



Article

Vision-Aided Localization and Mapping in Forested Environments Using Stereo Images

Lucas A. Wells ^{*,†}  and Woodam Chung 

Department of Forest Engineering, Resources and Management, College of Forestry, Oregon State University, Corvallis, OR 97331, USA; woodam.chung@oregonstate.edu

* Correspondence: lucas@silvxlabs.com

† Current address: SilvX Labs, Missoula, MT 59802, USA.

Abstract: Forests are traditionally characterized by stand-level descriptors, such as basal area, mean diameter, and stem density. In recent years, there has been a growing interest in enhancing the resolution of forest inventory to examine the spatial structure and patterns of trees across landscapes. The spatial arrangement of individual trees is closely linked to various non-monetary forest aspects, including water quality, wildlife habitat, and aesthetics. Additionally, associating individual tree positions with dendrometric variables like diameter, taper, and species can provide data for highly optimized, site-specific silvicultural prescriptions designed to achieve diverse management objectives. Aerial photogrammetry has proven effective for mapping individual trees; however, its utility is limited due to the inability to directly estimate many dendrometric variables. In contrast, terrestrial mapping methods can directly observe essential individual tree characteristics, such as diameter, but their mapping accuracy is governed by the accuracy of the global satellite navigation system (GNSS) receiver and the density of the canopy obstructions between the receiver and the satellite constellation. In this paper, we introduce an integrated approach that combines a camera-based motion and tree detection system with GNSS positioning, yielding a stem map with twice the accuracy of using a consumer-grade GNSS receiver alone. We demonstrate that large-scale stem maps can be generated in real time, achieving a root mean squared position error of 2.16 m. We offer an in-depth explanation of a visual egomotion estimation algorithm designed to enhance the local consistency of GNSS-based positioning. Additionally, we present a least squares minimization technique for concurrently optimizing the pose track and the positions of individual tree stem[s].



Citation: Wells, L.A.; Chung, W. Vision-Aided Localization and Mapping in Forested Environments Using Stereo Images. *Sensors* **2023**, *23*, 7043. <https://doi.org/10.3390/s23167043>

Academic Editor: Stefano Berretti

Received: 14 July 2023

Revised: 4 August 2023

Accepted: 7 August 2023

Published: 9 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: direct visual odometry; simultaneous localization and mapping; non-linear least squares; computer vision; GNSS/GPS

1. Introduction

The widespread adoption of sustainable forestry practices in much of the developed world has given rise to increasingly complex forest management objectives catering to a diverse array of interests. Consequently, silvicultural prescriptions often take into account numerous non-monetary factors, including forest resiliency and adaptability, wildlife conservation, aesthetic preservation, hydrological values, and other ecological functions that are dependent on stand structure [1]. In general, the incorporation of non-timber values into management objectives has served as the primary driver for transitioning silvicultural practices from homogeneous, even-aged systems to heterogeneous, uneven-aged systems. This paradigm shift directly affects forest operations; implementing complex silvicultural prescriptions becomes more costly in terms of layout, harvesting, and administration [2]. This challenge has spurred the demand for advanced precision forestry tools that offer accurate, real-time machine positioning, as well as forest measurement and mapping capabilities. Such tools have the potential to eliminate certain layout tasks, such as individual tree and boundary marking, thereby reducing operational costs and enhancing the economic feasibility of alternative silvicultural systems.

Real-time machine positioning and forest mapping are essential components of sustainable forest harvesting [3]. However, practical low-cost technologies for precise localization and large-scale mapping in forested environments during harvesting remain underdeveloped. Unlike the substantial advancements in automation introduced in agricultural systems [4], forest harvesting has not experienced groundbreaking technological progress in terms of automation and robotics. This lag can be attributed, in part, to the challenging environments in which forest machines operate, as well as potential cultural factors, resistance to change, or inadequate research and development efforts.

In this paper, we introduce visual-processing algorithms that facilitate precise real-time machine positioning and forest mapping. We employ visual egomotion estimation from a stereo camera to maintain the position of a forestry machine during instances of degraded or failed reception from a global navigation satellite system (GNSS). Furthermore, we demonstrate how precise position maintenance can be utilized to generate tree maps when combined with a tree-detection algorithm. Below, we itemize the main contributions of this work:

- A novel camera-based system that integrates GNSS, visual odometry, and a tree-detection system and egomotion estimation.
- A low-cost system using an off-the-shelf stereo camera and a consumer-grade GNSS receiver.
- An efficient pixel-selection method with predictable inter-frame runtime for direct image alignment.
- A global orientation parameter in the optimization framework for preliminary alignment between the GNSS and visual odometry pose track.
- A system that provides the tree position and dendrometric information to the user in real time.

In the remainder of this introduction, we will discuss the crucial role of GNSS in forest operations, along with the associated challenges and limitations. Additionally, we will introduce the concepts of visual odometry (VO) and simultaneous localization and mapping (SLAM).

1.1. Global Positioning

Global navigation satellite systems (GNSSs) have played a significant role in forest resource management. These systems have been employed for inventory plot localization [5], forest traverse surveys [6], mapping forest disturbances [7], machine tracking [8,9], automated time studies [10,11], and general operational monitoring [12]. Recently, GNSS technology has been applied to enhance occupational safety in logging operations by providing virtual geofences around workers on the job site [13–15]. Most modern forest machines come equipped with a GNSS receiver from the factory, enabling real-time positioning of the machine during operation.

While highly accurate GNSS positioning may not be essential for many forestry applications, it becomes crucial when constraining machines with a virtual boundary, particularly when boundaries coincide with ownership demarcations or delineate high-value, sensitive, or hazardous areas. High-precision localization is also necessary for mapping individual trees. Challenges with GNSS accuracy have impeded the widespread adoption of GNSS technology in the virtual boundary and mapping domains.

GNSS accuracy depends on various factors, one of which is the number and geometry of visible satellites. This is characterized by an index called position dilution of precision (PDOP), which is a multiplicative term that scales the expected accuracy of the receiver. PDOP values less than 1 provide the highest possible accuracy, while PDOP values greater than 20 generally render coordinate readings futile, e.g., a PDOP value of 20 with a GNSS receiver capable of 3 m accuracy results in an actual accuracy of 60 m. The forest canopy can block signals from reaching the receiver, an effect known as the canopy effect [6]. Another important factor—primarily responsible for degraded GNSS accuracy under forest canopy—is signal diffraction and reflection, known as multipath errors [16]. Lastly, GNSS

accuracy also depends on the class of receiver, such as survey-grade or consumer-grade receivers. Research on GNSS accuracy in forested environments has shown consumer-grade receivers to have an accuracy range between 4 and 12 m [17–21]. In this work, we utilize a specific GNSS called the global positioning system (GPS), operated by the United States Air Force. The GPS satellite constellation comprises 33 satellites, 31 of which are currently operational.

1.2. Visual Odometry

Estimating the egocentric motion of a camera is a fundamental task in computer vision. Egomotion estimation aims to determine a 3D geometric transformation that describes the incremental translational and rotational change of a camera in motion relative to the observed environment. Motion estimation occurs in discrete time steps, where a new estimate at time t reveals the camera's motion relative to its pose at time $t - 1$. Each new estimate can be incrementally composed with the previous to produce a pose graph, i.e., a track of position and orientation over time. Visual odometry (VO), coined by Nistér et al. [22], is the term often used to describe time-integrated motion estimates and is analogous to other variants of odometry, e.g., wheeled odometry, where wheel encoders are used to estimate the traveled distance in ground vehicles. Odometric navigation is subject to errors that accumulate over time, eventually leading to positional drift, a well-known property of dead reckoning navigation systems.

Early research on the problem of recovering relative camera poses tackled the problem of structure-from-motion [23,24]. Structure-from-motion is the problem of recovering both 3D structure and camera poses from a set of unordered images. VO is a special case of structure-from-motion, where images are sequenced and pose recovery is the sole objective. To date, most algorithms for VO are based on a feature extraction and matching pipeline [25–28]. These algorithms are known in the vision community as feature-based methods. In general, feature-based methods involve three steps: (1) feature extraction and description using one of many available approaches (e.g., FAST detector [29,30], SIFT detector/descriptor [31] Harris detector [32], Shi and Tomasi detector [33], and SURF detector/descriptor [34]); (2) temporal feature matching, e.g., using mutual consistency or constrained matching [35]; and (3) pose recovery by minimizing the 3D-2D reprojection error using a perspective from n point (PnP) algorithm, e.g., EPnP [36], which is typically embedded in a random sample consensus (RANSAC) scheme [37]. See [38] for an overview of VO fundamentals in the context of feature-based methods.

Recent approaches to VO have migrated away from feature-based approaches due to complexity and the plethora of configurations in feature-based pipelines. Recent approaches use image intensities directly rather than extracting and matching features [39–44]. These methods are called direct methods and are founded on the work by Lucas and Kanade [45] introducing parametric image alignment. In contrast to feature-based methods, which recover relative camera motion by minimizing reprojection error, direct methods minimize the photometric error, the sum of squared differences in image intensities between two consecutive frames.

In this work, we employ a direct method to solve for the incremental 6-DoF motion parameters. We provide a detailed description of image warping and pixel selection techniques, as well as the optimization procedure. Our method is somewhat simplified in contrast to state-of-the-art methods, e.g., [40,43], which involve keyframe selection and windowed bundle adjustment optimization to mitigate trajectory drift. As we will show in the localization and mapping section, we align the odometry track with global coordinates from a GNSS, and therefore, maintaining global consistency within our VO framework is unnecessary. Instead, in this work, we focus on local consistency and computational efficiency.

1.3. Simultaneous Localization and Mapping

SLAM is a critical problem in robotics, which involves constructing a map of an unknown environment while concurrently determining the robot's location within it. If the

robot's position and orientation are known at any given time, mapping its surroundings using sensor data becomes a straightforward task. Conversely, if the map is already known, the problem is reduced to localization, where the robot's observations are utilized to determine its position and orientation within the map. When neither the pose nor the map is known, both the map and path must be estimated simultaneously. SLAM can be broadly classified into online and full SLAM problems. Online SLAM incrementally estimates the current pose along with the map, while full SLAM estimates the entire path and map using data from all poses and observations [46]. There has been research in forest harvesting related to the localization problem [47,48]; however, a stem map is required in order to localize the machine, which we consider to be a major limitation of such an approach since stem maps, in general, are not readily available.

Visual odometry (VO) and vision-based SLAM are closely related, as VO provides an estimate of the robot's path. The key difference is that SLAM mandates maintaining a map—even if it is not ultimately needed—so that the robot can recognize previously explored areas. When the robot revisits an area on the map, it provides additional constraints to optimize the trajectory and map, ensuring global consistency. This process is known as loop closure. Generally, VO is focused on local consistency, while SLAM aims for global consistency. It is worth noting that if VO is free from drift, the SLAM problem is reduced to merely mapping observations. However, error-free VO has never been achieved in practice, making loop closures essential for maintaining global consistency. To recognize loop closures, robots need to operate in environments containing distinct features that can be reidentified upon returning to a previously mapped location.

Detecting loop closures in forested environments can be challenging, as the spatial configurations of features, such as trees, may not be unique enough to provide robust information for redetection. Integrating GNSS positioning with SLAM can assist in identifying loop closures by maintaining a globally consistent position path. In this paper, we demonstrate how intermittent and degraded GPS reception can be fused with VO to provide a globally consistent position estimate. Additionally, we present a graph-based SLAM algorithm for refining the estimated path and map simultaneously. For an introductory tutorial on graph-based SLAM, readers can refer to [49].

1.4. Notation

In our method description below, we denote vectors as bold lowercase letters, e.g., \mathbf{v} , matrices as bold capital letters, e.g., \mathbf{M} , and scalars as lowercase italic letters, e.g., s . We use the notation $\|\cdot\|$ as a shorthand for $\|\cdot\|_2$, i.e., the Euclidean norm. We represent images as functions, $\mathcal{I} : \Omega \rightarrow \mathbb{R}^3$ for 3-channel color images, and $\mathcal{I} : \Omega \rightarrow \mathbb{R}$ for gray-scale images where $\Omega \subset \mathbb{R}^2$ is the image domain. Sets are denoted by capital script letters, e.g., \mathcal{A} , and the number of elements in a set is given by $|\mathcal{A}|$. See Appendix A for an overview of Lie groups and rigid transformations used in this paper.

2. Direct Visual Odometry

Given a reference image $\mathcal{I}_{t-1} : \Omega \rightarrow \mathbb{R}$ acquired at time $t - 1$ and an input image $\mathcal{I}_t : \Omega \rightarrow \mathbb{R}$ acquired at time t , we seek to estimate the 3D egomotion of the camera between the frames. Estimating the camera's motion is performed by solving for the parameters of a *warp*, $\xi_{t-1:t}$, which relates the pixels in \mathcal{I}_{t-1} to the pixels in \mathcal{I}_t . For brevity, we drop the time subscript on the warping parameters and denote it by ξ , and denote the reference image by \mathcal{I} and the input image by \mathcal{I}' . The intensity of a pixel in the reference image is given by $\mathcal{I}(\mathbf{p})$, where $\mathbf{p} = (u, v)^\top \in \Omega$. Similarly, the intensity of a pixel in the input image is given by $\mathcal{I}'(\mathbf{p}')$, where the position vector \mathbf{p}' is the result after warping \mathbf{p} according to the parameters ξ ,

$$\mathbf{p}' = \mathcal{W}(\mathbf{p}; \xi). \quad (1)$$

Leaving the warping function undefined for the moment, the images are registered, or aligned, by minimizing the photometric error according to the following objective:

$$\min_{\xi} \sum_{\mathbf{p} \in \Phi} \left\| \mathcal{I}(\mathbf{p}) - \mathcal{I}'(\mathcal{W}(\mathbf{p}; \xi)) \right\|^2, \quad (2)$$

where $\Phi \subset \Omega$ is a set of selected pixels from the image domain. Solving this expression is non-linear regardless of the warping function, as pixel intensities are unrelated to their coordinates. A common method for minimizing the objective is the Gauss–Newton (GN) algorithm. In general, the expression

$$\sum_{\mathbf{p} \in \Phi} \left\| \mathcal{I}(\mathbf{p}) - \mathcal{I}'(\mathcal{W}(\mathbf{p}; \xi + \Delta\xi)) \right\|^2, \quad (3)$$

is linearized w.r.t. some small change in the parameters, $\Delta\xi$, and then the parameters are updated by

$$\xi_{\kappa+1} = \xi_{\kappa} + \Delta\xi, \quad (4)$$

until ξ and $\Delta\xi$ converge.

This formulation, as well as a solution procedure, was first given by Lucas and Kanade [45] in their seminal work, on which parametric image alignment, and subsequently direct VO, is founded. For a detailed presentation of optimization algorithms and alternative formulations, see Baker and Matthews [50]. We will discuss our optimization technique in a latter section after introducing the warping function.

2.1. Image Warping

The egomotion of a camera with no assumed holonomic constraints has 6 degrees of freedom; rotation about the x , y and z axes and translation along the x , y and z axes. The motion according to these degrees of freedom is represented by a transformation matrix in the special Euclidean Lie group:

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \text{SE}(3), \quad (5)$$

where $\mathbf{R} \in \text{SO}(3)$ is a 3D rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is a 3D displacement or translation vector. This is a rigid-body transformation that encodes the change in rotation and translation of a non-deformable object in motion between two discrete points in time. Although the matrix \mathbf{T} has 6 degrees of freedom, there are 12 values that make up the non-homogeneous portion of the matrix: 9 values in the rotation matrix and 3 values for translation. This imposes unnecessary computational demands in an optimization setting. For this reason, we use the Lie algebra $\mathfrak{se}(3)$ to parameterize the transformation. We denote the transformation matrix as a function of the parameters $\xi \in \mathfrak{se}(3)$,

$$\mathbf{T}(\xi) = \exp(\hat{\xi}) \in \text{SE}(3). \quad (6)$$

The exponential map of $\mathfrak{se}(3)$ can be computed in closed form, as can the logarithm map that takes $\text{SE}(3)$ back to the algebra $\mathfrak{se}(3)$,

$$\xi = \ln(\mathbf{T}(\xi)) \in \mathfrak{se}(3). \quad (7)$$

Using the Lie algebra parameterization of the transformation, we define a warping function $\mathcal{W} : (\mathbb{R}^3 \times \mathbb{R}^6) \rightarrow \mathbb{R}^2$ that takes a homogeneous pixel coordinate as an input, back projects

it to \mathbb{R}^3 , transforms the back projection according to the warping parameters, and then projects it back to \mathbb{R}^2 ,

$$\mathcal{W}(\tilde{\mathbf{p}}; \boldsymbol{\xi}) = \pi\left(\mathbf{T}(\boldsymbol{\xi})\pi^{-1}(\tilde{\mathbf{p}}, z)\right), \quad (8)$$

where the notation $\tilde{\mathbf{p}} = (u, v, 1)^\top \in \mathbb{P}^2$ denotes a pixel position vector in homogeneous coordinates. The function $\pi^{-1}(\cdot)$ performs the back-projection and is defined as

$$\pi^{-1}(\tilde{\mathbf{p}}, z) = \left((z\mathbf{K}^{-1}\tilde{\mathbf{p}})^\top, 1 \right)^\top \in \mathbb{R}^4, \quad (9)$$

where z is a depth measurement for the pixel and \mathbf{K}^{-1} is the inverse of the projection matrix. Note that we homogenized, i.e., appended a fictitious coordinate, to the back-projected vector so that it is compatible with the transformation matrix. The transformation matrix is not homogeneous, i.e., the last row of the matrix shown in Equation (5) is omitted, so the resulting vector following the transformation has three dimensions. The function that performs the forward projection is defined as

$$\pi(\mathbf{x}) = \tilde{\mathbf{n}}(\mathbf{K}\mathbf{x}) \in \mathbb{R}^2, \quad (10)$$

where $\mathbf{x} = (x, y, z)^\top \in \mathbb{R}^3$, \mathbf{K} is the camera projection matrix, and $\tilde{\mathbf{n}}: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ normalizes the homogeneous coordinates, i.e., $\tilde{\mathbf{n}}\left((x, y, w)^\top\right) = (x/w, y/w)^\top$. Finally, the projection matrix and its inverse are defined as

$$\mathbf{K} = \begin{pmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{K}^{-1} = \begin{pmatrix} \frac{1}{f} & 0 & -\frac{c_u}{f} \\ 0 & \frac{1}{f} & -\frac{c_v}{f} \\ 0 & 0 & 1 \end{pmatrix}, \quad (11)$$

where f is the focal length, assuming unit aspect ratio pixels, and (c_u, c_v) is the principal point.

The only variable that still needs attention is the depth measurement z used in the back-projection function. We estimate the depth of each pixel by first computing the stereo correspondence via semi-global matching [51]. We use a real-time GPU implementation presented in [52]. Due to the smoothness constraints imposed by the semi-global matching scheme, occlusions are filled by some non-zero value. We handle occlusions by right-left consistency check: The disparity map $\mathcal{D}^{\ell,r}$ is computed first, then the second disparity map $\mathcal{D}^{r,\ell}$ by reflecting the left and right images along the v -axis and using the right image in place of the left, and the left in place of the right. The final disparity map \mathcal{D} is equal to $\mathcal{D}^{\ell,r}$ when the absolute difference between $\mathcal{D}^{\ell,r}$ and $\mathcal{D}^{r,\ell}$, evaluated at a position vector \mathbf{p} , does not exceed some threshold δ . Otherwise, $\mathcal{D}(\mathbf{p})$ takes zero,

$$\mathcal{D}(\mathbf{p}) = \begin{cases} \mathcal{D}^{\ell,r}(\mathbf{p}) & \text{if } |\mathcal{D}^{\ell,r}(\mathbf{p}) - \mathcal{D}^{r,\ell}(\mathbf{p})| \leq \delta, \\ 0 & \text{otherwise} \end{cases}, \quad \forall \mathbf{p} \in \Omega. \quad (12)$$

In this work, we use $\delta = 1$. We do not perform sub-pixel refinement to interpolate disparity values. We simply use positive integers to represent the disparity map, $\mathcal{D}: \Omega \rightarrow \mathbb{N}$. The depth estimate, z , for the pixel \mathbf{p} is given via triangulation,

$$z(\mathbf{p}) = \frac{fb}{\mathcal{D}(\mathbf{p})}, \quad (13)$$

where f is the focal length and b is the baseline distance of the stereo rig in centimeters. This assumes that the stereo camera is calibrated and row-aligned. We follow methods presented in [53] for camera calibration.

2.2. Optimization

We will now extend our discussion regarding the minimization of photometric error. As noted earlier, minimizing the expression defined in Equation (2) is a non-linear optimization task. This formulation requires a linearization step during each GN iteration. Namely, the Jacobian of the warp and the Hessian need to be computed during each iteration, which can lead to significant computational demands depending on the size of the Jacobian. Following from Baker and Matthews [50], we redefine the objective under the inverse compositional (IC) formulation by interchanging the roles of the reference and input image and solving for incremental warp parameters instead of additive updates as in the Lucas–Kanade formulation. Given some initial guess of the parameters, ξ , the objective is to minimize the following expression w.r.t the incremental warp parameters,

$$\Delta\xi^* = \operatorname{argmin}_{\Delta\xi} \sum_{\mathbf{p} \in \Phi} \left\| \mathcal{I}(\mathcal{W}(\tilde{\mathbf{p}}; \Delta\xi)) - \mathcal{I}'(\mathcal{W}(\tilde{\mathbf{p}}; \xi)) \right\|^2. \quad (14)$$

The parameters are updated by inverting the incremental warp parameters and composing with the current estimate, $\xi \leftarrow \xi \circ \Delta\xi^{-1}$, where the notation \circ denotes composition. The incremental warp parameters need to be inverted at each GN iteration as the linearization, which we will discuss next, is performed on the reference image. The update rule can be explicitly written as

$$\begin{aligned} \xi_{\kappa+1} &= \xi_{\kappa} \circ \Delta\xi^{-1} \\ &= \ln\left(\exp(\hat{\xi}_{\kappa}) \exp(-\Delta\hat{\xi})\right) \\ &= \ln\left(\begin{array}{cc} \mathbf{R}_{\kappa} \Delta\mathbf{R}^{\top} & \mathbf{R}_{\kappa}(-\Delta\mathbf{R}^{\top} \Delta\mathbf{t}) + \mathbf{t}_{\kappa} \\ \mathbf{0}^{\top} & 1 \end{array}\right). \end{aligned} \quad (15)$$

According to the GN algorithm, the incremental warp parameters $\Delta\xi$ are given by the normal equations,

$$\mathbf{J}^{\top} \mathbf{J} \Delta\xi = -\mathbf{J}^{\top} \mathbf{r} \implies \Delta\xi = -(\mathbf{J}^{\top} \mathbf{J})^{-1} \mathbf{J}^{\top} \mathbf{r}, \quad (16)$$

where \mathbf{J} is a $m \times 6$ Jacobian matrix, $\mathbf{J}^{\top} \mathbf{J}$ is the Gauss–Newton approximation of the Hessian matrix, and \mathbf{r} is the vector of residuals given by

$$\mathbf{r} = \mathcal{I}'(\mathcal{W}(\tilde{\mathbf{p}}; \xi)) - \mathcal{I}(\mathbf{p}). \quad (17)$$

The Jacobian encodes the partial derivatives of the reference image at each pixel $\mathbf{p}_{\{i\}_1^m}$ with respect to the six warping parameters,

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \mathcal{I}(\mathbf{p}_1)}{\partial \xi}^{\top} \\ \frac{\partial \mathcal{I}(\mathbf{p}_2)}{\partial \xi}^{\top} \\ \vdots \\ \frac{\partial \mathcal{I}(\mathbf{p}_m)}{\partial \xi}^{\top} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathcal{I}(\mathbf{p}_1)}{\partial \xi_1} & \frac{\partial \mathcal{I}(\mathbf{p}_1)}{\partial \xi_2} & \cdots & \frac{\partial \mathcal{I}(\mathbf{p}_1)}{\partial \xi_6} \\ \frac{\partial \mathcal{I}(\mathbf{p}_2)}{\partial \xi_1} & \frac{\partial \mathcal{I}(\mathbf{p}_2)}{\partial \xi_2} & \cdots & \frac{\partial \mathcal{I}(\mathbf{p}_2)}{\partial \xi_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{I}(\mathbf{p}_m)}{\partial \xi_1} & \frac{\partial \mathcal{I}(\mathbf{p}_m)}{\partial \xi_2} & \cdots & \frac{\partial \mathcal{I}(\mathbf{p}_m)}{\partial \xi_6} \end{pmatrix}. \quad (18)$$

The linearization of Equation (14) can be achieved by performing a first-order Taylor expansion about the current estimate of the parameters. Denoting the i th row in the Jacobian as $\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \xi}$, corresponding to some pixel $\mathbf{p} \in \Phi$ and applying the chain rule, we obtain

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \xi}^{\top} = \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \xi}. \quad (19)$$

The partial derivative of the reference image w.r.t. some pixel position \mathbf{p} is simply the gradient vector of the image along the u and v axes,

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}} = \nabla \mathcal{I}(\mathbf{p}) = (\mathcal{I}_u(\mathbf{p}), \mathcal{I}_v(\mathbf{p})). \quad (20)$$

We purposefully denoted the gradient vector as a row vector. Taking \mathbf{p} to be equal to a reduced form of the forward projection function $\pi(\cdot)$ such that $\mathbf{p} = (fx/z + c_u, fy/z + c_v)^\top$, and $\mathbf{x} = (x, y, z)^\top$ to be a back projection of the pixel, the partial derivative can be written as

$$\frac{\partial \mathbf{p}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{f}{z} & 0 & -f \frac{x}{z^2} \\ 0 & \frac{f}{z} & -f \frac{y}{z^2} \end{pmatrix}. \quad (21)$$

Finally, the partial derivative of the back-projected pixel \mathbf{x} w.r.t. the warping parameters evaluated at the identity warp $\boldsymbol{\xi} = \mathbf{0}$ takes the form,

$$\left. \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\mathbf{0}} = ([\mathbf{x}]_{\times} \mathbf{I}_3) = \begin{pmatrix} 0 & -z & y & 1 & 0 & 0 \\ z & 0 & -x & 0 & 1 & 0 \\ -y & x & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (22)$$

This result follows from the skew-symmetric matrix operator used when computing the SE(3) exponential map of the warp parameters. Multiplying out the last two partials gives

$$\begin{aligned} \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \boldsymbol{\xi}}^\top &= \nabla \mathcal{I}(\mathbf{p}) \frac{\partial \mathbf{p}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \\ &= (\mathcal{I}_u(\mathbf{p}), \mathcal{I}_v(\mathbf{p})) \begin{pmatrix} f \frac{xy}{z^2} & f \frac{x^2 - z^2}{z^2} & \frac{xy}{z} & \frac{f}{z} & 0 & -f \frac{x}{z^2} \\ f \frac{y^2 + z^2}{z^2} & -f \frac{xy}{z^2} & -f \frac{x}{z} & 0 & \frac{f}{z} & -f \frac{y}{z^2} \end{pmatrix}. \end{aligned} \quad (23)$$

For convenience, we also show a row in the Jacobian corresponding to some pixel \mathbf{p} written out explicitly,

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \frac{\mathcal{I}_u(\mathbf{p})fxy + \mathcal{I}_v(\mathbf{p})(fy^2 + fz^2)}{z^2} \\ \frac{\mathcal{I}_u(\mathbf{p})(-fx^2 - fz^2) - \mathcal{I}_v(\mathbf{p})fxy}{z^2} \\ \frac{\mathcal{I}_u(\mathbf{p})fy - \mathcal{I}_v(\mathbf{p})fx}{z} \\ \frac{\mathcal{I}_u(\mathbf{p})f}{z} \\ \frac{\mathcal{I}_v(\mathbf{p})f}{z} \\ \frac{-\mathcal{I}_u(\mathbf{p})fx - \mathcal{I}_v(\mathbf{p})fy}{z^2} \end{pmatrix}^\top \in \mathbb{R}^6. \quad (24)$$

The row vector stated above is computed for each pixel $\mathbf{p} \in \Phi$ and stacked into the $m \times 6$ Jacobian matrix, again where m is the number of pixels in Φ . The columns of the Jacobian can be visualized as the steepest descent images shown in Figure 1.

Since the linearization is performed on the coordinate frame of the reference image, the Jacobian \mathbf{J} and the GN approximation of the Hessian $\mathbf{H} = \mathbf{J}^\top \mathbf{J}$, as well as its inverse \mathbf{H}^{-1} only need to be computed once. These are the computational savings of the IC formulation [50] over the original Lucas–Kanade algorithm [45].

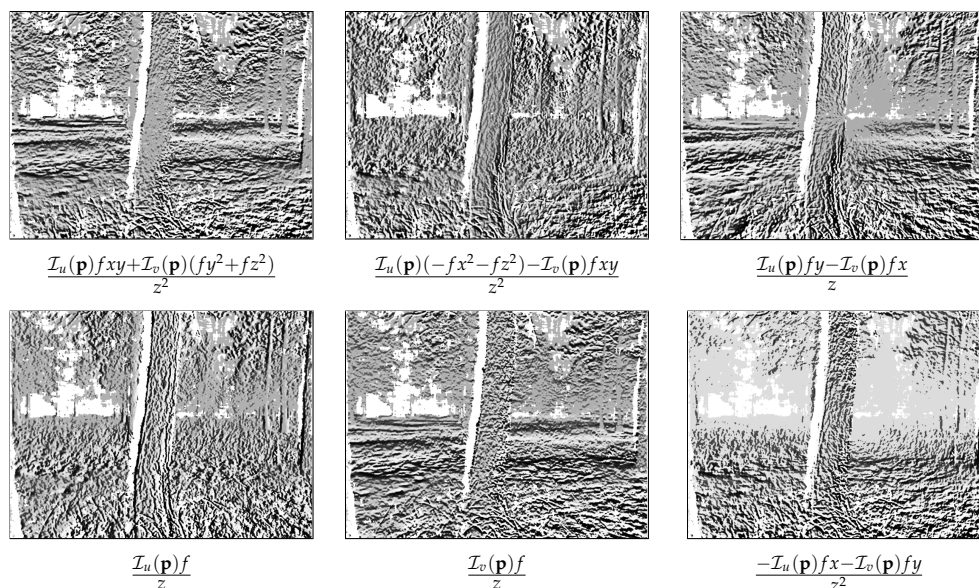


Figure 1. Steepest descent images; each image corresponds to a column in the Jacobian matrix.

2.2.1. Pixel Selection

Since minimizing photometric error is based on image gradients, only pixels with a non-zero gradient contribute to optimization. Therefore, including pixels with zero gradient in the Jacobian introduces unnecessary computations. In feature-rich environments, such as forests, selecting pixels with a non-zero gradient might not significantly reduce the number of pixels used in optimization. As shown in Figure 2a, selecting all non-zero gradient pixels results in using approximately 85% of the image. We can further reduce the number of pixels, thus decreasing computation, by thresholding pixels based on gradient magnitude. This approach, however, results in a Jacobian matrix that is subject to change in dimension between frames since the distribution of gradient magnitude is not guaranteed to be consistent; we must allocate enough memory to store the expected maximum number of non-zero gradient pixels across all frames, which cannot be determined in advance, or dynamically allocate memory prior to optimizing each frame. This issue can be resolved by selecting a percentage of the total number of pixels in the image either by performing binary search for a gradient magnitude threshold that results in the desired number of pixels or sorting the gradients in descending order and selecting the first $N \frac{p}{100}$ pixels, where p is the desired percentage and N is the number of pixels in the image. In Figure 2b–d, we show the selected pixels resulting from desired percentages ranging from 25% to 75%.

Prior to selecting pixels based on the gradient magnitude, we discard all pixels with a disparity value of zero since these pixels are projected to infinity during image warping. Zero disparity pixels are apparent in Figure 2 as areas in the images that obviously have non-zero gradients but are too distant to have a non-zero disparity value.

2.2.2. Robustness

GN optimization assumes Gaussian distributed errors. It is often the case in real-world data, however, that non-Gaussian errors arise due to inaccurate pixel correspondences during disparity computation, motion-induced occlusions, illumination changes and auto-exposure adjustments. A cost function for minimizing photometric error that is insensitive to non-Gaussian distributed noise is said to be robust. In the GN framework, this can be achieved with iterative re-weighted least squares (IRLS) using a robust cost function. The decision of the cost function is somewhat arbitrary, typically selected through empirical evaluation or by means of some prior knowledge regarding the structure of outliers in the data. In this work, we choose to use Tukey's biweight cost function [54,55] since it suppresses large residuals in contrast to Huber's cost function [56] which simply down weights their influence.

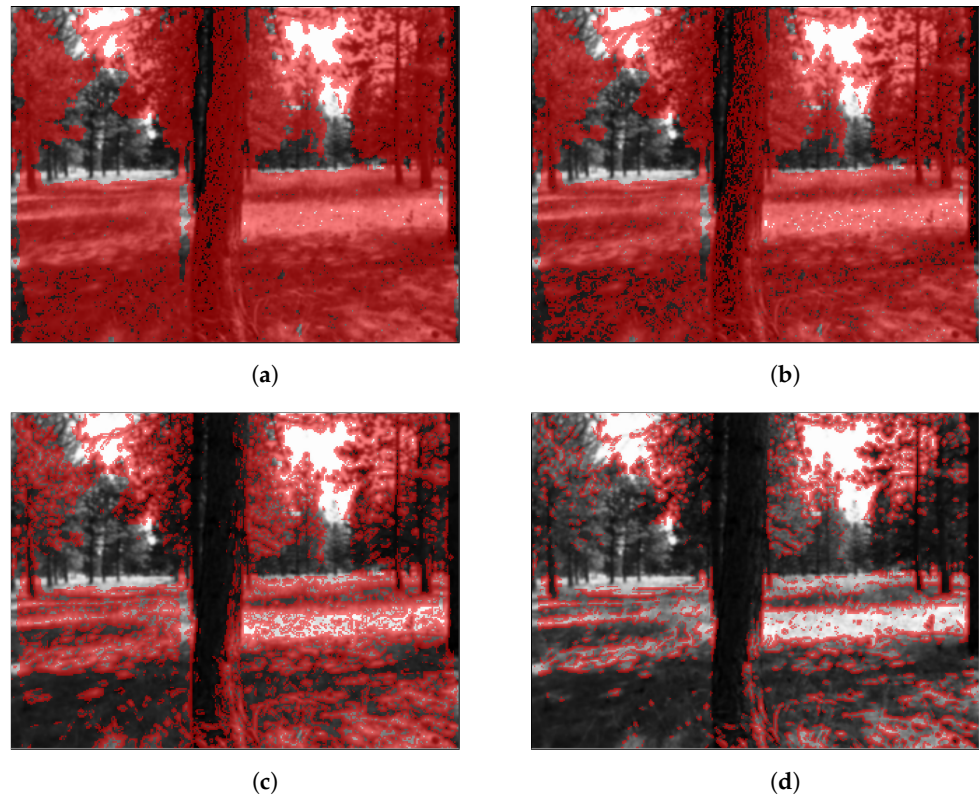


Figure 2. Gradient magnitude-based pixel selection; selected pixels shown in red. Subfigure captions indicate the number of selected pixels for 240×320 resolution images. (a) Dense ($\approx 65,000$ px); (b) 75% ($\approx 57,600$ px); (c) 50% ($\approx 38,400$ px); (d) 25% ($\approx 19,200$ px).

Tukey's biweight cost function takes the form

$$\rho(r_i) = \begin{cases} \left(1 - \left(\frac{r_i}{c}\right)^2\right)^2 & \text{if } |r_i| \leq c \\ 0 & \text{otherwise} \end{cases}, \quad (25)$$

where the constant c is usually chosen to be 4.6851 to achieve 95% asymptotic efficiency with the normal distribution and r_i is the i th residual from the error vector given by Equation (17). The cost function assumes that the residuals have unit variance. A common choice to estimate the scale parameter to standardize the residuals is to compute the median absolute deviation (MAD) and multiply by the expected MAD for a standard normal distribution,

$$\hat{s} = k \cdot \text{median}_i |r_i|, \quad (26)$$

where $k = 1.4826$. To incorporate robustness in the GN minimization routine, we construct a weight vector \mathbf{w} , where each weight $w_i = \rho(r_i/\hat{s})$ and set the diagonal entries of a weight matrix equal to the weight vector, i.e., $\mathbf{W} = \text{diag}(\mathbf{w})$. It follows that the solution to the incremental update of the linearized system under the IRLS framework takes the form

$$\Delta\boldsymbol{\xi} = -\left(\mathbf{J}^T \mathbf{W} \mathbf{J}\right)^{-1} \mathbf{J}^T \mathbf{W} \mathbf{r}, \quad (27)$$

which is used in place of the normal equations shown in Equation (16).

3. Localization and Mapping

In this section, we present a graph-based algorithm for maintaining a globally and locally consistent pose track using the estimated frame-to-frame egomotion parameters from the previous section and a consumer-grade GPS receiver. We also show how the optimized

pose graph can be refined to generate a map of detected tree stems. Our algorithm consists of two phases: global alignment, in which we minimize errors between the odometry-based pose graph and global positions provided by the GPS receiver, and local refinement, where we relate poses by multiple observations of tree stems and simultaneously optimize the configuration of the pose graph and tree stem positions.

In order to increase the efficiency of computations, we use the $\mathfrak{se}(2)$ Lie algebra of rigid transformations to represent the graph, and ultimately the map, as opposed to the $\mathfrak{se}(3)$ parameters we optimized for in the odometry section. We do this by simply extracting the translation parameters corresponding to the x and z axes in $\mathfrak{se}(3)$ to represent the translation along the x and y axes on a planar pose graph, and the rotation component about the y axis from the $\mathfrak{se}(3)$ parameters to represent the heading. Using $\mathfrak{se}(2)$ parameters produces a meaningful map that can easily be presented on a 2D display monitor.

3.1. Global Alignment

Given a set of frame-to-frame odometry observations, $\{\Delta\xi_1, \Delta\xi_2, \dots, \Delta\xi_m\}$, where each $\Delta\xi_i = (x, y, \theta)^\top \in \mathfrak{se}(2)$, we seek to align a pose graph constructed from the odometry observations with a set of global position readings from a GPS receiver. We assume odometry observations to be locally consistent but subject to drift and assume GPS coordinates to be locally bounded by a Gaussian distribution specified by an arbitrarily large covariance matrix that captures the expected errors due to multi-pass signals and geometric dilution of precision. We denote a GPS coordinate as $\mathbf{g} = (x, y)^\top \in \mathbb{R}^2$, where x and y represent the global position estimate in meters within the Universal Transverse Mercator (UTM) coordinate system. We also convert the translation component of the odometry observations to meters for compatibility with the UTM coordinate frame.

3.1.1. Graph Construction

A pose graph is used to represent the camera poses and the motion constraints between the poses. A node, or vertex, in the graph denotes a pose, i.e., a position and orientation, and an edge denotes the relative motion constraint given by the odometry observation. We use \mathbf{v}_i to denote the i th node in the graph and $\Delta\xi_i$ to denote the relative motion between \mathbf{v}_i and \mathbf{v}_{i+1} . We construct the initial pose graph by sequentially transforming poses with the odometry observations. First, we fix the first node in the graph to the zero vector, then each subsequent pose is computed by right multiplying the previous pose with a homogeneous transformation matrix $\mathbf{T}(\Delta\xi) \in \text{SE}(2)$, representing the exponential map of the odometry observation,

$$\mathbf{v}_1 = (0, 0, 0)^\top, \quad (28a)$$

$$\mathbf{v}_{i+1} = \mathbf{T}(\xi_i)\mathbf{v}_i, \quad \forall i = \{1, 2, \dots, n\}. \quad (28b)$$

To simplify the notation, we use ξ_{ij} to denote the motion constraint between the poses \mathbf{v}_i and \mathbf{v}_j where $\mathbf{v}_j \stackrel{\text{def}}{=} \mathbf{v}_{i+1}$. We also take n to be equal to the number of poses, which is the number of odometry constraints plus one, i.e., $n \stackrel{\text{def}}{=} m + 1$. Associated with each motion constraint is a 3×3 covariance matrix Σ_{ij} that represents the uncertainty of the motion. As we describe in the optimization section that follows, we use the information matrix $\mathbf{Q}_{ij} = \Sigma_{ij}^{-1}$ to represent the strength of the edge, or constraint, in the graph.

Let \mathcal{C} be an ordered set of 2-tuples representing the correspondences between poses and GPS readings. Thus, the tuple $(i, k) \in \mathcal{C}$ specifies that pose \mathbf{v}_i corresponds to the GPS reading \mathbf{g}_k . We insert GPS coordinates as nodes in the graph and add an edge to the corresponding camera pose. We also translate all GPS coordinates according to the first correspondence in \mathcal{C} . We do this by storing the translation, $\mathbf{g}_0 \leftarrow \mathbf{g}_{k \in \mathcal{C}_1}$, where \mathcal{C}_1 is the first GPS-odometry correspondence, and subtracting \mathbf{g}_0 from all coordinates in the track,

$$\mathbf{g}_k \leftarrow \mathbf{g}_k - \mathbf{g}_0 \quad \forall k = \{1, 2, \dots, |\mathcal{C}|\}. \quad (29)$$

We store the translation so that we can invert the pose graph back to the original UTM coordinates after optimization. Associated with each GPS coordinate is a 2×2 covariance matrix representing the expected accuracy of the receiver. We invert to covariance matrix, as we did with the odometry covariance, to obtain an information matrix \mathbf{Q}_k . The values in this matrix depend on the expected accuracy of the GPS receiver and the environment in which the receiver is operating. For example, a consumer-grade GPS device operating under a dense forest canopy will have a relatively large uncertainty and thus small values in the information matrix.

Although we anchor the GPS track to a camera pose in the graph, their global orientation will likely differ. Thus, we introduce a global orientation parameter ϕ as another node in the graph that imposes a constraint on each node representing a camera pose that has a corresponding GPS reading. See Figure 3 for an illustration of the graph.

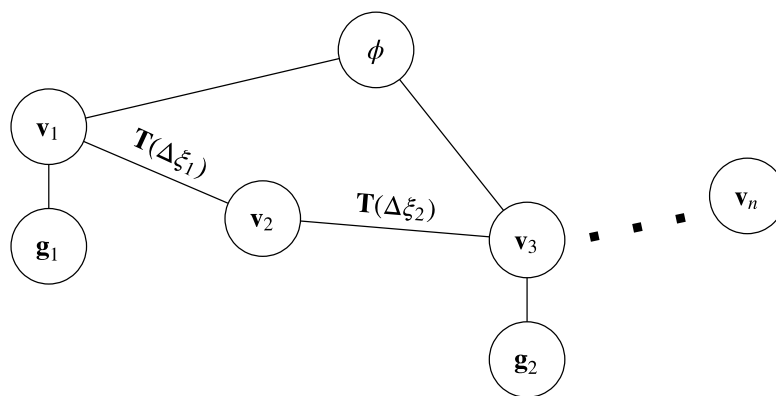


Figure 3. Graphical model of global alignment. Circles denote nodes (vertices) and lines denotes edges (constraints).

3.1.2. Optimization

Given the graph structure outline above, we seek to find the optimal configuration of the state vector,

$$\mathbf{s} = \left(\phi, \mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_n^T \right)^T, \tag{30}$$

that minimizes the sum of squared errors. The state is simply a vector consisting of the global orientation parameter followed by the camera poses, where each pose is parameterized as $\mathbf{v}_i = (x, y, \theta)^T$. Thus, the size of this vector is $1 + dn$, where d is the number of dimensions used to describe a camera pose, in this case 3, and n is the number of camera poses. The errors associated with the state configuration are given by two functions: one corresponding to the relative motion constraints given by the odometry observations and one corresponding to the constraints imposed by the observations from the GPS. The odometry error function, which is equivalent to the error function used in [49], is given by

$$\begin{aligned} \mathbf{r}_{ij}(\mathbf{v}_i, \mathbf{v}_j) &= \mathbf{T}(\boldsymbol{\zeta}_{ij})^{-1} \left(\mathbf{T}(\mathbf{v}_i)^{-1} \mathbf{T}(\mathbf{v}_j) \right), \\ &= \begin{pmatrix} \mathbf{R}(\theta_{ij})^T \left(\mathbf{R}(\theta_i)^T (\mathbf{t}_j - \mathbf{t}_i) - \mathbf{t}_{ij} \right) \\ \theta_j - \theta_i - \theta_{ij} \end{pmatrix}. \end{aligned} \tag{31}$$

where the notation $\mathbf{R}(\theta)$ represents a 2D rotation matrix of the form

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \tag{32}$$

and θ_i, θ_j and θ_{ij} are the rotation angles corresponding to $\mathbf{v}_i, \mathbf{v}_j$ and $\boldsymbol{\zeta}_{ij}$, respectively. The notations $\mathbf{t}_i, \mathbf{t}_j$ and \mathbf{t}_{ij} represent the translation vectors from $\mathbf{v}_i, \mathbf{v}_j$ and $\boldsymbol{\zeta}_{ij}$, respectively. The

error function defined in Equation (31) gives the translational and rotational errors by first transforming pose \mathbf{v}_j into the coordinate frame of \mathbf{v}_i , then computing the differences according to the odometry observation. Thus, this function returns zero when the state vector is configured according to Equations (28a) and (28b).

The second error function calculates the position difference between a pose and its corresponding GPS coordinate according to the global orientation parameter,

$$\mathbf{r}_{ik}(\mathbf{v}_i, \phi) = \mathbf{R}(\phi)\mathbf{t}_i - \mathbf{g}_k. \quad (33)$$

This function rotates the translation vector of pose \mathbf{v}_i about the origin of the coordinate frame according to a rotation matrix constructed from the global orientation parameter ϕ and returns the offset to the corresponding GPS coordinate.

Given the two error functions, we can write the sum of squared errors of the state configuration \mathbf{s} as

$$\epsilon(\mathbf{s}) = \sum_{ij} \mathbf{r}_{ij}^T \mathbf{Q}_{ij} \mathbf{r}_{ij} + \sum_{(i,k) \in \mathcal{C}} \mathbf{r}_{ik}^T \mathbf{Q}_k \mathbf{r}_{ik}. \quad (34)$$

where the error vectors \mathbf{r}_{ij} and \mathbf{r}_{ik} are weighted by the information matrices to incorporate the degree of belief given to the observations. This leads to the following objective function:

$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmin}} \epsilon(\mathbf{s}). \quad (35)$$

This is a non-linear least-squares optimization problem that can be solved with the GN algorithm. Specifically, the error function is linearized about a current estimate of the state configuration and we iteratively solve the linear system and incrementally update the state vector. Since our initial estimate of the state vector might be far from a minimum solution, we use a dampened version of the GN algorithm called the Levenberg–Marquardt (LM) algorithm [57,58]. The linear system that is solved during each LM iteration takes the form

$$(\mathbf{H} + \lambda \mathbf{I})\Delta \mathbf{s} = -\mathbf{b}, \quad (36)$$

where $\Delta \mathbf{s}$ is the solution to the linear system that provides incremental improvements to the state vector in the non-linear solution space by $\bar{\mathbf{s}} \leftarrow \bar{\mathbf{s}} + \Delta \mathbf{s}$, where $\bar{\mathbf{s}}$ is the current estimate of the state configuration. The variable λ is a non-negative damping factor that, when large, forces the update to behave as the steepest descent and, when small, brings the algorithm closer to GN. We initialize $\lambda = \operatorname{trace}(\mathbf{H})$ and divide by two if the objective value decreases and multiply by two if the objective value increases.

To construct the Hessian \mathbf{H} and the gradient vector \mathbf{b} , we first take the derivatives of the error functions with respect to the state parameters evaluated at the current estimated of the state. We initialize the poses in the state vector according to Equations (28a) and (28b), and set the global orientation parameter to $\phi = 0$. The partial derivatives of the odometry error function can be written as

$$\mathbf{A}_{ij} = \left. \frac{\partial \mathbf{r}_{ij}}{\partial \mathbf{v}_i} \right|_{\mathbf{s}=\bar{\mathbf{s}}} = \begin{pmatrix} -\mathbf{R}(\theta_{ij})^T \mathbf{R}(\theta_i)^T & \mathbf{R}(\theta_{ij})^T \frac{\partial \mathbf{R}(\theta_i)^T}{\partial \theta_i} (\mathbf{t}_j - \mathbf{t}_i) \\ \mathbf{0}^T & -1 \end{pmatrix}, \quad (37a)$$

$$\mathbf{B}_{ij} = \left. \frac{\partial \mathbf{r}_{ij}}{\partial \mathbf{v}_j} \right|_{\mathbf{s}=\bar{\mathbf{s}}} = \begin{pmatrix} \mathbf{R}(\theta_{ij})^T \mathbf{R}(\theta_i)^T & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (37b)$$

where $\partial \mathbf{R}(\theta_i)^T / \partial \theta_i$ in \mathbf{A}_{ij} is given by

$$\frac{\partial \mathbf{R}(\theta)^T}{\partial \theta} = \begin{pmatrix} -\sin \theta & -\cos \theta \\ \cos \theta & -\sin \theta \end{pmatrix}. \quad (38)$$

The derivatives for the second error function $\mathbf{r}_{ik}(\mathbf{v}_i, \phi)$ are defined as

$$\mathbf{C}_{ik} = \left. \frac{\partial \mathbf{r}_{ik}}{\partial \phi} \right|_{\mathbf{s}=\bar{\mathbf{s}}} = \frac{\partial \mathbf{R}(\phi)^T}{\partial \phi} \mathbf{t}_i, \quad (39a)$$

$$\mathbf{D}_{ik} = \left. \frac{\partial \mathbf{r}_{ik}}{\partial \mathbf{v}_i} \right|_{\mathbf{s}=\bar{\mathbf{s}}} = \begin{pmatrix} \mathbf{R}(\phi) & \mathbf{0} \end{pmatrix}. \quad (39b)$$

The partial derivate of the rotation matrix w.r.t. the rotation angle in \mathbf{C}_{ik} takes the same form as presented in Equation (38). Note that we have not taken any derivatives w.r.t. the GPS coordinates, as we do not wish to reconfigure them, and thus they do not appear in the state vector. For clarity, we specify the dimensions of these matrices: both \mathbf{A} and \mathbf{B} are 3×3 matrices, \mathbf{C} is a 2×1 matrix, and \mathbf{D} is a 2×3 matrix. These derivatives lead to sparse Jacobian matrices for each of the error functions,

$$\mathbf{J}_{ij} = \begin{pmatrix} \dots & \dots & \mathbf{A}_{ij} & \mathbf{B}_{ij} & \dots \end{pmatrix}_{3 \times 1+3n}, \quad (40a)$$

$$\mathbf{J}_{ik} = \begin{pmatrix} \mathbf{C}_{ik} & \dots & \mathbf{D}_{ik} & \dots & \dots \end{pmatrix}_{2 \times 1+3n}. \quad (40b)$$

We use ellipses to indicate that unspecified values in the matrix are zeros. Since the odometry part of the pose graph only has constraints between consecutive nodes, the Jacobian \mathbf{J}_{ij} will always have a contiguous 3×6 block of non-zero values corresponding to the odometry constraints between nodes i and j . Furthermore, the Jacobian \mathbf{J}_{ik} will always have non-zero values in the first 2×1 block corresponding to the global orientation parameter and a 2×3 non-zero block at the i th node representing the constraint between the GPS-odometry correspondences. Now that the Jacobians are specified, we obtain the Gauss–Newton approximation to the Hessian matrices by

$$\mathbf{H}_{ij} = \mathbf{J}_{ij}^T \mathbf{Q}_{ij} \mathbf{J}_{ij}, \quad (41a)$$

$$\mathbf{H}_{ik} = \mathbf{J}_{ik}^T \mathbf{Q}_k \mathbf{J}_{ik}, \quad (41b)$$

and the gradient vectors by

$$\mathbf{b}_{ij} = \mathbf{J}_{ij}^T \mathbf{Q}_{ij} \mathbf{r}_{ij}, \quad (42a)$$

$$\mathbf{b}_{ik} = \mathbf{J}_{ik}^T \mathbf{Q}_k \mathbf{r}_{ik}. \quad (42b)$$

From an implementation standpoint, it is easier to construct the Hessian directly using our definitions for the non-zeros blocks in the Jacobians. The Hessian matrix for the i th pose in graph, assuming there is a corresponding GPS coordinate with the node, takes the form

$$\mathbf{H}_{ij} + \mathbf{H}_{ik} = \begin{pmatrix} \mathbf{C}_{ik}^T \mathbf{Q}_k \mathbf{C}_{ik} & \dots & \mathbf{C}_{ik}^T \mathbf{Q}_k \mathbf{D}_{ik} & \dots & \dots \\ \vdots & \ddots & \vdots & & \\ \mathbf{D}_{ik}^T \mathbf{Q}_k \mathbf{C}_{ik} & \dots & \mathbf{A}_{ij}^T \mathbf{Q}_{ij} \mathbf{A}_{ij} + \mathbf{D}_{ik}^T \mathbf{Q}_k \mathbf{D}_{ik} & \mathbf{A}_{ij}^T \mathbf{Q}_{ij} \mathbf{B}_{ij} & \dots \\ \vdots & & \mathbf{B}^T \mathbf{Q}_{ij} \mathbf{A}_{ij} & \mathbf{B}_{ij}^T \mathbf{Q}_{ij} \mathbf{B}_{ij} & \\ & & \vdots & & \ddots \end{pmatrix}. \quad (43)$$

Since the Jacobians are sparse, the resulting Hessian matrix is also sparse. Therefore, in practice, it is advantageous to use a memory-efficient sparse storage scheme for these matrices, e.g., compressed sparse column or row matrices.

The gradient vector for the i th pose can be constructed directly with

$$\mathbf{b}_{ij} + \mathbf{b}_{ik} = \begin{pmatrix} \mathbf{C}_{ik}^T \mathbf{Q}_k \mathbf{r}_{ik} \\ \vdots \\ \mathbf{A}_{ij}^T \mathbf{Q}_{ij} \mathbf{r}_{ij} + \mathbf{D}_{ik}^T \mathbf{Q}_k \mathbf{r}_{ik} \\ \mathbf{B}_{ij}^T \mathbf{Q}_{ij} \mathbf{r}_{ij} \\ \vdots \end{pmatrix}. \quad (44)$$

The final Hessian matrix and gradient vector for the linear system are obtained by summing over all the constraint-wise Hessians and descent vectors,

$$\mathbf{H} = \sum_{ij} \mathbf{H}_{ij} + \sum_{(i,k) \in \mathcal{C}} \mathbf{H}_{ik}, \quad (45a)$$

$$\mathbf{b} = \sum_{ij} \mathbf{b}_{ij} + \sum_{(i,k) \in \mathcal{C}} \mathbf{b}_{ik}. \quad (45b)$$

This linearization is performed during each iteration of the LM algorithm. We take advantage of the sparse structure of the Hessian and solve the system using sparse Cholesky factorization. We terminate the algorithm when the Euclidean norm of the linear increment to the state vector $\|\Delta \mathbf{s}\|$ is less than some small threshold, e.g., 0.001, and take the optimal configuration as $\mathbf{s}^* = \bar{\mathbf{s}} + \Delta \mathbf{s}$. Finally, we rotate each pose in the optimal state vector according to the global orientation parameter and translate back to the UTM coordinate system and update the rotation component of each pose,

$$\hat{\mathbf{v}}_i = \begin{pmatrix} \mathbf{R}(\phi^*) \mathbf{t}_i^* + \mathbf{g}_0 \\ \theta_i^* + \phi^* \end{pmatrix}, \quad \forall \{i\}_1^n. \quad (46)$$

Hereinafter, we omit the global orientation parameter and denote the globally aligned state vector as

$$\hat{\mathbf{s}} = \left(\hat{\mathbf{v}}_1^T, \hat{\mathbf{v}}_2^T, \dots, \hat{\mathbf{v}}_n^T \right)^T. \quad (47)$$

In order to resolve the global orientation parameter, it is required that we have a minimum of two GPS observations. We also add robustness to non-Gaussian distributed GPS errors using the same approach as in Section 2.2.2. We compute weights for the residuals corresponding to the global position coordinates using Tukey's biweight cost function, and the weights are used to scale the information matrix \mathbf{Q}_k .

We conclude this section by providing a note on the values in the information matrices. In general, the information matrix for the odometry observations consists of large values relative to the values in the information matrix for the global position observations. This is consistent with the fact that global position measurements are typically degraded under the canopy of a forest. Furthermore, VO is expected to perform well in feature-rich environments, such as a forest. We also note that we provide a substantially larger value to the position in the odometry information matrix corresponding to the heading. This makes the translation component of the nodes in the pose graph more elastic in order to conform to the GPS track while keeping the heading stiff to help maintain an accurate reconstruction of tree stem observations, which we address in the next section.

3.2. Local Refinement

Given the globally aligned state vector $\hat{\mathbf{s}}$, we will now optimize for a refined state vector by taking into account observations of tree stems. There are three main steps in local refinement: (1) We transform each tree stem observation to the world coordinate frame according to the globally aligned state vector we optimized in the previous section. (2) We associate the observations corresponding to an individual tree stem position in the

global map. (3) We optimize for a new configuration of the state that minimizes both the error functions from the previous section and an additional error function representing the discrepancy between observations of tree stems and their associated global position.

3.2.1. Graph Augmentation

Tree stems are detected in each frame using the convolutional neural network (CNN) object detector outlined in [59]. The input image to the network is resized to a resolution of 128 columns and 352 rows for real-time performance. We also detect the ground plane and breast height using the RANSAC-based algorithm presented in [60]. For each bounding box predicted by the CNN object detector, we extract the image coordinate, where the center-line of the bounding box along the u -axis of the image intersects the ground plane positioned at breast height. A disparity value is assigned to this image coordinate using Equation (5) in [59]. Using the inverse projection matrix and the disparity assignment, we project the image coordinate to \mathbb{R}^3 . We represent a tree stem observation in the camera coordinate frame of the i th camera pose as $\mathbf{z}_{iq} = (x, y)^T$, where x and y correspond to the back-projected image coordinate along the x and z axes of the camera coordinate frame. The index $q \in \mathcal{Z}_i$ where \mathcal{Z}_i is a set of indices denoting the observations of stems from the i th camera. Thus, $|\mathcal{Z}_i|$ is the number of tree stem observations in camera i and $|\mathcal{Z}|$ is the number of camera poses in the graph. We use the notation $(i, q) \in \mathcal{Z}$ to index the q^{th} observation in camera i .

Recalling that a camera pose in the globally aligned state vector is represented as $\hat{\mathbf{v}}_i = (\hat{x}, \hat{y}, \hat{\theta})^T$, we can transform the observations from the camera coordinate frame to the world frame with

$$\hat{\mathbf{z}}_{iq} = \mathbf{R}(\hat{\theta}_i)\mathbf{z}_{iq} + \hat{\mathbf{t}}_i, \quad \forall (i, q) \in \mathcal{Z}. \quad (48)$$

Given all tree stem observations in the global coordinate frame, we perform data association by clustering spatially similar observations. We use the density-based spatial clustering of applications with the noise (DBSCAN) algorithm [61] to cluster the observations. DBSCAN takes two parameters and a distance function. For the distance function, we simply use the Euclidean distance. The two parameters correspond to the search radius and the minimum number of points required for a cluster. We use a search radius of 1 m and 10 as the minimum number of points in a cluster. The algorithm yields a label for each observation that specifies to which cluster observation $\hat{\mathbf{z}}_{iq}$ belongs. A label equal to zero denotes an outlier, i.e., an observation that does not belong to any cluster. We denote the set of unique labels as \mathcal{L} and use the correspondence set \mathcal{M} to specify that observation $\hat{\mathbf{z}}_{iq}$ is assigned to label $\ell \in \mathcal{L}$. The notation $((i, q), \ell) \in \mathcal{M}$ is used to denote that observation $\hat{\mathbf{z}}_{iq}$ is assigned to cluster ℓ .

For each cluster of tree stem observations, we compute the center of the clusters by

$$\mathbf{m}_\ell = \frac{1}{\sum \mathbf{1}_\ell} \sum_{(i, q) \in \mathcal{Z}} \mathbf{1}_\ell \hat{\mathbf{z}}_{iq}, \quad \forall \ell \in \mathcal{L}, \quad (49)$$

where $\mathbf{m}_\ell = (x, y)^T$ is the center of the cluster in the global coordinate frame, and the notation $\mathbf{1}_\ell$ takes 1 when the observation $\hat{\mathbf{z}}_{iq}$ is assigned to cluster ℓ and zero otherwise. Given the cluster centers we extend our state vector as

$$\hat{\mathbf{s}} = \left(\hat{\mathbf{v}}_1^T, \hat{\mathbf{v}}_2^T, \dots, \hat{\mathbf{v}}_n^T, \mathbf{m}_1^T, \mathbf{m}_2^T, \dots, \mathbf{m}_\ell^T \right). \quad (50)$$

Note that we omitted the global orientation parameter. We can reconcile this in the Hessian matrix and gradient vector by removing the first row and column in the Hessian and the first row in the gradient vector, and redefining \mathbf{D}_{ik} as $(\mathbf{I}_2, \mathbf{0})$. See Figure 4 for a graphical illustration of the augmented graph.

where the Hessian matrix is now a $(1 + 3n + 2|\mathcal{L}|) \times (1 + 3n + 2|\mathcal{L}|)$ matrix. Similarly, the entries to the gradient vector are

$$\mathbf{b}_{il} = \begin{pmatrix} \vdots \\ \mathbf{E}_{il}^T \mathbf{Q}_{iq} \mathbf{r}_{il} \\ \vdots \\ \mathbf{F}_{il}^T \mathbf{Q}_{iq} \mathbf{r}_{il} \\ \vdots \end{pmatrix} \quad (55)$$

The complete linear system is constructed by summing the Hessian matrices and gradient vectors for all the tree stem observations and adding them to the other Hessians and gradient vectors defined in Equation (45). Again, the first row and column of the Hessian, as well as the first row in the gradient vector, are removed to account for the omission of the global orientation parameter. We solve the linear system, $(\mathbf{H} + \lambda \mathbf{I})\Delta \mathbf{s} = -\mathbf{b}$, using sparse Cholesky factorization and update the state vector by $\hat{\mathbf{s}} \leftarrow \Delta \mathbf{s}$. As we did for global alignment, we terminate the algorithm when $\Delta \mathbf{s}$ is less than some predetermined convergence threshold.

4. Analysis and Discussion

To test VO and the localization and mapping algorithms outlined above, we acquired a video sequence of a 1115 m path through a sparse ponderosa pine (*Pinus ponderosa* Douglas ex Lawson) forest in Western Montana. The video was captured by walking a hand-held 12 cm baseline ZED stereo camera [62] through the forest. The camera was operated at 10 frames per second and VGA resolution (480×640). Mounted on top of the camera field monitor was an antenna connected to a GlobalTop FGPMOPA6H GPS module [63] that was queried for a GPS coordinate reading and a PDOP value after each video frame capture. The GPS coordinates and video frames were stored on an embedded backpack computer. Figure 5a shows the GPS coordinate readings projected on the UTM coordinate system after translating the position track by subtracting the first GPS coordinate from all coordinates.

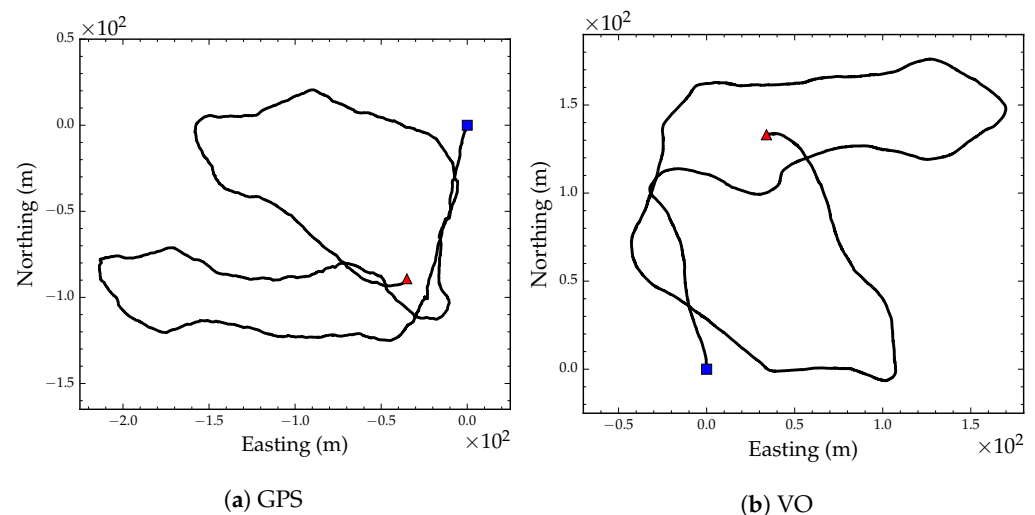


Figure 5. GPS position track projected on a UTM coordinate system (a). VO position track (b). The blue square denotes the starting position and the red triangle denotes the end position.

We estimated camera egomotion using the direct VO algorithm presented in Section 2. We used a camera resolution of 240×320 to estimate egomotion and a 25% gradient magnitude threshold during pixel selection. Frame-to-frame egomotion parameters were composed to construct an odometry track using Equation (28). Figure 5b shows the visual odometry position track after converting the translation components of the egomotion parameters to meters. Figure 6 shows the GPS track (black line) and the optimized VO track

after global alignment and local refinement (red line). The blue dashed line in Figure 6 shows the VO track after only applying the optimized global orientation parameter. As the figure suggests, the VO position track, although locally consistent, is subject to drift after approximately 300 m.

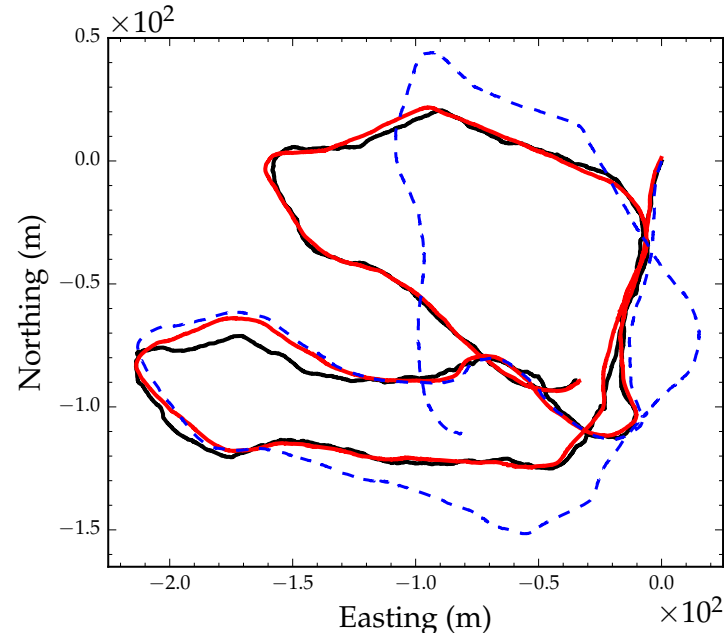


Figure 6. GPS position track denoted by the black solid line. Optimized position track denoted by the red solid line. VO track rotated by the global orientation parameter denoted by the blue dashed line.

4.1. Localization

In this section, we demonstrate the performance of GPS and VO integrated localization under various scenarios of degraded and intermittent GPS reception. We refer to the optimized position track shown in Figure 6 (red line) as the ground truth position track. We acknowledge that this track has not actually been ground truthed with survey grade equipment; however, this track is deemed optimal, given the available data.

4.1.1. Degraded GPS Reception

We simulated degraded GPS reception by adding zero mean Gaussian noise with a standard deviation of 5 m to each observed GPS coordinate. The incremental update to the state vector and the current estimate of the state converged after 15 LM iterations. Figure 7 shows six snapshots during optimization. The global orientation parameter converged after the 6th iteration. Figure 8 shows the converged path after global alignment and local refinement with a root mean squared error (RMSE) of less than 0.1 m compared to the ground truth position track. This suggests that, as long as GPS errors are Gaussian distributed, we can expect accurate position tracking.

We also tested robustness to non-Gaussian distributed noise in GPS readings. We randomly selected 5% of the coordinates from the GPS track and added uniformly distributed noise with a range of 0 to 200 m. Figure 9 shows the converged path after global alignment and local refinement with non-Gaussian distributed noise in GPS coordinates plotted over the ground truth path (RMSE < 0.1 m). According to this result, the algorithm is insensitive to GPS coordinate outliers.

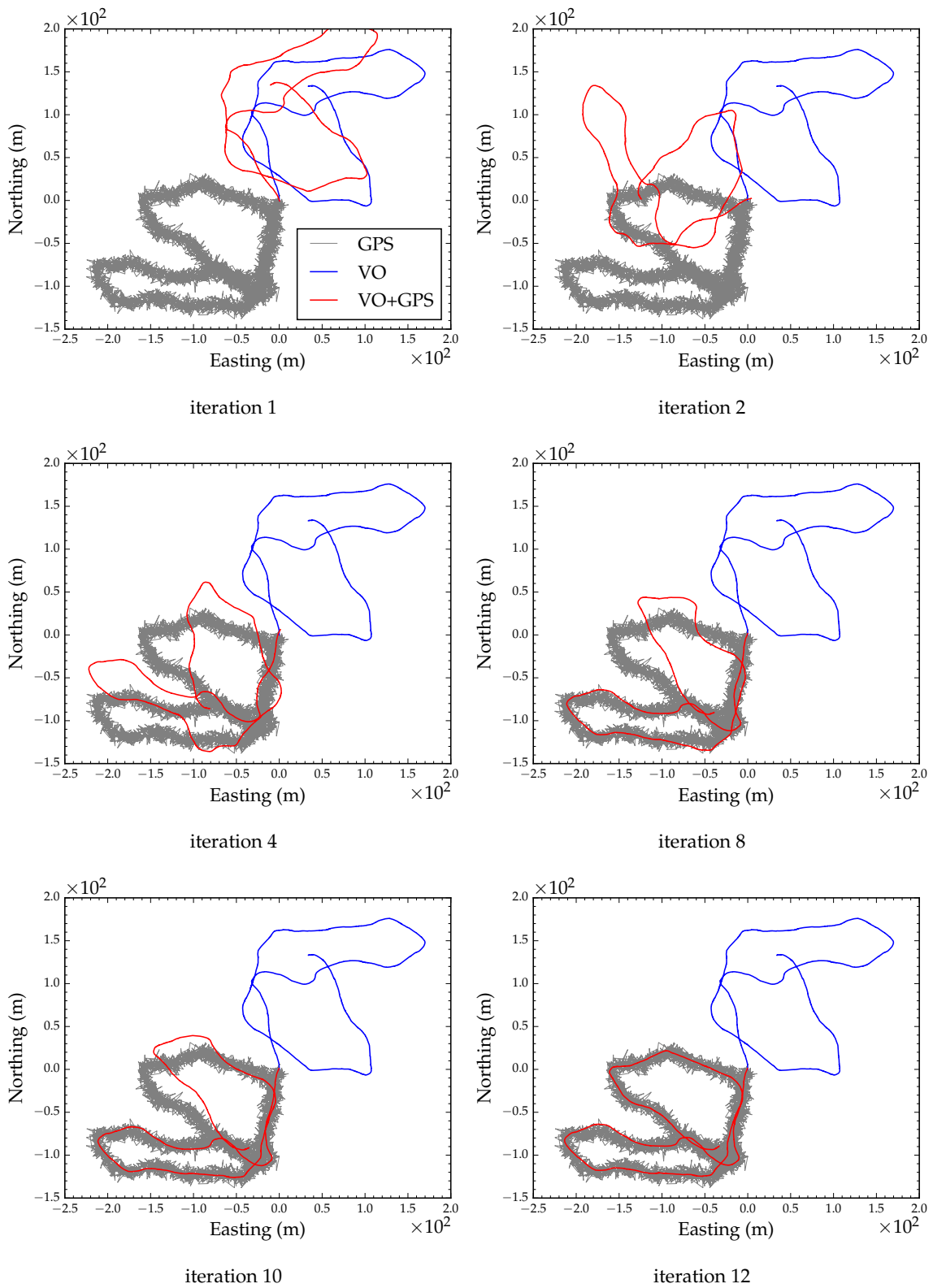


Figure 7. Global alignment with degraded GPS reception. Each subfigure shows a snapshot during optimization.

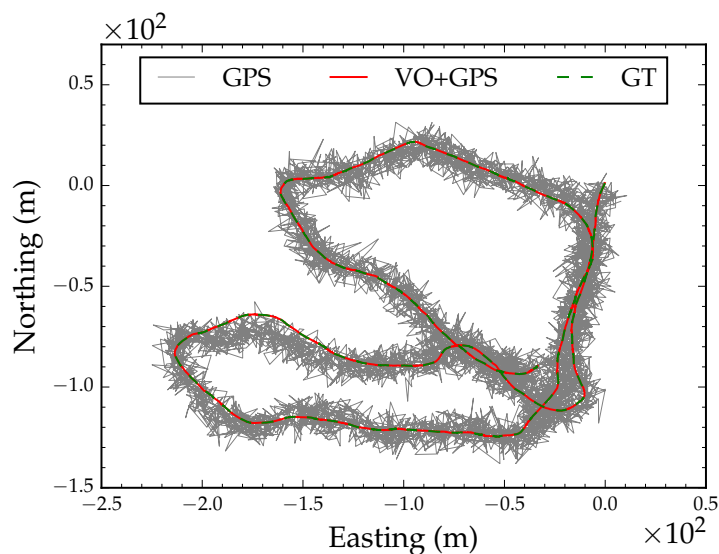


Figure 8. Converged globally aligned path with degraded GPS reception.

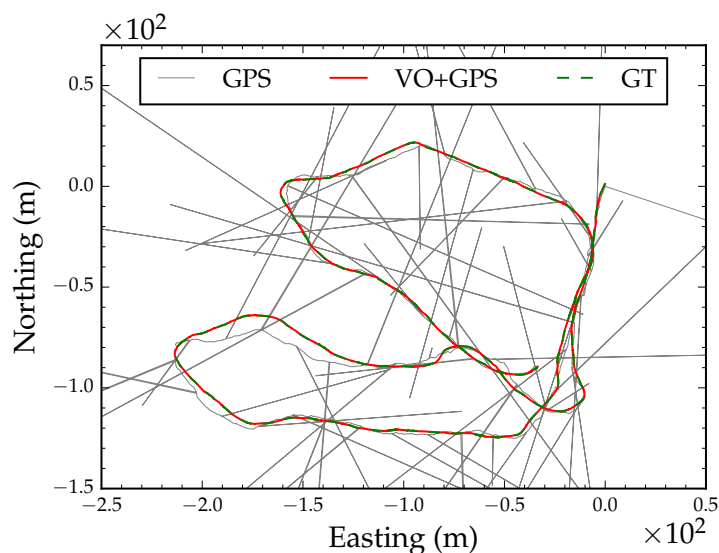


Figure 9. Converged globally aligned path with non-Gaussian GPS errors.

4.1.2. Intermittent GPS Reception

When GPS is used under an extremely dense forest canopy, reception might only be intermittently reliable when the receiver is stationary for long periods of time or when the receiver crosses openings in the canopy. We simulated this scenario by extracting 12 coordinates from the GPS track; we extracted the starting position, ending position and randomly selected 10 coordinates along 100 m intervals from the track. Figure 10 shows six snapshots during optimization with 12 intermittent GPS coordinate readings. The algorithm converged after 21 iterations. Following local refinement, the optimized path had a RMSE of 2.7 m compared to the ground truth position track in Figure 11. This is an important application of the proposed algorithm in situations where GPS reception is only available intermittently.

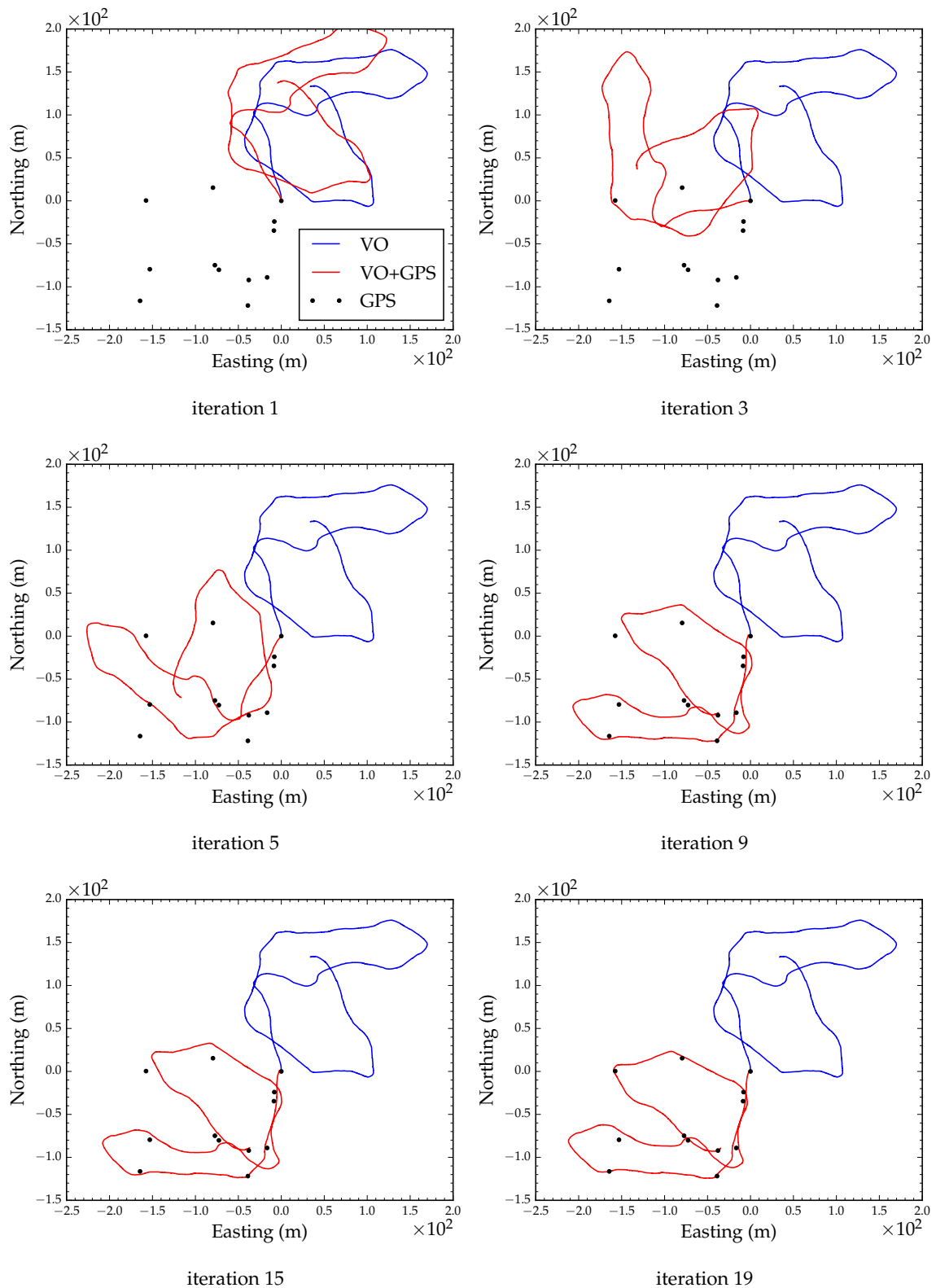


Figure 10. Global alignment with intermittent GPS reception. Each subfigure is a snapshot during optimization.

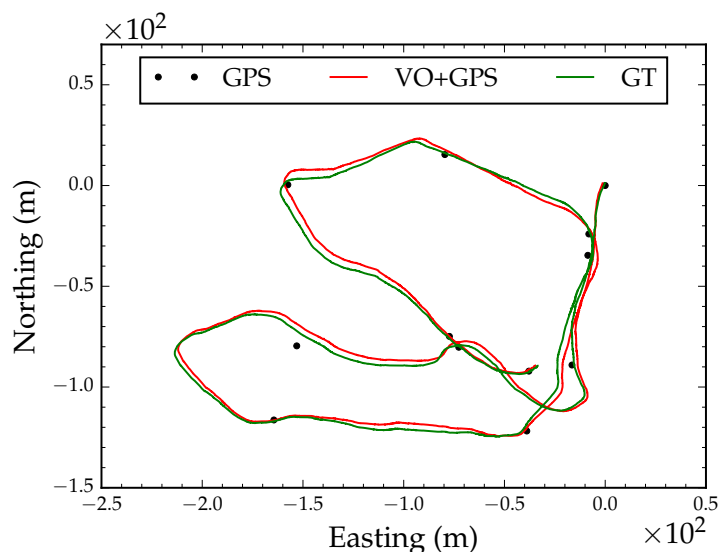


Figure 11. Converged globally aligned path with intermittent GPS reception.

4.2. Mapping

In order to generate a stem map, we performed global alignment using all the coordinates from the GPS track and the VO path shown in Figure 5a,b. Figure 12 shows the result after clustering the observed tree stem position. The black dots in the figure denote the cluster centers, and the ellipses around each center show the covariance matrix of the clusters at four standard deviations. There was a total of 6205 tree stem observations and 140 clusters, i.e., individual tree stems. Figure 13 shows the optimized cluster centers, i.e., tree stem positions, and the covariance structure of the observations at four standard deviations. Local refinement converged after nine iterations. Note that the ellipses representing the covariances of the clusters are smaller after local refinement. This is a result of simultaneously optimizing the position track and the cluster centers.

To test the accuracy of the stem map generated from the local refinement step, we collected a ground truth stem map of the 12 acre forest, from which we acquired the video and GPS track. We obtained global coordinates for each individual tree in the stand using a TruePulse 360B laser range finder and a 13-bit BEI industrial rotary encoder mounted on a tripod. We installed a reflector target near the center of the stand, at which a GPS coordinate was acquired using a mapping grade receiver. We mapped subsections of the stand by first aligning the laser and encoder to north, using the appropriate declination for the area, then recorded the distance and angle to the target to globally localize the plot center. We recorded the distance and angle to each individual tree within view from the plot center. We repeated this process moving clockwise around the reflector target mapping small subsections of the stand until the entire stand was mapped.

Figure 14 shows the ground truth stem map (green triangles), the predicted tree stem position from the camera (red circles), and the optimized position track (yellow line) superimposed on an aerial photograph of the stand. We manually associated each observed tree stem position from the camera with its corresponding ground truth tree stem. Among the 140 predicted tree stem positions, we classified 2 observations as false positives, i.e., predicted tree stems that correspond to a tree in the aerial photograph that were not recorded during the collection of the ground truth stem map, and 7 tree stems as duplicate observations, i.e., observed tree stems that belong to the same ground truth stem. For all predicted and ground truth stem correspondences ($n = 140$), we calculated a RMSE of 2.16 m. This is a 46% improvement over the best-case expected accuracy of a consumer-grade GPS device under forest canopy (4 m).

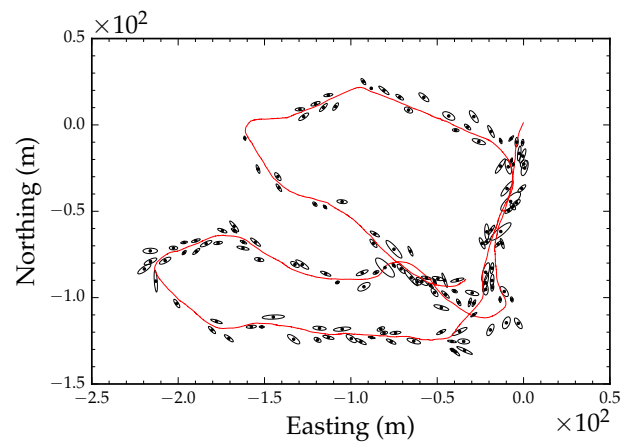


Figure 12. Tree stem position averages (black dots), tree stem position covariances (black ellipses), and pose track (red line) before local refinement

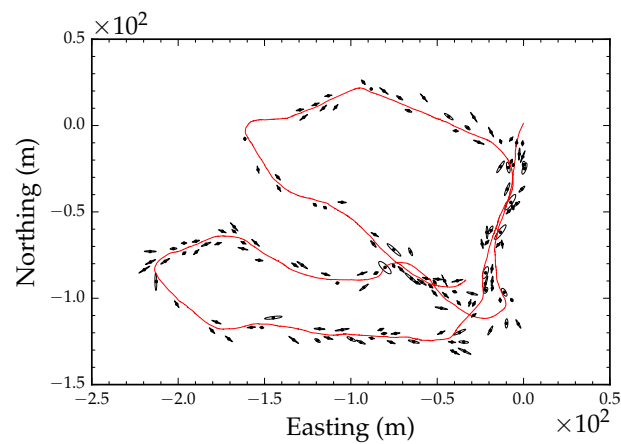


Figure 13. Tree stem position averages (black dots), tree stem position covariances (black ellipses), and pose track (red line) after local refinement.

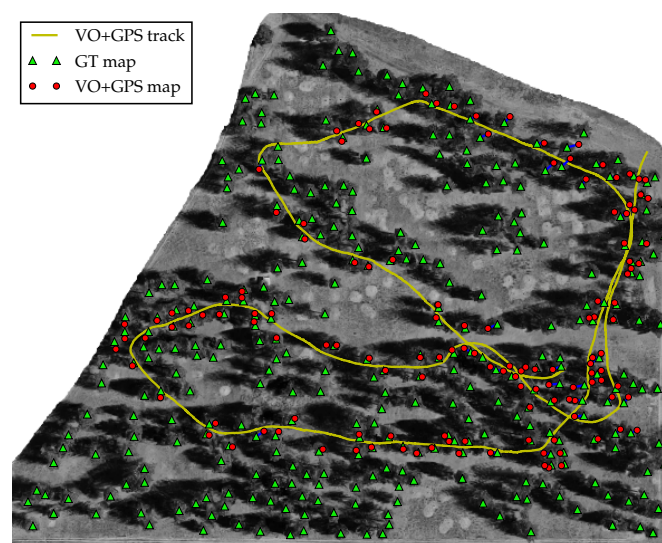


Figure 14. Ground truth tree stem positions and estimated tree stems positions superimposed on an aerial photograph of the 12 ac forest stand. Green triangles denote ground truth stem positions, red circles show the estimated stem position from the camera, the yellow line denotes the optimized position track, and the blue lines show association between observed and estimated tree positions.

5. Conclusions

In this paper, we presented a real-time algorithm for VO estimation and a novel method for integrating VO and a vision-based tree stem detection system with GNSS positioning to generate accurate stem maps. Our approach is based on existing and widely applied optimization techniques, i.e., GN and LM non-linear least squares, which provide efficient solution procedures to the optimization problems formulated in this paper.

Although GNSS positioning is used to maintain a globally consistent position track, the heading of the camera is subject to drift; global positioning cannot be used to infer the heading of the camera. This issue did not appear to be significant in our dataset; however, we expect that camera heading will eventually drift since it is unconstrained and only estimated from visual egomotion. This issue can be resolved by incorporating a global direction sensor, i.e., magnetometer, to maintain a globally consistent heading. The inclusion of directional data in the global alignment optimization step requires minimal modifications to the presented algorithm. Since electronic direction sensors are relatively inexpensive, we recommend using such a sensor for large-scale mapping applications.

The successful integration of real-time localization and mapping into forestry practices presents new opportunities for enhancing decision-making processes and implementing complex silvicultural prescriptions. This research provides a practical, low-cost solution that addresses the need for accurate and efficient positioning and mapping in forested environments.

6. Patents

US Patent No. US011481972B2: Method of performing dendrometry and forest mapping.

Author Contributions: Conceptualization, L.A.W. and W.C.; methodology, L.A.W. and W.C.; software, L.A.W.; validation, L.A.W. and W.C.; formal analysis, L.A.W.; investigation, L.A.W. and W.C.; resources, L.A.W.; data curation, L.A.W.; writing—original draft preparation, L.A.W.; writing—review and editing, L.A.W. and W.C.; visualization, L.A.W.; supervision, W.C.; project administration, W.C.; funding acquisition, W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the U.S. Forest Service National Technology and Development Program under contract number 16CS-1113-8100-017.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We gratefully acknowledge the invaluable assistance of Jenny Perth from the USDA Forest Service's National Technology Development Program, whose diligent efforts in collecting validation data significantly contributed to the success of this research.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this appendix, we provide an overview of the mathematical foundations for special orthogonal and special Euclidean Lie groups. We discuss the closed-form exponential maps and logarithm maps for these groups, as well as their corresponding Lie algebras. This section aims to offer a convenient reference for the reader, summarizing essential concepts and formulas related to these topics, even though these derivations might exist in textbooks.

The special orthogonal Lie groups $SO(2)$ and $SO(3)$ represent rotations in 2 and 3 dimensions, respectively. In $SO(2)$, we represent a transformation by a 2×2 matrix,

$\mathbf{R} \in \text{SO}(2)$, and its Lie algebra by $\theta \in \mathfrak{so}(2)$. The closed-form exponential map that converts the Lie algebra, $\mathfrak{so}(2)$, to the Lie group, $\text{SO}(2)$, is given by

$$\mathbf{R} = \exp(\theta_{\times}) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \in \text{SO}(2), \quad (\text{A1})$$

where θ_{\times} denotes a skew symmetric matrix generated from θ . The inverse of the exponential map is given by the logarithm map that brings the transformation matrix back to the algebra,

$$\theta = \ln(\mathbf{R}) = \tan^{-1} \frac{r_{21}}{r_{11}} \in \mathfrak{so}(2), \quad (\text{A2})$$

where r_{21} and r_{11} are entries in \mathbf{R} . In practice the \tan^{-1} function is replaced by the two-argument inverse tangent function $\arctan2(y, x)$ for appropriate quadrant checking.

In the case of 3-dimensional rotations, we represent the transformation as a 3×3 matrix $\mathbf{R} \in \text{SO}(3)$, and its algebra as $\omega \in \mathfrak{so}(3)$, where $\omega = (\alpha, \beta, \gamma)^{\top}$ is a vector encoding the rotation angles about the x, y and z -axes. The closed-form exponential is given by

$$\mathbf{R} = \exp(\omega_{\times}) = \mathbf{I} + \left(\frac{\sin \vartheta}{\vartheta}\right) \omega_{\times} + \left(\frac{1 - \cos \vartheta}{\vartheta^2}\right) \omega_{\times}^2 \in \text{SO}(3), \quad (\text{A3})$$

where $\vartheta = \sqrt{\omega^{\top} \omega}$ and ω_{\times} is defined as

$$\omega_{\times} = \begin{pmatrix} 0 & -\gamma & \beta \\ \gamma & 0 & -\alpha \\ -\beta & \alpha & 0 \end{pmatrix}. \quad (\text{A4})$$

Equation (A3) is Rodrigues' formula for rotating a vector in space, given an axis and a rotation angle. The logarithm map that brings the transformation matrix back to the algebra is given by

$$\vartheta = \cos^{-1} \frac{\text{Tr}(\mathbf{R}) - 1}{2}, \quad (\text{A5a})$$

$$\omega_{\times} = \ln(\mathbf{R}) = \frac{\vartheta}{2 \sin \vartheta} (\mathbf{R} - \mathbf{R}^{\top}) \in \mathfrak{so}(3). \quad (\text{A5b})$$

The algebra is taken from the off-diagonal components of ω_{\times} , i.e., skew symmetric matrix to vector. The notation $\text{Tr}(\cdot)$ is the trace function of a square matrix, which returns the sum of the elements along the diagonal.

In both the 2D and 3D cases, the column vectors of the matrices representing the transformations are orthogonal. Thus, the inverse is equivalent to the matrix transpose,

$$\mathbf{R}^{-1} = \mathbf{R}^{\top} \in \text{SO}(2), \quad (\text{A6a})$$

$$\mathbf{R}^{-1} = \mathbf{R}^{\top} \in \text{SO}(3). \quad (\text{A6b})$$

Rigid transformations in 2 and 3 dimensions are represented by the special Euclidean Lie groups $\text{SE}(2)$ and $\text{SE}(3)$. By definition, these groups include rotations from the special orthogonal group described in the previous section,

$$\text{SE}(2) \stackrel{\text{def}}{=} \text{SO}(2) \times \mathbb{R}^2, \quad (\text{A7a})$$

$$\text{SE}(3) \stackrel{\text{def}}{=} \text{SO}(3) \times \mathbb{R}^3. \quad (\text{A7b})$$

A rigid transformation in 2D takes the form

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} \in \text{SO}(2) & \mathbf{t} \in \mathbb{R}^2 \\ \mathbf{0}^{\top} & 1 \end{pmatrix} \in \text{SE}(2). \quad (\text{A8})$$

This matrix is homogeneous for compatibility with inversion and composition operations. We represent the parameter space, i.e., the Lie algebra, as $\xi = (\theta, x, y)^T = (\theta, \mathbf{u})^T \in \mathfrak{se}(2)$. The exponential map is given in closed form as

$$\mathbf{T} = \exp(\hat{\xi}) = \begin{pmatrix} \exp(\theta_{\times}) & \mathbf{V}\mathbf{u} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \text{SE}(2) \quad (\text{A9})$$

where $\exp(\theta_{\times})$ is defined in Equation (A1) and

$$\mathbf{V} = \frac{1}{\theta} \begin{pmatrix} \sin \theta & -(1 - \cos \theta) \\ 1 - \cos \theta & \sin \theta \end{pmatrix}. \quad (\text{A10})$$

The logarithm map that brings the transformation matrix back to the algebra is given by

$$\xi = \begin{pmatrix} \theta \\ \mathbf{V}^{-1}\mathbf{t} \end{pmatrix} \in \mathfrak{se}(2), \quad (\text{A11})$$

where $\theta = \ln(\mathbf{R})$ is defined in Equation (A2) and \mathbf{V}^{-1} is defined as

$$\vartheta = \frac{\sin \theta}{\theta}, \quad (\text{A12a})$$

$$\phi = \frac{1 - \cos \theta}{\theta}, \quad (\text{A12b})$$

$$\mathbf{V}^{-1} = \frac{1}{\vartheta^2 + \phi^2} \begin{pmatrix} \vartheta & \phi \\ -\phi & \vartheta \end{pmatrix}. \quad (\text{A12c})$$

Rigid transformations in 3-dimensional space are written similarly to our definition for $\text{SE}(2)$ but with 3D rotation matrices and 3-vector translations,

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} \in \text{SO}(3) & \mathbf{t} \in \mathbb{R}^3 \\ \mathbf{0}^T & 1 \end{pmatrix} \in \text{SE}(3). \quad (\text{A13})$$

We use the variable ξ to represent the algebra, $\xi = (\omega, \mathbf{u})^T \in \mathfrak{se}(3)$, where $\omega = (\alpha, \beta, \gamma)^T$ and $\mathbf{u} = (x, y, z)^T$. The exponential map again can be computed in closed form,

$$\exp(\hat{\xi}) = \exp \begin{pmatrix} \omega_{\times} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{V}\mathbf{u} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \text{SE}(3), \quad (\text{A14})$$

where \mathbf{R} , defined in Equation (A3) is restated below for convenience, and \mathbf{V} are given by

$$\theta = \sqrt{\omega^T \omega}, \quad (\text{A15a})$$

$$\vartheta = \frac{\sin \theta}{\theta}, \quad (\text{A15b})$$

$$\phi = \frac{1 - \cos \theta}{\theta^2}, \quad (\text{A15c})$$

$$\psi = \frac{1 - \vartheta}{\theta^2}, \quad (\text{A15d})$$

$$\mathbf{R} = \mathbf{I} + \vartheta \omega_{\times} + \phi \omega_{\times}^2, \quad (\text{A15e})$$

$$\mathbf{V} = \mathbf{I} + \phi \omega_{\times} + \psi \omega_{\times}^2. \quad (\text{A15f})$$

Finally, the logarithm map is written as

$$\xi = \ln(\mathbf{T}) = \begin{pmatrix} \ln(\mathbf{R}) \\ \mathbf{V}^{-1}\mathbf{t} \end{pmatrix} \in \mathfrak{se}(3). \quad (\text{A16})$$

The logarithm map of the rotation part is given by Equation (A5), and the inverse of \mathbf{V} is defined as

$$\mathbf{V}^{-1} = \mathbf{I} + \frac{1}{2}\boldsymbol{\omega}_{\times} + \frac{1}{\theta^2}\left(1 - \frac{\theta}{2\phi}\right)\boldsymbol{\omega}_{\times}^2. \quad (\text{A17})$$

Composition of two transformation is equivalent for SE(2) and SE(3),

$$\mathbf{T}_1 \circ \mathbf{T}_2 = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 & \mathbf{R}_1\mathbf{t}_2 + \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (\text{A18})$$

and inverting a transformation is given by

$$\mathbf{T}^{-1} = \begin{pmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (\text{A19})$$

As a result, composition and inversion in a single operation is written as

$$\mathbf{T}_1 \circ \mathbf{T}_2^{-1} = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2^T & \mathbf{R}_1(-\mathbf{R}_2^T\mathbf{t}_2) + \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (\text{A20})$$

References

- Mizunaga, H.; Nagaike, T.; Yoshida, T.; Valkonen, S. Feasibility of silviculture for complex stand structures: Designing stand structures for sustainability and multiple objectives. *J. For. Res.* **2010**, *15*, 1–2. [\[CrossRef\]](#)
- Hartley, D.; Han, H. Effects of Alternative Silvicultural Treatments on Cable Harvesting Productivity and Cost in Western Washington. *West. J. Appl. For.* **2007**, *22*, 204–212. [\[CrossRef\]](#)
- Holopainen, M.; Vastaranta, M.; Hyypä, J. Outlook for the Next Generation's Precision Forestry in Finland. *Forests* **2014**, *5*, 1682–1694. [\[CrossRef\]](#)
- Grift, T.; Zhang, Q.; Kondo, N.; Ting, K.C. Review of Automation and Robotics for the Bioindustry. *J. Biomechatron. Eng.* **2008**, *1*, 37–54.
- Evans, D.; Carraway, R.; Simmons, G. Use of global positioning system (GPS) for forest plot location. *South. J. Appl. For.* **1992**, *16*, 67–70. [\[CrossRef\]](#)
- Liu, C.J.; Brantigan, R. Using differential GPS for forest traverse surveys. *Can. J. For. Res.* **2011**, *25*, 1795–1805. [\[CrossRef\]](#)
- McDonald, T.P.; Carter, E.A.; Taylor, S.E. Using the global positioning system to map disturbance patterns of forest harvesting machinery. *Can. J. For. Res.* **2002**, *32*, 310–319. [\[CrossRef\]](#)
- Veal, M.W.; Taylor, S.E.; McDonald, T.P.; McLemore, D.K.; Dunn, M.R. Accuracy of Tracking Forest Machines with GPS. *Trans. ASAE* **2001**, *44*, 1903–1911.
- Devlin, G.J.; McDonnell, K. Performance Accuracy of Real-Time GPS Asset Tracking Systems for Timber Haulage Trucks Travelling on Both Internal Forest Road and Public Road Networks. *Int. J. For. Eng.* **2009**, *20*, 45–49. [\[CrossRef\]](#)
- Becker, R.M.; Keefe, R.F.; Anderson, N.M. Use of real-time GNSS-RF data to characterize the swing movements of forestry equipment. *Forests* **2017**, *8*, 44. [\[CrossRef\]](#)
- McDonald, T.P.; Fulton, J.P. Automated time study of skidders using global positioning system data. *Comput. Electron. Agric.* **2005**, *48*, 19–37. [\[CrossRef\]](#)
- Gallo, R.; Grigolato, S.; Cavalli, R.; Mazzetto, F. GNSS-based operational monitoring devices for forest logging operation chains. *J. Agric. Eng.* **2013**, *44*, 140–144. [\[CrossRef\]](#)
- Grayson, L.M.; Keefe, R.F.; Tinkham, W.T.; Eitel, J.U.H.; Saralecos, J.D.; Smith, A.M.S.; Zimbelman, E.G. Accuracy of WAAS-Enabled GPS-RF Warning Signals When Crossing a Terrestrial Geofence. *Sensors* **2016**, *16*, 912. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zimbelman, E.G.; Keefe, R.F.; Strand, E.K.; Kolden, C.A.; Wempe, A.M. Hazards in Motion: Development of Mobile Geofences for Use in Logging Safety. *Sensors* **2017**, *17*, 822. [\[CrossRef\]](#)
- Zimbelman, E.G.; Keefe, R.F. Real-time positioning in logging: Effects of forest stand characteristics, topography, and line-of-sight obstructions on GNSS-RF transponder accuracy and radio signal propagation. *PLoS ONE* **2018**, *13*, e0191017. [\[CrossRef\]](#) [\[PubMed\]](#)
- Danskin, S.; Bettinger, P.; Jordan, T. Multipath Mitigation under Forest Canopies: A Choke Ring Antenna Solution. *For. Sci.* **2009**, *55*, 109–116. [\[CrossRef\]](#)
- Wing, M.G.; Eklund, A.; Kellogg, L.D. Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability. *J. For.* **2005**, *103*, 169–173. [\[CrossRef\]](#)
- Wing, M.G. Consumer-Grade Global Positioning Systems (GPS) Receiver Performance. *J. For.* **2008**, *106*, 185–190. [\[CrossRef\]](#)

19. Andersen, H.E.; Clarkin, T.; Winterberger, K.; Strunk, J. An accuracy assessment of positions obtained using survey- and recreational-grade Global Positioning System receivers across a range of forest conditions within the Tanana Valley of interior Alaska. *West. J. Appl. For.* **2009**, *24*, 128–136. [[CrossRef](#)]
20. Bettinger, P.; Fei, S. One year's experience with a recreation-grade GPS receiver. *Int. J. Math. Comput. For. Nat. Resour. Sci.* **2010**, *2*, 153–160.
21. Wing, M.G. Consumer-grade GPS receiver measurement accuracy in varying forest conditions. *Res. J. For.* **2011**, *5*, 78–88. [[CrossRef](#)]
22. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; CVPR 2004; Volume 1, pp. 652–659. [[CrossRef](#)]
23. Longuet-Higgins, H.C. A computer algorithm for reconstructing a scene from two projections. *Nature* **1981**, *293*, 133. [[CrossRef](#)]
24. Harris, C.; Pike, J. 3D positional integration from image sequences. *Image Vis. Comput.* **1988**, *6*, 87–90. [[CrossRef](#)]
25. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry for ground vehicle applications. *J. Field Robot.* **2006**, *23*, 3–20. [[CrossRef](#)]
26. Howard, A. Real-time stereo visual odometry for autonomous ground vehicles. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008. [[CrossRef](#)]
27. Cumani, A. Feature localization refinement for improved visual odometry accuracy. *Int. J. Circuits Syst. Single Process.* **2011**, *5*, 151–158.
28. Jiang, Y.; Xu, Y.; Liu, Y. Performance evaluation of feature detection and matching in stereo visual odometry. *Neurocomputing* **2013**, *120*, 380–390. [[CrossRef](#)]
29. Rosten, E.; Drummond, T. Fusing points and lines for high performance tracking. In Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 17–21 October 2005; Volume 2, pp. 1508–1511. [[CrossRef](#)]
30. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Volume 1, pp. 430–443. [[CrossRef](#)]
31. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
32. Harris, C.; Stephens, M. A combined corner and edge detector. *Alvey Vis. Conf.* **1988**, *15*, 147–151.
33. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
34. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [[CrossRef](#)]
35. Fraundorfer, F.; Scaramuzza, D. Visual Odometry: Part II—Matching, Robustness, and Applications. *IEEE Robot. Autom. Mag.* **2012**, *19*, 78–90. [[CrossRef](#)]
36. Moreno-Noguer, F.; Lepetit, V.; Fua, P. Accurate Non-Iterative O(n) Solution to the PnP Problem. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8. [[CrossRef](#)]
37. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
38. Scaramuzza, D.; Fraundorfer, F. Visual Odometry: Part I—The first 30 years and fundamentals. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [[CrossRef](#)]
39. Comport, A.I.; Malis, E.; Rives, P. Accurate Quadrifocal Tracking for Robust 3D Visual Odometry. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 40–45. [[CrossRef](#)]
40. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22. [[CrossRef](#)]
41. Engel, J.; Sturm, J.; Cremers, D. Semi-Dense Visual Odometry for a Monocular Camera. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013.
42. Usenko, V.; Engel, J.; Stueckler, J.; Cremers, D. Direct Visual-Inertial Odometry with Stereo Cameras. In Proceedings of the International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
43. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [[CrossRef](#)]
44. Alismail, H.; Kaess, M.; Browning, B.; Lucey, S. Direct Visual Odometry in Low Light Using Binary Descriptors. *IEEE Robot. Autom. Lett.* **2017**, *2*, 444–451. [[CrossRef](#)]
45. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; IJCAI'81; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1981; Volume 2, pp. 674–679.
46. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2006.
47. Rossmann, J.; Schluse, M.; Schlette, C.; Buecken, A.; Emde, M. Realization of a highly accurate mobile robot system for multi purpose precision forestry applications. In Proceedings of the 2009 International Conference on Advanced Robotics, ICAR 2009, Munich, Germany, 22–26 June 2009; pp. 1–6.
48. Rossmann, J.; Krahwinkler, P.; Schlette, C. Navigation of Mobile Robots in Natural Environments: Using Sensor Fusion in Forestry. *Syst. Cybern. Inform.* **2010**, *8*, 67–71.
49. Grisetti, G.; Kummerle, R.; Stachniss, C.; Burgard, W. A Tutorial on Graph-Based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43. [[CrossRef](#)]

50. Baker, S.; Matthews, I. Lucas-Kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.* **2004**, *54*, 221–255. [[CrossRef](#)]
51. Hirschmuller, H. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 328–341. [[CrossRef](#)] [[PubMed](#)]
52. Hernandez-Juarez, D.; Chacón, A.; Espinosa, A.; Vázquez, D.; Moure, J.C.; López, A.M. Embedded Real-time Stereo Estimation via Semi-Global Matching on the GPU. In Proceedings of the International Conference on Computational Science 2016, ICCS 2016, San Diego, CA, USA, 6–8 June 2016; pp. 143–153. [[CrossRef](#)]
53. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
54. Beaton, A.E.; Tukey, J.W. The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics* **1974**, *16*, 147–185. [[CrossRef](#)]
55. Black, M.J.; Rangarajan, A. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vis.* **1996**, *19*, 57–91. [[CrossRef](#)]
56. Huber, P.J. Robust Estimation of a Location Parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [[CrossRef](#)]
57. Levenberg, K. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Q. Appl. Math.* **1944**, *2*, 164–168. [[CrossRef](#)]
58. Marquardt, D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM J. Appl. Math.* **1963**, *11*, 431–441. [[CrossRef](#)]
59. Wells, L.A.; Chung, W. Real-Time Computer Vision for Tree Stem Detection and Tracking. *Forests* **2023**, *14*, 267. [[CrossRef](#)]
60. Wells, L.A.; Chung, W. Evaluation of Ground Plane Detection for Estimating Breast Height in Stereo Images. *For. Sci.* **2020**, *66*, 612–622. [[CrossRef](#)]
61. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
62. StereoLabs. ZED Stereo Camera. 2016. Available online: <https://www.stereolabs.com/> (accessed on 12 September 2018).
63. GlobalTop Technology Inc. GlobalTop FGPMMOPA6H GPS Module. Acquired by Sierra Wireless. Available online: <https://www.gtop-tech.com/> (accessed on 19 September 2018).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.