Constrained Visual-Inertial Localization With Application And Benchmark in Laparoscopic Surgery

Regine Hartwig$^{1,3}$, Daniel Ostler$^1$, Jean-Claude Rosenthal$^2$
Hubertus Feußner$^1$, Dirk Wilhelm$^1$, Dirk Wollherr$^3$

Abstract—We propose a novel method to tackle the visual-inertial localization problem for constrained camera movements. We use residuals from the different modalities to jointly optimize a global cost function. The residuals emerge from IMU measurements, stereoscopic feature points, and constraints on possible solutions in SE(3). In settings where dynamic disturbances are frequent, the residuals reduce the complexity of the problem and make localization feasible. We verify the advantages of our method in a suitable medical use case and produce a dataset capturing a minimally invasive surgery in the abdomen. Our novel clinical dataset MITI is comparable to state-of-the-art evaluation datasets, contains calibration and synchronization and is available at [1].

1. INTRODUCTION

Many researchers have studied simultaneous localization and mapping (SLAM) in the computer vision, and robotics community [2], inspiring many innovative products. Applications such as virtual reality, augmented reality [3], unmanned vehicles, and autonomous robots [4] rely heavily on stable pose estimates based on SLAM systems.

Challenging situations include a dynamic environment which prevents the SLAM algorithms from being applicable in many use-cases. The works of [5–7] show that it is possible to add residuals specially designed for a use-case and jointly optimize a cost function. Motion priors, reprojection from a calibrated multi-camera setup, or additional measurements reduce the degrees of freedom on the solution space.

MIS has proven to have many medical advantages for the patient. Stable virtual/ augmented reality and assisting roboter arms could lead to enhanced hand-eye coordination in MIS and subsequently to a whole new way of conducting surgeries. Furthermore, SLAM is a game-changer for applications such as mapping detected metastasizing carcinoma for cancer staging, safety routines (e.g., collision avoidance), autonomous systems, image-guided surgery, input generation for artificial intelligence, and anatomic scene graphs. The huge disadvantages (magnetic distortion, line-of-sight, room consumption) disqualify available tracking equipment for most interventions. Data from this domain was acquired, and implementation shows the applicability of the proposed constrained visual-inertial SLAM method in this challenging dynamic surrounding.

The paper is structured as follows: Section [IV] presents the visual-inertial odometry algorithm with movement constraints. Section [V] discusses our novel dataset [1] and provides useful information for users. In Section [VI] we will discuss the implementation of the proposed algorithm, discuss properties of the calibrated data from our dataset, and evaluate the performance by experiments.

II. PROBLEM STATEMENT AND CONTRIBUTION

a) Constrained Visual-Inertial Localization: [3], and further developments have become popular state-of-the-art methods [8]. These algorithms focus on feature point matching based on binary descriptors, even for small viewpoint changes, where descriptor matching is made real-time applicable by speeding up the process with priorly trained or online learned [9] bag-of-words vocabularies [10, 11].

However, we focus on domains where feature point descriptor matching and tracking perform worse due to challenging image properties [12]. Sparse or semi-dense optical flow [5, 13, 14] are alternatives that perform well for high video framerate and can speed up trajectory generation for the tracking part of the SLAM system.

Additional information from inertial measurement units (IMU) [15–17] improves visual odometry methods in difficult situations, where tracking is lost, and does not need an external measurement equipment, i.e., avoiding line-of-sight problems.

In contrast to all previously mentioned strategies, we additionally exploit naturally given motion constraints [19], which is also famous in the medical domain as remote center-of-motion (RCM) [20][21], reducing the solution space to tackle the more challenging image domains, which contain large movements in the image, occlusions, fog, deformable surfaces or difficult illuminations conditions [21].

Subsequently, we outline our approach in more detail: We formulate residuals for IMU, stereo camera, and movement constraints and show that the solution space reduces by pivoting the camera around a point from 6 to 4 degrees of freedom. Afterward, we propose to use optical flow in combination with gyroscope readings as an odometry alternative to descriptor-based feature tracking for dynamic image data and high video frequency. Finally, for the refinement of this odometry, we formulate the global cost function and optimization strategies. Our method’s main advantage and contribution is the ability to relocalize the camera in unseen environments by minimizing the global cost function, which incorporates all residual terms emerging from sensors.

1 Authors are with Research Group MITI, Surgical Department, Technical University of Munich (TUM), Germany
2 Authors are with Fraunhofer HHI, Berlin, Germany
3 Authors are with Chair of Automatic Control Engineering, Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Germany regine.hartwig@tum.de
and constraints. The gravity and magnetic field provide the necessary absolute references. In addition, the Lucas-Kanade tracking of the stereo reconstructed locally rigid parts in the environment provides information about the distance from the pivot point.

b) Dataset: Publicly available datasets play a vital role in the evolution of new algorithms, and technologies [22–25]. Researchers focusing on the minimally invasive application have already created endoscopic datasets [21, 25, 27], which are suitable for investigating computer tomography registration, tissue deformation/reconstruction [28–30], and instrument segmentation. Since they are not complete or are not capturing a handheld intervention, those are unsuitable for testing SLAM applications. In this paper, we introduce a novel dataset suitable for testing SLAM in the MIS domain. To the best of our knowledge, our dataset is the first publicly available surgical dataset comparable to [22–24]. Our MITI Dataset [1], available under the Creative Commons Attribution license (CC-BY 4.0), provides all necessary data by a complete recording of a handheld surgical intervention at Research Hospital Rechts der Isar of TUM. It contains multimodal sensor information from IMU, stereoscopic video, and infrared (IR) tracking as ground truth for evaluation. Furthermore, calibration for the stereo scope, accelerometer, magnetometer, the rigid transformations in the sensor setup, and time-offsets are available. We wished to choose a suitable intervention, namely diagnostic laparoscopy, that contains very few cutting and tissue deformation and shows a full scan of the abdomen with a handheld camera such that it is ideal for testing SLAM algorithms. It incorporates 667 seconds with images (1080p60 RGB) for each camera, readings of two IMU sensors at 220Hz, and IR transformations to two different targets at 20Hz.

Intending to promote the progress of visual-inertial algorithms designed for MIS application, we hope that our clinical training dataset helps and enables researchers to enhance algorithms.

III. NOTATION

The following notions are adapted from [31–33] using the rotation groups SO(3) and the rigid motion group SE(3), which fulfill the properties of a Lie group. We use the notation exp : m → M to denote the mapping from tangent space, i.e. Lie algebra m to Lie group M. The Lie algebra so(3) = skew(3) of SO(3) has d=3 degrees of freedom and axe(3), tangent space of SE(3), has d=6 degrees of freedom. They are isometric isomorphism to R^d, and we use the function Exp as the composition of the hat operator and the matrix exponential exp. The function can explicitly be given in closed form by the Rodrigues formula. The inverse mapping is denoted by Log, which is the mapping from the Lie-Group to R^d. We further use the ⋉ : M × M → R^d as

\[ T_1 \bowtie T_2 := \log (T_1^{-1} T_2), \]  

which computes the difference between two transformation matrices mapped to the tangent spaces’ isomorph space R^d.

IV. VISUAL-INERTIAL SENSOR FUSION

Our method splits into two parts: Firstly, estimating the pose starting from previous keyframes by movement constraints, gyroscope readings, and Lucas-Kanade tracks of stereoscopic matches. Secondly, the algorithm conducts a windowed optimization over gathered keyframes, changing their location by jointly minimizing the residuals from all modalities, eliminating any drift that might accumulate.

A. Residuals

This subsection describes the residuals gathered from different modalities. The residuals are functions of the localization pose \( T \in SE(3) \) with the property that they are zero at the actual position and orientation.

a) Motion Constraints: The transformation \( T \in SE(3) \) transforms a vector in homogeneous coordinates from \( T \) to \( C \) by multiplication from the right. Since the motion of \( C \) is constrained to move around pivot point \( T \) but can also move forward and backward in the x-direction, we can reduce the 6 degrees of freedom to 4 degrees of freedom

\[ T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0^T & 1 \end{bmatrix}, \]

with \( t = t_x e_x \). This means a point in \( T \) is first rotated in the pivot point and then translated along the x-axis. The corresponding residual that enforces the result to have only 4 degrees of freedom is

\[ r_{\text{pivot}} = t_y e_y + t_x e_z. \]

This adds a penalty to the minimization problem if the solution is not in the reduced space.

b) Accelerometer: The accelerometer readings constrain the current camera pose by one corresponding bearing vector in world coordinates. We assume constant velocity as also proposed by [34]. Rotating the gravity \( g \) bearing vector from \( T \) to \( C \) and comparing it with the measurement \( \dot{g} \) gives the residual

\[ r_a = \dot{g} - \dot{R} \cdot g. \]

c) Magnetometer: The magnetometer provides similar to the accelerometer a correspondence in world coordinates

\[ r_m = m - R \cdot m, \]

with the magnetic field \( m \) in \( T \) and measurement \( m \) in \( C \).

d) Global Mappoints: The distance between a map point \( x \) in global coordinates and a matched 2D position \( u \) in image plane of current camera coordinates is

\[ r_{2D} = \Pi(T^T x) - u \]

with \( \Pi(\cdot) \) being the projection into distorted image plane and \( u = u_d \) the distorted image coordinates. In (6), the map points \( x \) are connected rigidly to the world coordinate system \( T \) at initialization by using the current camera pose estimate. We need to rewrite (6), attaching the mappoints to the camera coordinate system. Otherwise, since we optimize only the poses \( T \), the minimization of other residuals leads to the increase of (6). For this sake, we store the landmark
positions relative to the camera coordinate system $C$, in which the stereo matcher initialized them. This adds another computation step and one more parameter $\hat{T}'$ at detection time $t'$ to the residual in (6), namely
\[
\hat{x} = \hat{T}' \hat{x}' .
\]

Thus we have
\[
r_{2D} = \prod(\hat{T} \hat{T}' \hat{x}) - u ,
\]
which connects the observation frame with the initialization frame of a landmark.

e) Gyroscope: The gyroscope measurements in tangent space $\omega_{t'} \in \mathbb{R}^3 \equiv \mathfrak{so}(3)$ are integrated by gyroscope sample time $\Delta t$ and the non-abelsh product
\[
\omega_{t} := \prod(\omega_{t'})^{\hat{T}} \prod_{t' \in \{t_1, \ldots, t_2 - \Delta t\}} \exp(\omega_{t'} \Delta t) .
\]
This gives the update rule
\[
\hat{R}_{t_2} = \hat{R}_{t_1} \hat{\Delta R}_{t_1, t_2}
\]
and the residual term
\[
r_g = \hat{R}_{t_2} \otimes \hat{R}_{t_1} \hat{\Delta R}_{t_1, t_2} ,
\]
which constraints consecutive transformations to have a certain deviation in orientation, namely the integrated measurements of the angular velocity.

B. Visual-Inertial Odometry Tracking

This subsection describes our approach for the odometry tracking thread. It updates from the pose of the previous keyframe, generating good initializations for the optimization using gyroscope data and Lucas Kanade tracks of stereo correspondences. Algorithm 1 sketches the procedure, and in the following, we will explain line by line the computation steps the algorithm conducts.

After entering the loop, it first updates the pose based on the gyroscope measurements in Line 2. Although (11) usually only provides information about the orientation, a position update is possible due to the camera movement having only 4 degrees of freedom
\[
\hat{T}_{t_2} = \begin{bmatrix} I & \hat{t}_{t_1} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \hat{\Delta R}_{t_1, t_2} & \hat{R}_{t_1} & 0 \\ 0^T & 1 \end{bmatrix}
\]
with $\hat{t}_{t} = \hat{t}_{x,t} \hat{e}_x$. The new pose is on a sphere with radius $\hat{t}_{x,t}$, around the trocar entry point, leaving only the radius unknown but estimating all other degrees of freedom.

During the first iteration, the Lines 3 to 4 have no effect. The first frame is a keyframe, and the algorithm computes stereo correspondences by descriptor matching using ORB descriptors [35] in Line 3. ORB descriptors are faster to compute and compare than SIFT or SURF, whose benefits, e.g., scale and rotation invariance, are less essential for stereo-matching. For faster matching, it uses the projections provided by the camera calibration parameters as initialization for finding matches in the second image [8]. The epipolar error threshold [36] is adjusted to filter outliers
\[
\hat{x}^T E \hat{x} > d_{th} ,
\]
with epipolar matrix $E = \hat{t} \hat{R}$ and $\hat{\cdot}: \mathbb{R}^3 \rightarrow \mathfrak{skew}(3)$, defining the transformation between left $C_0$ and right $C_1$ lens. The matched points initialize new landmarks in Line 7.

Afterwards, in Line 8 for every incoming frame it updates the landmark projections $u \in \mathbb{R}^2$ in the 2D image plane, producing residuals of the form (6). For that, it uses the variational Lucas-Kanade-Method [37] taking two consecutive frames for every camera. This variational method performs well (small baseline, photometric consistency) if video frequency is high. For large displacements, the drift becomes large, and for fast movements, we lose track. Nevertheless, optical flow is suitable for local tracking.

With initialization computed in Line 2 it determines the camera pose $\hat{T}_{t}$ based on the 3D-2D-correspondences in Line 7 by a perspective-n-point algorithm [38] inside a random sample consensus (RANSAC) [39] scheme with refinement provided by OpenGV [40]. A sanity check on the rotation part using the IMU measurements and on the translation part using the constraints ensures locally rigid features.

Algorithm 1 Visual-Inertial Odometry with Optimization

Input:
- Initialization $\hat{T}_{t_0}$, Images $\{I_{0,t_i}, I_{1,t_i}\}_{i \in t}$
- IMU Data $\{\hat{\Delta R}_{t_i,t_{i+1}}, \hat{a}_{t_i}, \hat{m}_{t_i}\}_{i \in t}$

Output:
- $\{\hat{T}_{t_i}\}_{i \in t}$

1: while True do
2: \hspace{1em} Update $\hat{T}_{t_{i-1}}$ based on $\hat{\Delta R}_{t_{i-1}}$ and movement constraints
3: \hspace{1em} Lukas-Kanade tracking of landmarks
4: \hspace{1em} Update $\hat{T}_{t_i}$ based on 2D-landmark projections
5: \hspace{1em} if Decision for keyframe then
6: \hspace{2em} Stereo matching
7: \hspace{2em} Landmark insertion
8: \hspace{1em} end if
9: \hspace{1em} $i = i + 1$
10: end while

C. Optimization

The optimization presented in this subsection jointly minimizes all previously formulated residuals over a window of keyframes. As shown later the camera can globally relocalize itself by solving the proposed optimization problem solely by small tracks of locally rigid feature points and IMU measurements. In order to use all the information from sensor readings and images we build a cost function by summing up the normalized, weighted and robustified residuals that have been collected, i.e., (3) (5) (8) (11) which are available at each timestep. We denote them as residuals $r_k, k \in K$, where $K$ indicates time and residual type.

The overall optimization problem is
\[
\arg \min_{\hat{T}_{n:m}} \sum_{k} \rho_k (\|\alpha_k r_k (\hat{T}_{n:m})\|) := \sum_k f_k (\hat{T}_{n:m}) ,
\]
optimizing over the window of keyframe localizations $\hat{T}_{n:m} := (\hat{T}_{t_n}, \ldots, \hat{T}_{t_m})$. In above formulation, $\alpha_k$ stands...
for the scaling factor, and \( \rho \) for the robustification function. Each residual gets an individual scaling factor \( \alpha_k \). The evaluation part of this paper gives a more detailed overview of the chosen weight factors, optimization window size, and trigger of the optimization problem.

For robustification of residuals that are prone to outliers, we choose the Huber loss

\[
\rho_k(x) = \begin{cases} \frac{1}{2}x^2 & \text{for } |x| \leq \delta, \\ \delta(|x| - \frac{1}{2}\delta), & \text{otherwise}. \end{cases} \tag{15}
\]

The function input \( x \) in our application is the normalized and weighted residual \( ||\alpha_k r_k|| \). For residuals with a dense error distribution and no outliers, \( \rho_k \) is simply the identity.

Each of the residuals \( r_k(T_{n:m}) \) depends only on a maximum of two parameters \( T \in \text{SE}(3) \), and further, many parameters do not have a residual in common. The residual terms give the optimization problem the typical graph structure of SLAM problems. This sparse dependence on parameters makes the Hesse-matrix sparse, and the optimization problem suitable for a sparse Schur solver [41].

V. DATASET

In this section, we introduce our dataset and calibration procedures. In Fig. 1 we show the sensor setup during the data acquisition process. The sensor data consists firstly of the video the displayed timestamp has been captured at the start and end of the intervention. The sensor interfaces for IR/IMU are ethernet/bluetooth respectively, which causes a remaining timeoffset in the collected data. This offset is minimized by evaluating the tangent \( \xi_{t_i} \in \mathbb{R}^6 \approx \text{se}(3) \) for each sensor at coordinate system \( C \) at time \( t_i \). We then calibrated the timeoffset \( dt \) by comparing the velocities acquired from different sensors

\[ j, k \in \{"Visual Odometry", "IR", "IMU0", "IMU1"\}. \]

\[ \omega^{(k)} \parallel \parallel \omega^{(k)} \]

\[ \alpha \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \zeta \]

\[ \eta \]

\[ \theta \]

\[ \vartheta \]

\[ \epsilon \]

\[ \zeta \]

\[ \eta \]

\[ \theta \]

\[ \vartheta \]

\[ \epsilon \]

\[ \zeta \]

\[ \eta \]

\[ \theta \]

\[ \vartheta \]

\[ \epsilon \]

\[ \zeta \]

\[ \eta \]

\[ \theta \]

\[ \vartheta \]

\[ \epsilon \]

\[ \zeta \]

\[ \eta \]

\[ \theta \]

\[ \vartheta \]

\[ \epsilon \]

\[ \zeta \]

\[ \eta \]

\[ \theta \]

\[ \vartheta \]
The sum of differences at different time offsets is
\[
\min_{\Delta t} \sum_{i} \| \xi^{(j)}_{t_i+\Delta t} - \xi^{(k)}_{t_i} \|.
\] (18)

For that, we determine the camera-based angular velocity using the visual odometry algorithm based on Lucas Kanade feature tracks without IMU and IR. The computed time offsets are 100ms/166ms for the IR/IMU readings. Time synchronization is vital for short-term odometry, unlike windowed optimization, where keyframes are more distant.

VI. EVALUATION AND EXPERIMENTS

We conduct experiments on an implementation of the approach described in Section IV. In Subsection VI-C we examine the influence of weight distribution on residuals, and in Subsection VI-B we discuss the advantages of utilizing robust loss functions. Subsection VI-A and VI-D are devoted to the evaluation of our method with our MITI dataset. In doing that, we measure the deviation of the algorithms output to ground-truth poses obtained by IR tracking.

We used several libraries: OpenCV for keypoint detection, feature tracking, and reprojection; OpenGV [40] for triangulation, RANSAC, and nonlinear refinement; as well as Ceres [45] and Sophus [46] for solving the optimization problem. For the experiment, we took the image sequence starting from timestamp 637273ms, which contains a fast movement from left to right with dynamic occlusions and enables us to demonstrate the algorithm’s performance in a scenario where standard SLAM approaches lose tracking and can not relocalize in the unseen environment.

A. Evaluation of Visual Odometry Tracking

In Fig. 4, one can inspect the debug view of the output and Fig. 4b and Fig. 4c show huge deviations for different settings in the algorithm. The isolated bundle of lines in Fig. 4b shows: without movement constraints and IMU measurements, the optimization is unable to converge to the correct poses since it loses tracking of 2D features in between keyframes. We computed the Log distance between IR tracked and constrained visual-inertial odometry (VO) estimated camera poses
\[
\xi = [\omega, \upsilon]^T = [\xi^{(IR)} \otimes \xi^{(VO)}] \in \mathbb{R}^6
\] (19)

for variants of the computation steps in Algorithm 1 that are listed in Table I. V1 integrates IMU sensor-readings in Line 2 into pose estimates as given in (12). Therefore, it cannot detect changes in the x-translation component, resulting in a large translational error as soon as the camera moves inside. V2 only uses the camera-based pose estimation of Lines 3 and 4 but without taking the initialization provided by the IMU in Line 2. The result of this variant can additionally be inspected in the debug view of Fig. 4b. V3 uses both camera-based and IMU-based tracking, which already leads to a significant decrease in both error residuals. Finally, in V4, instead of considering the pivot point given by the priorly computed position from IR tracking, we initialize and iteratively refine the pose purely based on visual odometry. The result is an essential step for the goal an eternal inside-out tracking.

TABLE I: Settings of Alg. 1 for results shown in Fig. 5

| Variant | Pivot point online | Line 2 | Line 3 and 4 | Optimization |
|---------|-------------------|--------|--------------|--------------|
| V1      | ✗                 | ✗      | ✗            | ✗            |
| V2      | ✗                 | ✔      | ✔            | ✗            |
| V3      | ✗                 | ✔      | ✔            | ✔            |
| V4      | ✔                 | ✔      | ✔            | ✗            |

B. Error Analysis and Choice of Robustification Functions

The residuals themselves are composed of errors in sensor readings and the deviation of the current pose estimate. Aiming to minimize only the part indicating a pose deviation, we examine errors of sensor readings and their propagation
Expectation
Unit
0.0511 cm
4.2801
0.0051 rad
0.4374
9.81 µm

(5)
Pixel

9
Variance
µ

0.0405
0.0235
(3)
(6)
1.6263

k
values indicate that this
with evaluations before and after optimization. Negative
different residual types in one optimization step for different
∆
decrease. The cost function difference
cost emerging from different sensors should simultaneously
γ
of
from the ground-truth IR tracked poses for different values
10. Fig. 6a and 6b show the deviation of the SLAM algorithm
ovation being triggered every 10 keyframes and a window size of
γ
with

N
∈

1
reprojection residuals per keyframe.

[0

f

, 207]. We conduct experimental runs with optimiza-

f

(20)

various

D. Evaluation of Trade-off and Optimization
A trade-off between gyroscope and camera-based residuals is
adjusted by \( \beta_k = \gamma \) for residual \( k \) belonging to gyroscope
measurements and \( \beta_k = (1 - \gamma) \) for reprojection residuals,
with \( \gamma \in [0, 1] \). We conduct experimental runs with optimization
being triggered every 10 keyframes and a window size of
10. Fig. 6a and 6b show the deviation of the SLAM algorithm
from the ground-truth IR tracked poses for different values
of \( \gamma \). Since we want to jointly optimize over all residuals, the
cost emerging from different sensors should simultaneously
decrease. The cost function difference \( \Delta f \) is

\[
\Delta f = f(\mathbf{T}_{n;m}^{(after)}) - f(\mathbf{T}_{n;m}^{(before)}),
\]
with evaluations before and after optimization. Negative
values indicate that this \( k \)-th part of the cost function was
minimized. In Fig. 6c we show the average cost difference
for different residual types in one optimization step for different
values of \( \gamma \). It exhibits a simultaneous cost decrease for
values close to \( \gamma = 0.4 \).

On the other hand, we observe by inspecting the error
trajectories in Fig. 6a and 6b that for values close to 0.5 the
optimization successfully incorporates the advantages of both
sensors: The rotational error is low at the beginning, which
is typical for camera-based tracking, but also stays down,
which is possible due to utilizing the information of IMU
sensors. The distribution of weights shows great potential in
influencing the overall outcome of the algorithm. An optimal
adjustment however, is a dynamic weight distribution, de-
pending on the current conditions that influence the reliability
of the sensor data.

VII. DISCUSSION
Movement constraints and additional sensor information
are necessary to provide stable pose estimates in dynamic
environments. Absolute references are provided by the earth’s
gravity and magnetic field, whereas gyroscope and Lucas-
Kanade tracks provide the odometry. For a more robust setup,
however, descriptor matching of rigid features can improve
the results of long-term SLAM.

VIII. CONCLUSIONS
This work shows that visual-inertial algorithms can be
applied to dynamically challenging fields such as minimally
invasive surgery. This sets a new paradigm in the applicabil-
ity of SLAM. The gain of reliability is due to the fact that
we use inherent motion constraints. Our results show for
the use-case of minimally invasive surgery, that self localization
is indeed feasible and hope that researchers can profit from
our publicly available dataset.
[43] Z. Zhang, “A flexible new technique for camera calibration,” Trans. on pattern analysis and machine intelligence, vol. 22, no. 11, pp. 1330–1334, 2000.
[44] V. Renaudin, M. H. Afzal, and G. Lachapelle, “Complete triaxis magnetometer calibration in the magnetic domain,” Journal of sensors, vol. 2010, pp. 1–10, 2010.
[45] S. Agarwal and K. Mierle, “Ceres solver: Tutorial & reference,” Google Inc, vol. 2, no. 72, p. 8, 2012.
[46] H. Strasdat, S. Lovegrove, and Others, “Sophus library,” https://github.com/strasdat/Sophus