})},$$

(5.3)

and $v$ is the score of a specific relative pose $s$ over all the putative correspondences and is also in the range $[0, 1]$. Due to the non-convexity of (5.2), optimizing this function may be difficult in noisy conditions. In order to regularize the solution, we define a measure of the proximity of the solution to the initial pose by the following Mahalanobis distance:

$$\lambda(s) = \frac{1}{|s|} \sqrt{\delta s^T \Sigma_{s_0}^{-1} \delta s},$$

(5.4)

where $|s|$ is the size of $s$ and $\delta s \triangleq s - s_0$. We add this regularization term to (5.2) in order to define the following minimization problem:

$$\hat{s} = \arg \min_s \left\{ c \left( \sum_{k \in \Omega_{all}} \tilde{w}(k)(1 - g(k)) \right) + (\lambda(s))^2 \right\},$$

(5.5)

where $c$ is a weighting parameter, that is fixed to the value of 3. This value constitutes a good balance between the measurements and the priors. This equation combines the objective to minimize the Sampson distances for as many putative correspondences as possible while keeping the solution close to the pose priors. The minimization can be done using any standard optimization method. In our implementation we use Levenberg-Marquardt [Lev44]. Once the algorithm converges, a hard threshold is applied to select the inliers. They serve only as an output of the algorithm.

This simple optimization procedure is the heart of SOREPP algorithm. The algorithm is simple and fast, as well as robust to low inlier fractions and significant pose noise. The next sections delve deeper into the algorithm, explaining how SOREPP works, and how it behaves under different conditions; additional components are also introduced.
5.3 Understanding SOREPP

The core principle of SOREPP is to avoid global search since pose prior data is available. SOREPP is based on two foundations: The robust M-estimator cost function (5.1), and the approximated relative pose uncertainty as expressed by $\Sigma_{s0}$. This section explains why such a simple scheme works well and handles low inlier fractions as well as pose noise.

5.3.1 Core Principle

Algorithms from the RANSAC family perform a global search and try to maximize a main target function, usually the number of inliers, in two steps: first, they randomly sample a minimal set of putative correspondences, searching for an outlier-free set. A model is estimated from each minimal set and ranked by the number of correspondences that support it. Then the main function is optimized by choosing the best model.

The existence of pose priors enables SOREPP to skip the global search over the whole parameter space and directly optimize the main target function after changing it to the robust, continuous and differentiable form of (5.2). Later in section 5.4.2 we will expand SOREPP to perform a focused search in the vicinity of the pose priors, but this point is ignored for now.

5.3.2 The Robust Cost Function

SOREPP is based on soft thresholding, applied by (5.1). Soft thresholding makes it possible for inliers to support the estimation even if they lie far beyond the threshold. This property is a main advantage of soft thresholding, and is the key to avoiding explicit inlier detection. Thus, it is also the key for dealing with low inlier fractions. However, since the vector of unknowns $s$ is estimated over all the putative correspondences simultaneously, what controls the performance of the algorithm is the size of the basin of attraction of our target function in the presence of outliers. If it is too small, then our method would not be practical and would be sensitive to pose noise and outliers.

This problem is illustrated by the toy example in Figure 5.1 where the true line has to be estimated, and it is assumed that a differential mathematical model connects the prior line and the points by the distances between them. The gradients of the robust scores $g$ (5.1) are plotted, including both inliers and outliers since no explicit inlier detection is applied. The gradients of the outliers, even when they are strong due to their proximity to the prior line, are scattered and typically cancel each other out. On the other hand, the inliers consistently pull the solution in the same direction, even if each gradient is relatively weak. This phenomenon is even stronger in our case than in the two-dimensional toy example because in the higher dimensional space, the gradients of the outliers are more widely scattered. This phenomenon is the main reason why SOREPP works in practice and can handle significant pose noise as well as low inlier...
Other M-estimators might be used instead of the Gaussian scores of (5.1). One promising function is Laplacian score, since the Laplacian distribution fits better the simulated Sampson distances as shown in Chapter 4.3, and it decays slower. However, experiments on other M-estimators such as Laplacian or Geman-McClure reveals that the choice of a specific M-estimator almost do not change the results. One possible explanation is that since most of the putative correspondences are outliers, the use of a function that better suits the inlier distribution has only minor effect.

5.3.3 The Regularization Term

Adding the regularization term as in (5.5) stabilizes the solution and constrains it to be in the vicinity of the pose priors. The vicinity is automatically adapted by the approximated relative covariance matrix $\Sigma_{s0}$. The fact that $\Sigma_{s0}$ describes the uncertainty in $s_0$ plays a major role here.

Consider the case in which the two cameras are close to each other. In this important case the distance between the cameras is in the same order of magnitude as the expected global position errors or even smaller. It that case, the uncertainty in the translation vector angles $\alpha, \beta$ increases, as shown in Chapter 4.3. This reduces the regularization constraint on these angles, up to eliminating the regularization weight on them completely. Then, the estimation of $\alpha$ and $\beta$ would come only from the image correspondences. This does not cause a problem, since close cameras yield similar images, resulting in high inlier fractions.
The close cameras case reveals a desired, inherent property of SOREPP: Inlier fractions and pose prior robustness are traded off. The balance is achieved through $\Sigma_{s0}$ and depends on the distance between the cameras.

For images from closely placed cameras, the inlier fraction is high. The regularization constraints on the translation vector angles are weak, allowing these angles to get estimated values far away from the priors but without a high penalty in the cost function value. Then noisy pose data or even a completely incorrect translation vector will have only a minor effect on the accuracy of the results. This is the case of robot navigation, where the pose priors are typically given by an IMU and the images being matched are typically successive in time and therefore similar. It is also a common case for smartphones.

For cameras placed farther from each other, the inlier fraction drops, and the regularization constraints on the translation vector angles are stronger. Then better pose accuracy is required. This is the case of smartphone images. For low inlier fractions of 10%, smartphones are accurate enough for estimation by SOREPP.

It is important to mention that the regularization term contributes to SOREPP’s performance but is not essential to it. This is demonstrated in experiments that will be described in greater detail in Chapter 6.3.2. Still, even when the importance of the regularization term decreases, $\Sigma_{s0}$ maintains its importance, since it is used also to define the parameter search area for the initial guesses, as will be explained in Section 5.4.2.

5.3.4 Other Pose Prior Sources

Although the paper focuses on pose priors given by smartphones or IMUs, SOREPP is not limited to them. Any pose priors given with a proper covariance matrix can be used for estimation, as long as the relation between the pose noise and the inlier fraction is in the virtual working envelope of SOREPP.

5.4 SOREPP Components

This section describes several additional components of the basic SOREPP algorithm presented in Section 5.2. It is briefly described in Algorithm 5.1.

5.4.1 Correspondence Weights

The weights $w(k)$ are one of the inputs to SOREPP. SOREPP can use any probability-like values. Our basic weights are built from the following simple formula based on the SIFT ratio test [Low04]:

$$w(k) = 1 - \frac{d_{1NN}^2(k)}{d_{2NN}^2(k)},$$

where $d_{1NN}(k), d_{2NN}(k)$ are the distances from the first and second nearest neighbors of the $k$-th correspondence.

31
Algorithm 5.1 SOREPP algorithm

1: Apply Pose Region Search: Find up to \( m \) promising initial states.
2: for (\( m \) initial states and the input one \( s_0 \)) do
3: Estimate the relative pose by optimizing (5.5)
4: Evaluate the solution according to (5.2)
5: if (The solution is good enough) then
6: Apply Solution Refinement: Reduce \( \sigma_h \) and re-estimate
7: end if
8: end for
9: Choose the best model
10: Calculate the grade

Figure 5.2: Illustration of the Pose Region Search component. The energy is calculated on a sparse grid based on the input pose and the expected noise levels. For a few local minima (shown in black circles) the optimization process is applied. In the figure, only three of the five relative pose parameters are shown.

A different way to define putative correspondences and weights is to follow BLOGS. Similarity is calculated by the inner product of the descriptor vectors. Putative correspondences are the feature pairs that get the highest similarity score in both directions of matching: A feature from image 1 is compared to all features in image 2, and vice versa. The weights are calculated by the following formula [BS09]:

\[
w(k) = \left(1 - \exp^{-q(k)}\right)^2 \left(1 - \frac{q_r(k)}{q(k)}\right) \left(1 - \frac{q_c(k)}{q(k)}\right),
\]

where \( q(k) \) is the highest score for the \( k \)-th correspondence, \( q_r(k) \) is the second highest score in one direction of matching and \( q_c(k) \) is the second highest score in the other direction. The first RHS of (5.7), i.e, \( (1 - \exp^{-q(k)})^2 \), is the similarity based component.
5.4.2 Pose Region Search

In order to further increase the robustness to noisy pose measurements, the parameter space around the input pose priors is searched. This is done by evaluating the energy of (5.2) at a sample pattern, where each relative pose parameter is sampled at several values around the prior value. The result is a small five-dimensional hypercube with target function values at each entry. In that hypercube local peaks are found, creating a set of initial candidate guesses for the optimization scheme. The optimization process of (5.5) is performed for up to the $m$ best candidates. The user defined parameter $m$ controls the number of candidates as well as the input pose and is usually in the range of 0 to 5, where $m = 0$ means starting only at the pose prior $s_0$. In all these cases, $\delta s$ is always defined relative to $s_0$. An illustration of this component is shown in Figure 5.2. The samples are spread uniformly over the space determined by $2\sigma s_0$ for every angle. The samples include only 3 values for $\theta_r$, $\phi_r$, and $\beta$, 5 values for $\alpha$, and 7 values for $\psi_r$. If the uncertainty in $\alpha$ becomes too big, for example if the cameras are close to each other or when no GPS data is available, the sampling scheme is changed, but in a way that the overall number of samples remains constant: $\theta_r$ and $\phi_r$ are not sampled (only the prior values are used), allowing $\alpha$ to be sampled at 15 values and $\beta$ at 9 samples, covering uniformly the whole unit sphere of 360° and 180° respectively.

Even though this sampling method is very simple and sparse, it yields good results, because one of the samples usually falls into the large basin of attraction of (5.5). Note that the computational cost is low. The cost function is evaluated 945 times, but the optimization scheme is applied only up to $m + 1$ times.

5.4.3 Solution Refinement

One of the main characteristics of SOREPP is that all the putative correspondences contribute to the solution. The main advantage of it is the ability to bypass the need to explicitly find inliers, which may be difficult when the inlier fractions are low. However, this characteristic is also a main drawback: outliers also participate in the estimation and skew it. The most harmful outliers are those which randomly obtain small Sampson distances with respect to the current solution. They are usually further than the inliers, but close enough to have significant weights $g$.

To deal with this problem we modify $\sigma_h$ by the following procedure, common for M-estimators. For a solution with energy lower than $v_{\text{threshold}}$ of (5.2), $\sigma_h$ is decreased and the solution is estimated again by optimizing (5.5), starting from the current estimation result $\hat{s}$. This way the weight of the correspondences that are close to the solution increases at the expense of the outliers, which are assumed to be further from the solution. We use the following values: $v_{\text{threshold}} = 0.65$ and reduce $\sigma_h$ by a factor of two.
5.4.4 Inverse Translation

The calculations in SOREPP are performed using the essential matrix $E$ calculated from $s$. Each such matrix represents four possible relative poses $s$, as explained in [HZ04, Chapter 9.6]. Usually only one of them falls within the pose priors’ uncertainty region. However, for close cameras it is possible that the inverse translation vector will also be a valid interpretation of $E$, in addition to the candidate $s$. Therefore, at the end of the estimation, both options are checked and the one with more intersecting points located in front of both cameras is selected. A similar check for the rotations is not necessary, since for orientation the constraint is usually tight enough so that only one option falls within the search space.

5.4.5 Solution Grade

We wish to obtain a match grade that describes the estimation’s reliability, in order to ignore incorrect relative pose solutions. Therefore a match grade with statistical significance is estimated. The grade estimation is based on kernel density estimation (KDE) analysis of the success probability as a function of the energy values $v$ of (5.2), learned on a set of image pairs that have been matched by SOREPP: $P(\text{success}|v)$. For a new image pair that has been matched, the energy $v$ is calculated according to (5.2), after which the grade is interpolated from the learned KDE function. An example of such a graph is shown in Figure 5.3.

Figure 5.3: SOREPP probability for success as function of $v$, as estimated by KDE on example results.
Chapter 6
Experiments

6.1 Introduction

The next two sections describe extensive experimental evaluations performed in various conditions. In the experiments described in this section, an analysis of the sensitivity of SOREPP to pose noise and inlier fractions based on synthetic data is described, as well as a comparison to some state-of-the-art algorithms. In the next section three large datasets consisting of 578 image pairs were used to evaluate SOREPP performance for real life difficult conditions.

The evaluation includes a comparison to state-of-the-art robust estimation algorithms. The comparison is naturally unfair, as SOREPP uses data that is unavailable to some of the other algorithms. Still, it demonstrates the advantage of using pose priors, the challenge that these datasets pose, and the importance of handling pose noise properly.

6.2 Experiments on Synthetic Data

6.2.1 Experimental Setup

Experiments on synthetic data allow to analyze the results in the pose domain, since the true pose of each camera is known exactly. It also removes any unmodeled effects such as inaccurate intrinsic camera parameters, radial distortions, correlated pose noise and more. The way the simulations have been performed is described below.

Evaluation Setup

Following a similar setup to [FTP10], the set was composed of 400 putative correspondences which were located at a depth of 50% of the scene, and the inliers were contaminated with normally distributed noise with a standard deviation of 1 pixel. The distance between the cameras was 10% of the distance to the scene and two camera configurations were used: side by side and one in front of the other. The second camera was randomly rotated around each axis in a way that the world points were still visible.
We simulated a typical smartphone camera by choosing a field of view of $59.6^\circ \times 46.3^\circ$ and resolution of $2048 \times 1536$ pixels.

We simulated two representative scenarios. In the first scenario, the cameras were located 100 meters one from the other. The pose data of each camera $i \in \{1, 2\}$ was contaminated with independent normally distributed noise, with the same standard deviations as described in 4.3:

$$\begin{pmatrix} \sigma_{\psi_i} & \sigma_{\theta_i} & \sigma_{\alpha_i} & \sigma_{y_i} & \sigma_{z_i} \end{pmatrix}^T = \begin{pmatrix} 5^\circ & 1^\circ & 1^\circ & 5 & 5 \end{pmatrix}^T. \quad (6.1)$$

In the second scenario the cameras were located 5 meters one from the other. In order to simulate indoor environment, where no position data is available, or an IMU with high errors, the position measurement of camera 2 was uniformly randomly drawn on the unit sphere around the position of camera 1. The orientation measurement noises were the same as in the first scenario.

The weights of the correspondences were randomly drawn from two distributions, given in [GS08] for inliers and outliers. They are based on kernel density estimation for SIFT ratio test values taken from real images.

The analysis was performed for increasing noise levels up to the full noise level of (6.1), and different inlier fractions. For each noise level and inlier fraction 100 trials were conducted, each with a random orientation of camera 2, random noises and random correspondence weights.

**Evaluation Method**

We evaluated the performance of the different algorithms in two ways: in the relative pose domain, by comparing the estimated relative pose to the ground truth, and in the image domain. The comparison in the relative pose domain was performed by calculating the angular difference between the estimated translation vector and orientation to the ground truth. The orientation error was defined by the smallest angle that can be used in order to bring the estimated orientation to the true one. When more than one relative pose exists, the one closest to the ground truth value is chosen. The comparison in the image domain was performed as follows. The pure inliers (without the added noise) served as control points. For each trial, we measured the distance between these control points on image 2 to the corresponding epipolar lines, which were calculated using the control points on image 1 and the estimated relative pose, and choose the maximal error as a quantitative measure. This is done in order to ensure that the estimation is accurate over most of the image. If this value was smaller than 15 pixels, we defined the estimated relative pose as “accurate” or “successful”.

36
Figure 6.1: Results on synthetic data, distant cameras, 80% inliers. Rows: (a): Side-by-side configuration, 80% inliers. (b): One-in-front-of-the-other configuration, 80% inliers. Columns: (1): Orientation errors. (2): Translation vector errors. (3): Success ratio, based on the maximal distance between the control points and the corresponding epipolar lines. All angles are given in degrees.

Implementations and Running Parameters

SOREPP was implemented in C++ using OpenCV. The implementation and a Matlab mex wrapper are available at [SOR]. For SOREPP, we set $m = 5$ according to the analysis that will be described in section 6.3.2. The input pose noise parameters were set to the values of (6.1). The inlier threshold was set to 6 pixels for all the algorithms, including $\sigma_i$ as the soft inlier threshold of SOREPP. We used USAC [RCP+13] in two modes: The first used the five-point algorithm, where the implementation was taken from [SSS06], and the second mode used our own implementation of the closed form three-point algorithm suggested by [NZG+12]. In both modes, the local optimization method was used, applying a non–linear Levenberg-Marquardt optimization for a few iterations over all the five relative pose parameters. USAC parameters were adopted to low inlier fractions by reducing the SPRT $\epsilon$ parameter to 0.05 and increasing the iterations limit to 850,000 for the five-point mode and 20,000 for the three-point mode, both are supposed to be sufficient for very low inlier fractions. Bundle adjustment was initialized by the pose prior. It used epipolar constraints with a robustified cost function and pose priors regularization, which is, in fact, the same inner optimization function as SOREPP (5.5). It is referred as “BA-Epi” in the rest of the paper. All random algorithm results were naturally averaged by the large number of trials and additionally
Figure 6.2: Results on synthetic data, distant cameras, 10% inliers. Rows: (a): Side-by-side configuration. (b): One-in-front-of-the-other configuration. Columns: (1): Orientation errors. (2): Translation vector errors. (3): Success ratio, based on the maximal distance between the control points and the corresponding epipolar lines. All angles are given in degrees.

averaged over 5 runs. The focal length was set to the value in the EXIF header, and the principal point to the image center. The evaluation was conducted on a Windows 7 64bit, Intel i5 4GB computer, using only a single core.

6.2.2 Distant Cameras

The results for distant cameras and a high inlier fraction of 80% are shown in Figure 6.1. As expected by the analysis described in 2.2, the accuracy of the pose prior data is low, resulting in low success ratio in the image domain. Among the different algorithms being compared, SOREPP yields the most accurate relative pose estimation. Bundle adjustment performs well for low noise levels, but is sensitive to increased noise, since in such cases the initial errors are beyond its basin-of-attraction. USAC with the three-point algorithm suffers from a similar phenomenon, because it takes the prior leveling angles as hard constraints. USAC with the five-point algorithm (followed by a non linear refinement) yields the closest results to SOREPP. Both succeeded to solve the epipolar geometry for most trials, as seen in (a3) and (b3). However, analyzing the relative pose domain reveals that the accuracy of SOREPP is higher even for high noise levels. This is explained by the fact that there is a small uncertainty region around the true relative pose, where different relative poses in that region all explain the epipolar geometry in
Figure 6.3: Results on synthetic data, close cameras with unknown translation vector, 80% inliers. Rows: (a): Side-by-side configuration, 80% inliers. (b): One-in-front-of-the-other configuration, 80% inliers. Columns: (1): Orientation errors. (2): Translation vector errors. (3): Success ratio, based on the maximal distance between the control points and the corresponding epipolar lines. All angles are given in degrees.

The results for low inlier fraction of 10% are shown in Figure 6.2. SOREPP succeeded to estimate the epipolar geometry for most of the trials, when a few failures occurred in the highest noise level. In one-in-front-of-the-other configuration, the epipolar lines have various directions, therefore the effect illustrated in Figure 5.1 becomes weakened, and the basin-of-attraction of the optimization process is decreased. This effect harms the performance of BA, as seen clearly by the graphs, but does not affect SOREPP, which overcomes the reduced basin-of-attraction by the Pose Region Search component. Even in the worst cases, the accuracy of SOREPP is similar to the one of USAC with the five-point algorithm. In side-by-side configurations, the advantage of SOREPP is significant. USAC with the five-point algorithm performs quite well. Still, it suffers from the fact that the five-point algorithm is sensitive to noise, since it is based on solving a tenth degree polynomial. Such noise in the low inlier fraction case is much more harmful than the high inlier fraction case. In addition, outliers that are randomly located close to the true epipolar geometry might also skew the solution.

Due to the weights of the putative correspondences, USAC usually converged in the
Figure 6.4: Results on synthetic data, close cameras with unknown translation vector, 10% inliers. Rows: (a): Side-by-side configuration. (b): One-in-front-of-the-other configuration. Columns: (1): Orientation errors. (2): Translation vector errors. (3): Success ratio, based on the maximal distance between the control points and the corresponding epipolar lines. All angles are given in degrees.

first iterations. Its average runtime for the 10% inlier fraction case was 0.5 seconds per image pair, the same runtime as for the 80% inlier fraction case. This is probably since most of the runtime was used by the local optimization process. SOREPP’s average runtime was 0.08 seconds per image pair.

6.2.3 Close Cameras

In this experiment, no prior data is available on the translation vector, but a random position is supplied instead. The results for high inlier fraction case of 80% are shown in Figure 6.3. SOREPP had no problem overcoming the incorrect translation vector, and estimated the epipolar geometry well, both in the relative pose domain and in the image domain. The estimated relative pose is more accurate than those of USAC with the five-point algorithm. BA, which does not contain the Pose Region Search component of SOREPP, failed to overcome the incorrect data.

The results for low inlier case of 10% inliers look different, as shown in Figure 6.4. Overcoming such low inlier fractions as well as incorrect translation vectors is too much for SOREPP. The results for the side-by-side configuration are better than the one-in-front-of-the-other configuration, but still worse than those of USAC with the five-point algorithm. This use-case is beyond the working envelop of SOREPP. However,
in practice this case should only rarely happen, since close cameras yield similar images, therefore typically high inlier fractions.

6.3 Experiments on Real Images

In this section three large datasets consisting of 578 image pairs were used to evaluate SOREPP performance for real life difficult conditions. Since existing images which were used previously to analyze epipolar geometry estimation algorithms do not contain pose measurements, new datasets had to be created. The datasets, including the images, the measured pose of each camera, the control points, and code that applies SOREPP on them, are all available at [SOR].

6.3.1 Experimental Setup

The setup data for the experiments on real images is similar to the setup for the experiments on synthetic data where it is possible. The differences are detailed below.

Evaluation Images

The evaluation data is composed of three datasets collected at different locations. *Open1* was taken using a variable zoom Canon PowerShot S3 IS camera. Because Euler angles were not measured but calculated afterwards by coordinate differences, the azimuth and the pitch were relatively accurate. The roll was approximated to 0°. *Open2* and *Urban* were taken by a Samsung Galaxy SII smartphone using SitisMobile’s “GeoCam” application [Sit12], supplying the pose associated with each image. The datasets present challenging scenarios with wide baseline images, small overlapping regions, scale changes, and nondescript objects that make feature matching difficult. Under these conditions the inlier fractions decreases often to less than 10%. Noisy pose measurements are an additional challenge. The characteristics of the datasets are given in Table 6.1 and representative image pairs are shown in Figures 1.1 and 6.6 to 6.9.

Evaluation Method

Since ground truth pose data is unavailable, we evaluate the estimated epipolar geometry based on the image domain only. Each dataset consists of several images of the same scene taken from different locations. A few control points have been defined in the scene and manually marked in each image, when visible. We tried to scatter the points over the images, depending on the ability to recognize the same locations accurately, and on the overlap regions between images. The Image pairs are created from images taken at two different locations, automatically generating a set of connected control points. This way even a small set of images creates a large set of image pairs to be used in the experiments. The control points were used as described in Section 6.2.1. The success threshold $\tau$ defined for each dataset is seen in Table 6.1. An algorithm’s performance is
Table 6.1: Characteristics of the different datasets. Im. Pairs are the image pair quantities, Cam dist are the distances between the cameras in meters, Fov X are the cameras’ fields-of-view in the x axis, Init Errs are the maximal errors calculated directly by the pose prior data in pixel units, and $\tau$ is the success threshold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Open1</th>
<th>Open2</th>
<th>Urban</th>
</tr>
</thead>
<tbody>
<tr>
<td>Im. Pairs</td>
<td>246</td>
<td>224</td>
<td>108</td>
</tr>
<tr>
<td>Cam dist</td>
<td>250 – 1640</td>
<td>40 – 440</td>
<td>15 – 200</td>
</tr>
<tr>
<td>Camera</td>
<td>Canon S3 IS</td>
<td>Galaxy SII</td>
<td>Galaxy SII</td>
</tr>
<tr>
<td>Res</td>
<td>640 × 480</td>
<td>2048 × 1536</td>
<td>2048 × 1232</td>
</tr>
<tr>
<td>Fov X</td>
<td>4.5° – 51°</td>
<td>59.6°</td>
<td>59.6°</td>
</tr>
<tr>
<td>Az Diff</td>
<td>$\leq$ 40°</td>
<td>$\leq$ 25°</td>
<td>$\leq$ 70°</td>
</tr>
<tr>
<td>Init Errs</td>
<td>$\leq$ 230</td>
<td>$\leq$ 375</td>
<td>$\leq$ 980</td>
</tr>
<tr>
<td>$\tau$</td>
<td>10</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

evaluated mainly by counting the number of successful image pair matches. Due to the asymmetric nature of the relative pose, we consider an image pair in reverse order to be a different pair.

**Implementations and Running Parameters**

All the algorithms used the same features, calculated by the implementation of SIFT provided by [Ved07] and adapted to be upright. The algorithms used the same putative correspondences, which are the 400 best ranked BLOGS correspondences (using the full set of putative correspondences decreases the inlier fractions, therefore reduces the quality of the results), besides BEEM which defines putative correspondences by the SIFT ratio test and uses it as part of the algorithm. SOREPP was tested with both putative correspondences types.

The algorithms used the same parameters as in the experiments of synthetic data. The pose noise standard deviations were also set to the same values as in the synthetic analysis, besides the azimuth value in Open1, which was calculated and not measured. It was therefore set to 1.5°.

We expanded the analysis to some additional algorithms. One algorithm is a conventional BA that tries to estimate the structure besides the cameras pose by minimizing re-projection errors (referred as “BA-Reproj” in the paper). We used our own implementation with a Gaussian M-estimator as a robust function and with pose priors regularization. Structure points were initialized by intersecting putative correspondences according to the pose priors. Two additional algorithms were BEEM and BLOGS, which estimated the fundamental matrix without taking into account the intrinsic calibration data.
Table 6.2: The contribution of the basic SOREPP components: the regularization term and the Pose Region Search component. The left column represents the pure optimization of (5.2), since both components are disabled. The number of successfully estimated image pairs are compared. Putative BLOGS correspondences were used in this experiment. See text for discussion.

<table>
<thead>
<tr>
<th>Regularization</th>
<th>×</th>
<th>√</th>
<th>×</th>
<th>×</th>
<th>√</th>
<th>√</th>
<th>√</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Open1</td>
<td>121</td>
<td>138</td>
<td>141</td>
<td>130</td>
<td>158</td>
<td>164</td>
<td>164</td>
</tr>
<tr>
<td>Open2</td>
<td>105</td>
<td>100</td>
<td>146</td>
<td>153</td>
<td>162</td>
<td>174</td>
<td>174</td>
</tr>
<tr>
<td>Urban</td>
<td>24</td>
<td>21</td>
<td>59</td>
<td>53</td>
<td>55</td>
<td>56</td>
<td>59</td>
</tr>
</tbody>
</table>

Table 6.3: Comparison between a single run of BEEM and SOREPP on the Open2 dataset.

<table>
<thead>
<tr>
<th></th>
<th>BEEM succeeded</th>
<th>BEEM failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOREPP succeeded</td>
<td>87</td>
<td>71</td>
</tr>
<tr>
<td>SOREPP failed</td>
<td>14</td>
<td>52</td>
</tr>
</tbody>
</table>

6.3.2 Analyzing SOREPP Components

The contribution of the two basic components of SOREPP was evaluated first: The regularization term in (5.5), and the Pose Region Search component under different numbers of optimization initializations $m$. For this experiment we disabled the Solution Refinement step described in section 5.4.3. Note that when the regularization term of (5.5) is disabled, then the optimization problem degenerates to (5.2). If additionally $m = 0$, then SOREPP performs only the pure optimization of (5.2). The combinations where $m = 0$ are equivalent to BA with epipolar constraints, with or without pose priors regularization.

The results using BLOGS correspondences are summarized in Table 6.2. These results reveal that the most important component is the Pose Region Search. For example, this component more than doubles the number of successfully estimated image pairs on the Urban dataset, compared to pure optimization: from less than 25 image pairs to more than 50. Its importance depends on the accuracy of the pose prior data. Usually choosing only the best candidate in addition to the input pose is sufficient for accurate estimation. The regularization term has reduced importance. Without regularization, the performance decreases in up to 20% on all the datasets, having the biggest effect for Open1 dataset, where the accuracy of the pose prior data is relatively good. The importance of the regularization was also explained in Section 6.2.2, where it helped to improve the accuracy of the estimated relative pose.

We compared the accuracy of SOREPP and BEEM on the Open2 dataset using the confusion matrix shown in Table 6.3, with default parameters and SIFT ratio putative correspondences. While in general SOREPP outperforms BEEM, it is interesting to
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Open1</th>
<th>Open2</th>
<th>Urban</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOGS</td>
<td>91.4</td>
<td>75.5</td>
<td>43</td>
<td>39.8%</td>
</tr>
<tr>
<td>BEEM</td>
<td>101.4</td>
<td>103</td>
<td>45.2</td>
<td>41.9%</td>
</tr>
<tr>
<td>USAC-5pts</td>
<td>76.4</td>
<td>90.2</td>
<td>42.2</td>
<td>39.1%</td>
</tr>
<tr>
<td>USAC-3pts</td>
<td>90.2</td>
<td>95.4</td>
<td>47.8</td>
<td>44.3%</td>
</tr>
<tr>
<td>BA-Reproj</td>
<td>65</td>
<td>73</td>
<td>11</td>
<td>10.2%</td>
</tr>
<tr>
<td>BA-Epi</td>
<td>138</td>
<td>100</td>
<td>21</td>
<td>19.4%</td>
</tr>
<tr>
<td>SOREPP-SR</td>
<td>158</td>
<td>158</td>
<td>58</td>
<td>53.7%</td>
</tr>
<tr>
<td>SOREPP-BL</td>
<td>166</td>
<td>174</td>
<td>58</td>
<td>53.7%</td>
</tr>
<tr>
<td>Prior Only</td>
<td>40</td>
<td>44</td>
<td>2</td>
<td>1.9%</td>
</tr>
</tbody>
</table>

Table 6.4: Performance comparison for the different algorithms. Accuracy is measured by counting correct image pairs. **SOREPP-SR** is based on SIFT ratio putative correspondences and **SOREPP-BL** uses BLOGS correspondences. **Runtime**: Average time in seconds for C/C++ implementations. **Prior Only** are reference results in which \( F \) was calculated directly from the measured pose.

study the cases where BEEM succeeded and SOREPP failed in order to understand SOREPP’s weaknesses. Although the solutions obtained by SOREPP were usually accurate at the majority of the control points, there were some isolated points with high errors, showing that these solutions were incorrect. These failures were a result of the soft threshold applied to all the putative correspondences, which allowed outliers close to the correct solution, or large clusters of correspondences, to skew the solution. These outliers, or clusters, had larger influence in (5.5) than some isolated correct correspondences. RANSAC-based algorithms have a decreased sensitivity to this phenomenon as the solution is estimated only by a small sampled set with no role played by any of other correspondences. Still, failures of this type are not dominant, and SOREPP succeeds to estimate epipolar geometry accurately for most of the image pairs.

### 6.3.3 Performance Comparison

The performance of SOREPP is compared to that of several state-of-the-art algorithms in Table 6.4. The datasets represent a greater challenge than the synthetic data, since in many image pairs the inlier fractions are even lower than 10%, the inliers may not be scattered well on the image, the intrinsic camera parameters are not accurately known, and other real world effects. As expected, epipolar geometry calculated directly from the pose prior data is inaccurate and cannot be used without further image-based processing. For example, on the **Urban** dataset the pose prior data is accurate enough for only 2 image pairs. SOREPP demonstrates its ability to successfully match real images taken by a smartphone when the pose measurements are contaminated with substantial noise and the inlier fractions are low. It outperforms algorithms that do not use pose priors (BLOGS, BEEM and USAC with the five-point algorithm), thus demonstrating their importance. For example, on **Open1** dataset, SOREPP gets the score of 166 while
the best RANSAC based method gets only 101. It also performs better than USAC with the three-point algorithm, achieving only 90 successful estimations on this dataset, demonstrating the advantage of using all the priors and not only the leveling angles.

USAC with the five-point algorithm performed worse than on the synthetic data. This has several explanations. Some of the failures occurred due the low inlier fraction, which were many times lower than 10%. USAC is based on PROSAC, therefore needs less iterations than RANSAC, but if the weights of the putative correspondences are not good enough and no solution is found, it degenerates to RANSAC and needs many iterations. However, taking such a large number of 850,000 iterations is impractical, all the more so if even more iterations are required. Other failures might occur due to real world conditions such as the scattering of the points on the image or the limited accuracy of the intrinsic parameters. Since the five-point algorithm is based on solving a tenth degree polynomial, it is relatively sensitive to such effects, in comparison to SOREPP.

The results emphasize an attribute of pose priors that was not mentioned yet. In low inlier fraction cases, it is difficult to distinguish between the true solution and a false one, since both might have a similar number of detected inliers. Pose priors, where available and used by the estimation algorithm, serve also to reject false solutions, which are typically farther from the pose priors than the true one.

Between the two bundle adjustment variants, the one which is based on epipolar constraints works better, and gives the closest results to SOREPP on average. It is reasonable, since the core of SOREPP is the same optimization process that is used in this BA variant. Yet, the BA performances are significantly worse than that of SOREPP, demonstrating the advantage of the SOREPP framework and the importance of the Pose Region Search component in order to overcome large pose errors.

The Urban dataset is characterized by high pose errors, reducing the advantage of SOREPP. Still, even on this dataset, better results are obtained when pose prior data is used. SOREPP achieves the score of 58, in comparison to 42 – 48 of all the RANSAC based methods. SOREPP performs much better than BA, which can not overcome such high prior noises and has succeeded only on 21 image pairs.

SOREPP’s performance using BLOGS putative correspondences is better than its performance using correspondences that are based on the SIFT ratio test, having the biggest advantage on Open2 dataset, where the score has improved from 158 to 174.

The runtime column in Table 6.4 reveals that SOREPP is fast, having an average runtime of about 0.1 seconds per image pair. SOREPP’s runtime is not affected by the low inlier fractions, which slow down all the RANSAC-like algorithms, for example more than 100 seconds for USAC with the five-point algorithm. RANSAC based methods runtime can be reduced by choosing a smaller iteration limit, at the expense of their accuracy. SOREPP’s runtime can be further reduced by better implementation and parallelizing the computation. This can be done, for example, by solving each of the different initializations on a different core.
Figure 6.5: Algorithm performance compared over all the sets. The curves represent the number of correct image pairs as a function of $\tau$. See Table 6.4 for details about the various algorithms. Best viewed in color.

Figure 6.5 shows the performance of each algorithm as a function of a continuous threshold $\tau$, showing again the low accuracy of the measured pose (prior only) while emphasizing the advantage of using it in an estimation algorithm such as SOREPP. For example, for a threshold of 10 pixels, SOREPP obtains the score of 357, while the second best method obtains the score of 237 (about 33% less) and the pose priors gets only 54. Figures 6.6 to 6.9 present some examples from the datasets and compares the performance of the algorithms on them. SOREPP succeeds in estimating the epipolar geometry when other algorithms have difficulty doing so. For example, in Figure 6.9(8), SOREPP overcomes prior error of 337.8 pixels and gets an error of 10.6 pixels. The second best method is BLOGS, achieving the maximal error of 26.3 pixels over the control points.
Figure 6.6: Images from Open1 dataset matched using some of the algorithms. For each image pair, the left image is the reference image with several control points marked. The right image is the target image, with the control points and their corresponding epipolar lines estimated using SOREPP. All the algorithms besides BEEM used the best 400 BLOGS putative correspondences. Inliers are counted as detected by SOREPP. The Prior only column gives reference results calculated directly by the pose prior data. Bold represents successful estimations, where the maximal control point error is smaller than the proper $\tau$ (10 pixels on this dataset).
Figure 6.7: More images from Open1 dataset. See Figure 6.6 for details. $\tau = 10$ pixels.
<table>
<thead>
<tr>
<th>id</th>
<th>#Control points</th>
<th>Prior only</th>
<th>#Inliers (SOREPP detet.)</th>
<th>BLOGS</th>
<th>BEEEM</th>
<th>USAC 5pts</th>
<th>USAC 3pts</th>
<th>BA Epipolar</th>
<th>SOREPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>79.5</td>
<td>23 (1.75%)</td>
<td>54.2</td>
<td>59.9</td>
<td>46</td>
<td>40.1</td>
<td>68.1</td>
<td>8.1</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>306.1</td>
<td>39 (9.75%)</td>
<td>22</td>
<td>16.4</td>
<td>21.1</td>
<td>19.8</td>
<td>259.4</td>
<td>8.1</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>65.6</td>
<td>52 (13%)</td>
<td>10.1</td>
<td>16.9</td>
<td>19.3</td>
<td>16.1</td>
<td>59</td>
<td>40.7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>41.6</td>
<td>20 (5%)</td>
<td>511.9</td>
<td>19.7</td>
<td>100.6</td>
<td>103.4</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.8: Images from Open2 dataset. See Figure 6.6 for details. \( \tau = 15 \) pixels.
Figure 6.9: Images from *Urban* dataset. See Figure 6.6 for details. $\tau = 15$ pixels.
Chapter 7

Conclusion and Further Research

The paper introduces SOREPP, a novel robust algorithm designed to estimate epipolar geometry using pose priors. Pose priors are used to apply an optimization scheme, while their estimated uncertainty is used in order to focus the search in the relevant area in the parameter space. SOREPP is able to deal with extremely low inlier fractions of less than 10% while overcoming significant pose noise, such as an azimuth error of $7^\circ$. It does so without slowing down the runtime which, unlike RANSAC-based algorithms, is not sensitive to the inlier fractions. In addition, SOREPP can estimate epipolar geometry for close cameras, when the translation vector might be incorrect. Extensive evaluation on hundreds of image pairs was performed in various conditions, demonstrating the ability of SOREPP to estimate epipolar geometry even under severe, realistic conditions. When the pose priors are supplied, SOREPP outperforms current state-of-the-art methods, such as BEEM [GS08], BLOGS [BS09], and USAC [RCP+13] applying the five point [Nis04] or the three point [NZG+12] algorithms.

It looks interesting to adapt the SOREPP framework for other geometric vision problems, under a further research. SOREPP is based on optimizing the minimization problem defined in (5.5), which is brought again due to its importance:

$$\hat{s} = \arg \min_s \left\{ c \left( \sum_{k \in \Omega_{all}} \hat{w}(k) \left( 1 - \exp \left( -\frac{d(k)^2}{2\sigma_h^2} \right) \right) \right) + (\lambda(s))^2 \right\},$$

where $s$ is the vector of unknowns, $\hat{w}(k)$ is a normalized weight of the $k$-th putative correspondence, $d(k)$ is the error of that putative correspondence, and $\sigma_h$ and $c$ are parameters. The energy function combines the objective to minimize the errors for as many putative correspondences as possible while keeping the solution close to the pose priors. In order to solve other geometric problems, the same energy function structure can be used, but with different distance functions $d(k)$ and different unknowns vector $s$. For example, such framework may be used for homography estimation, or for the trifocal tensor.

Adaption of SOREPP for homography estimation is not trivial, since unlike epipolar
geometry, homography incorporates the scene plane parameters in the mathematical model, besides the relative pose of the cameras. In order to apply SOREPP for this estimation problem, first an analysis needs to be performed, examines the distribution of a plane’s normal direction, given that a common point on this plane has been found in two images using a similarity invariant feature detector and descriptor such as the SIFT, and given the relative pose of the two cameras. This distribution is essential to initialize the optimization process. It may also be used to sample additional initial values, and to regularize the estimation.

For three view geometry estimation, a main difficulty arises from the standard formulation of multiview geometry, where the structure (the location of the points in the scene) needs to be estimated. Applying SOREPP to estimate a large unknown vector, which contains the location of all the points besides the relative pose of the cameras, might fail the algorithm. The location of these points has no good initialization, and so large unknowns vector gives the algorithm too many degrees of freedom, preventing it from converge, at least in the difficult cases. A solution to this problem may be structure-less formulation, such as the one given in [Ind11].
Appendix A

Rotation Matrix Calculations

The rotation matrix can by constructed by three rotation matrices, each representing a rotation by a single Euler angle around one axis, as shown in (2.6):

\[ R = R_y(\phi)R_x(\theta)R_z(\psi). \]

The three Euler rotation matrices has the following structure:

\[
R_z(\psi) = \begin{pmatrix}
\cos(\psi) & -\sin(\psi) & 0 \\
\sin(\psi) & \cos(\psi) & 0 \\
0 & 0 & 1
\end{pmatrix}, \quad (A.1)
\]

\[
R_x(\psi) = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos(\theta) & \sin(\theta) \\
0 & -\sin(\theta) & \cos(\theta)
\end{pmatrix}, \quad (A.2)
\]

\[
R_y(\phi) = \begin{pmatrix}
\cos(\phi) & 0 & -\sin(\phi) \\
0 & 1 & 0 \\
\sin(\phi) & 0 & \cos(\phi)
\end{pmatrix}. \quad (A.3)
\]

The resulting rotation matrix can be expressed explicitly by the Euler angles:

\[
R = \begin{pmatrix}
\cos(\phi) \cos(\psi) + \sin(\phi) \sin(\psi) & -\cos(\phi) \sin(\psi) + \sin(\phi) \sin(\theta) \cos(\psi) & -\sin(\phi) \cos(\theta) \\
\cos(\theta) \sin(\psi) & \cos(\theta) \cos(\psi) & \sin(\theta) \\
\sin(\phi) \cos(\psi) - \cos(\phi) \sin(\theta) \sin(\psi) & -\sin(\phi) \sin(\psi) - \cos(\phi) \sin(\theta) \cos(\psi) & \cos(\phi) \cos(\theta)
\end{pmatrix}. \quad (A.4)
\]

Using (A.4), \( R \) can be decomposed to Euler angles using (2.8):

\[
\psi = \arctan 2(R(2,1), R(2,2)),
\]

\[
\theta = \arctan 2\left(R(2,3), \sqrt{R(2,1)^2 + R(2,2)^2}\right),
\]

\[
\phi = \arctan 2(-R(1,3), R(3,3)).
\]
Bibliography


The third innovation concerns the method of experiments. Instead of taking pairs of unrelated images as is customary, we took a collection of images from different viewpoints of the same landscape. By marking control points on each image only once, we were able to build a large database including hundreds of pairs of images linked by a connection to the control points in an automatic manner. With this method, we could compare the algorithms on a large database of images, and therefore obtained more reliable results than a short test on a small number of pairs of images. The test against the information against artificial information and real images: the algorithm was tested in two ways: it was found that the new algorithm is able to deal with high clusters and significant noise, and still measure the relative location of the cameras in a generally more precise way. The location of the cameras in the test information is not known, since the actual location of the cameras is unknown, it is important to mention that the difference between the results of the new algorithm and the methods available today is very large, in real conditions. Therefore, the impact of the algorithm is not used at all, even if noise, the experiments indicate the value of the early information on the location, and, in addition, the ability of the new algorithm to measure geometric relationships in cases where existing methods fail to do so.
The algorithm for estimating parameters and matching images is not straightforward, and it involves complex calculations. One approach is to use a new algorithm, called SOREPP, which is based on an optimization process of a price-resistant function (M-estimator). The algorithm is designed to be robust against outliers, and it can handle situations where the percentage of outliers is high.

The algorithm is implemented in Matlab and is available for download. It has been tested on large datasets of hundreds of object pairs, and it shows significant improvement over the best existing methods.

The main advantage of the algorithm is its ability to handle large noise in the parameter space, and it is especially useful when the initial information is inaccurate. Furthermore, the algorithm is able to handle situations where the initial estimates are far from the true values.

The algorithm is also capable of dealing with noise in the image matching process, which can be a significant problem in real-world applications.

In summary, the algorithm presented in this paper is a significant advancement in the field of image matching and estimation.

© Technion - Israel Institute of Technology, Elyachar Central Library
הנה תקציר:

האפקטורי האפומרוטי הוא האפקטורי המפונקטי בין שני מבטים F, והיא מגולמת במטריצה היסודית, גיאומטריה אפומרוטי היא הגיאומטריה הפנימית בין שני מבטים F. שערוך גיאומטריה אפומרוטי היא אחת משאריות היסוד בראיה ממוחשבת והיא משמשת כרכיב בסיסי עבור אלגוריתמי סטריאו,�回ה תלת מימד, ניווט רובוטים והシステムים הפנימיים של התמונה, לדגמה באצטדיון נוכחות ענים כל לחץ מניפה, תחיית משותף קורונה, F, שערוך זה מתבצע דרך קבע בשני שלבים. בירחון הראשון, נקודות עניין של כל התמונה, SIFT, משולבים בין התמונות יתלע התמונה יריי ראיית, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, למתוך ניסח המехать התמונה, F, שערוך זה מתבצע בשני שלבים. בירחון הראשון, נקודות עניין של כל התמונה, SIFT, משולבים בין התמונות יתלע התמונה, real RANSAC, בטרון זרם, למתוך ניסח המехה התמונה, F. SIFT, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, למתוך ניסח המехה התמונה, F. הערה: לשערוך real RANSAC, התמונה של שני מבטים F בראשון התמונה יריי ראיית, תשלול חוסכים נוכחים, shogun. real RANSAC, בטרון זרם, למתוך ניסח המехה התמונה, F. SIFT, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, למתוך ניסח המехה התמונה, F. SIFT, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, למתוך ניסח המехה התמונה, F. SIFT, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, L. SIFT, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, L. SIFT, תשלול חוסכים נוכחים שערום, real RANSAC, בטרון זרם, L.
שערוך חסון של גיאומטריה אפיפולרית
באמציהו של יומן בדיאגוג מוקדם
אכ רועש על מייקם המצלמה

תיבנה על מחקר

לשם מ holster חלקי של הדירישות لكنבלה התוואר
מגישים lem Điểmים במדעי המחשב

יהונתן גולדמן

הוגש לסנט הטכניון - מכון טכנולוגי לישראל
סינו החשעי"ד
تشك 2014
שערוך חסון של גיאומטריה אפיפולרית
בأمצעות יומימיות במידע موقف
אך רעשי על מיקום המצלמות

יהונתן גולדמן