Large-scale 3D Mapping of Subarctic Forests

Philippe Babin, Philippe Dandurand, Vladimír Kubelka, Philippe Giguère and
François Pomerleau*

Abstract The ability to map challenging subarctic environments opens new horizons for robotic deployments in industries such as forestry, surveillance, and open-pit mining. In this paper, we explore possibilities of large-scale lidar mapping in a boreal forest. Computational and sensory requirements with regards to contemporary hardware are considered as well. The lidar mapping is often based on the Simultaneous Localization and Mapping (SLAM) technique relying on pose graph optimization, which fuses the Iterative Closest Point (ICP) algorithm, Global Navigation Satellite System (GNSS) positioning, and Inertial Measurement Unit (IMU) measurements. To handle those sensors directly within the ICP minimization process, we propose an alternative approach of embedding external constraints. Furthermore, a novel formulation of a cost function is presented and cast into the problem of handling uncertainties from GNSS and lidar points. To test our approach, we acquired a large-scale dataset in the *Forêt Montmorency* research forest. We report on the technical problems faced during our winter deployments aiming at building 3D maps using our new cost function. Those maps demonstrate both global and local consistency over 4.1 km.

Key words: Lidar, cost function, ICP, penalties, mapping, GNSS, GPS, winter, forest, anisotropic

1 Introduction

Autonomous mobile robots require a representation (i.e., a map) of the environment in order to perform specific tasks. For instance, maps are needed internally to plan motions and avoid obstacles. The map itself can also be the objective, captured by

---
* The authors are from the Northern Robotics Laboratory, Université Laval, Canada.
philippe.babin.1@ulaval.ca

robots and used for later analysis [1], including forestry inventories [2]. Although many solutions exist for localization and mapping, the environment itself influences the complexity of the task, and thus dictates which algorithm to use. In this study, we targeted snowy, subarctic forests to explore new challenges to large-scale mapping. Indeed, this type of environment is minimally-structured, making registration more difficult. Moreover, the ruggedness of terrains dictates the need for a full six Degrees of Freedom (DoF) solution, with little assumption on trajectory smoothness.

Fig. 1: The boreal forest *Forêt Montmorency* presents an environment ideal to test robotic applications in a subarctic region. For lidar mapping, it offers challenging dense forest conditions. Credit: *Forêt Montmorency*.

Another difficulty brought by subarctic environments is the lack of distinctive visual features during snowy periods [3]. With images of snowy surfaces, it is challenging to extract enough features in order to support visual odometry or vision-based Simultaneous Localization and Mapping (SLAM). This precludes the use of passive camera-based localization systems, making lidar the sensing modality of choice for these conditions. A natural approach to mapping in this case is to incrementally build a 3D point cloud map from scans taken at different locations, using the Iterative Closest Point (ICP) algorithm [4]. The ICP algorithm iteratively finds corresponding points between two point clouds and looks for a rigid transformation minimizing an alignment error. This approach ensures local consistency, yet inevitably suffers from global drift [5]. This drift problem can be mitigated by SLAM techniques based on pose graph optimization [6, 7]. The key idea behind the latter is to identify loop closures and optimize all estimated transformations between the individual lidar scans to ensure global consistency. Unfortunately, in environments which do not allow loop closure (such as a long straight trajectory), drift cannot be avoided without additional external localization sources, such as Global Navigation Satellite System (GNSS). Moreover, the pose graph optimizer requires uncertainty estimations in the form of covariance matrices for all transformations, including those from ICP. As shown by Landry et al. [8], this uncertainty can be modeled, learned or sampled, but the Gaussian distribution assumption does not hold up well in complex 3D environments.

On the other hand, autonomous robots operating on polar ice sheets [9] often rely on GNSS as their main source of positioning. In unstructured environments (e.g., boreal forests, taiga), GNSS cannot be used this way due to high uncertainty of position estimates. This uncertainty is caused by interference of the canopy with the signals from satellites [10]. Still, the main advantage of GNSS is that it provides

a global source of positioning, which shows minimal and bounded bias compared to the ICP. Meanwhile, ICP creates maps that are crisp (i.e. locally consistent).

The goal of our paper is to demonstrate large-scale mapping of difficult environments, while generating maps that are *i)* crisp, *ii)* without long-term drifts, and *iii)* that can be updated swiftly. To satisfy the first two criteria, we experimented with embedding external constraints directly within the ICP minimization process using a novel formulation to handle uncertainty on positions. More precisely, we propose to augment the ICP algorithm by adding penalty terms based on the global GNSS positioning and the Inertial Measurement Unit (IMU) attitude estimates, weighed by their uncertainty. This formulation has the advantage that the uncertainty associated with these external constraints, contrary to the ICP's, can usually be readily estimated. Second, it avoids undue oscillations induced by alternating between graph minimization and point cloud registration, as both algorithms have no guarantees of sharing the same minimum. The third criterion pertains to the need for fast point-cloud map update in autonomous systems. The main bottleneck of the ICP algorithm is the update of the KD-tree, a data structure used for a fast nearest-neighbor search. Since our objective is to build large maps, this slowdown becomes unacceptable. We investigate a simple optimization technique to reduce the execution time of registrations by reducing the size of the map portion used for the KD-tree update. We tested our approach on large-scale maps recorded in a boreal forest during the winter (shown in Fig. 1). A particular emphasis has been placed on discussing problems related to the different aspects of lidar mapping for this type of environment.

2 Related Work

The context our work involves robotic deployments in harsh, snowy environments. This problem of mapping and localization in these conditions has only been investigated by a few publications. For example, visual SLAM for a robotic snowmobile platform was deployed by [11] on a glacier in Alaska. The authors report difficulties relating to the relatively low number of visual features in close vicinity of the mapping platform, compared to visual features located on the horizon. Since the nearby features are vital for translation estimation, image processing techniques are proposed to improve their extraction. Effects of changing shape and appearance of snowy areas on a path-following algorithm are also discussed in [12]. Their findings further motivate the use of lidars for mapping in these conditions. In our approach, the deployment of the lidar sensor translates the problem of extracting image features for localization into the problem of locating against 3D geometry. Areas covered by the boreal forests comply well with this requirement. On the other hand, on open plateaus where 3D features are sparse, the GNSS constraints assure consistent localization and mapping. Moreover, similarly to [11] and [12], we do not require wheel or track odometry measurements from the mobile platform. This feature simplifies integration of the mapping system into different mobile vehicles which do

not offer any odometry (in our case, the sleigh). We, however, benefit from an IMU which provides attitude prior with unbiased roll and pitch angles.

Laser scans can be captured from a ground-based static sensor as in the work of [13] who employed a stationary high density lidar sensor for terrain classification and identification of tree stems. Alternatively, the Airborne Lidar Scanning (ALS) approach allows the mapping of vast forested areas from the air. Besides the ALS, *Structure from Motion* technique [14] and stereo imagery [15] are further alternatives to creating 3D maps, suitable mainly for light aerial drones. Our goal is creating and maintaining globally consistent 3D maps for autonomous ground robots. In the case of ALS, the global consistency is easier to achieve because of the high-altitude point of view and unobstructed GNSS reception. Contrarily, ground robots only observe a limited portion of the area at a time and their GNSS reception is partially occluded by the canopy. On the other hand, ground-based 3D scans offer high details, also useful for in-depth vegetation analysis. The problem of storing and managing large amounts of data is common to all of the mentioned works. In our approach, we propose a technique to limit the computation demands during the mapping process.

Fueled by the increasing interest in self-driving cars, multiple large-scale urban datasets, most notably [16], (containing lidar and GNSS information beside other sensors) have become available. These datasets have accelerated development, refining a variety of visual- and lidar-based SLAM algorithms. Contrary to structured urban environments, we investigate the characteristics of mapping in unstructured ones (forests) in harsh winter conditions. Additionally, any improvement on the accuracy of registration algorithms reduces pressure on loop-closure and graph minimization algorithms, leading to more robust lidar-based SLAM algorithms overall.

From the extensive family of ICP variants [17], our contribution relates mainly to incorporating generalized noise models into the ICP algorithm. Since the GNSS positioning provides a confidence estimate in the form of a covariance matrix, simplification to an isotropic noise model discards potentially important information. Ohta and Kanatani [18] were first to consider anisotropic and inhomogeneous noise models when estimating optimal rotation of features extracted from stereo-pair depth images for 3D reconstruction. Later, the Generalized Total-Least-Squares ICP (GTLS-ICP) algorithm was introduced by Estépar et al. [19] for registering medical fiduciary markers. The work considers an anisotropic noise model in the registration phase of ICP and accounts for optimizing translation component as well. Further improvements were introduced by Maier-hein et al. [20], where the matching phase is modified to benefit both from KD-tree search speed and Mahalanobis-distance metric. This technique has been eventually enhanced by the introduction of a new kind of KD-tree which directly supports Mahalanobis-distance and a new minimizer [21]. In our approach, the anisotropic noise is strictly limited to the GNSS position measurements, making the problem slightly different. We look for a way to integrate this positioning information together with its anisotropic noise into the ICP. More closely related to robotic applications, the Generalized-ICP (GICP) algorithm [22] preserves the possibility to model measurement noise, while focusing on minimizing the plane-to-plane metric. However, using an iterative minimizer such

as Broyden-Fletcher-Goldfarb-Shanno (BFGS) inside the matching loop of ICP is prohibitively slow. In this paper, we investigate how to link point set registration to include penalty terms brought by the GNSS and IMU measurements within the same mathematical framework generic to anisotropic noise.

3 Theory

3.1 New Formulation for Point-to-Gaussian Cost Function

The ICP algorithm aims at estimating a rigid transformation $\widehat{T}$ that best aligns a set of 3D points $\mathcal{Q}$ (i.e., a *map* point cloud) with a second set of 3D points $\mathcal{P}$ (i.e., *scan* point cloud), given a prior transformation $\check{T}$. For a better representation of surfaces, the points of the map point cloud can be represented locally by planes. The problem of rigid registration using points from the scan and planes from the map [23] can be summarized as minimizing the point-to-plane cost function $J_{\text{p-n}}(\cdot)$ following

$$\widehat{T} = \arg\min_{T} J_{\text{p-n}}\left(\mathcal{Q}, \mathcal{P}, \check{T}\right), \quad \text{with} \tag{1}$$

$$J_{\text{p-n}} = \sum_{i=1} \sum_{j=1} w_{ij}(e_{ij}^T n_i)^2, \quad \text{and} \quad e_{ij} = q_i - \check{R} p_j - \check{t}, \tag{2}$$

where e_{ij} is the error vector between the ith point q of $\mathcal{Q}$ and the jth point p of $\mathcal{P}$, n_i is the normal of the plane, $\check{R}$ and $\check{t}$ are respectively the rotation and translation part of $\check{T}$, and w_{ij} is a weight limiting the impact of outliers as surveyed by Babin et al. [24]. The double summation in (1) is expensive to compute and is typically approximated using a subset of pairs using nearest neighbor points of each scan point. To simplify the notation, we will use e_m for each error to be minimized, with m being the index of this subset. Point-to-plane error outperforms point-to-point error in most cases [5]. However, it does not represent non-planar surface well. Point-to-Gaussian provides a more versatile representation [25]. Instead of being represented by a plane, each point in $\mathcal{Q}$ is the mean of a Gaussian and its incertitude is represented by a covariance. The point-to-Gaussian cost function $J_{\text{p-g}}$ thus becomes the following:

$$J_{\text{p-g}} = \sum_{m=1} \left(e^T W^{-1} e\right)_m, \tag{3}$$

where W^{-1} is the inverse of the covariance. In point-to-Gaussian, the Mahalanobis distance is minimized instead of the Euclidean distance (point-to-point) or the projected distance to a plan (point-to-plane). Instead of using a second iterative solver within the matching loop of ICP [19, 22, 21], we propose a novel decomposition to minimize the point-to-Gaussian error (3) directly using the equations for point-to-plane error (1). The inverse of the covariance W^{-1} can be expressed as a matrix

$\boldsymbol{N}$ of eigenvectors and a diagonal matrix $\boldsymbol{\Lambda}$ holding the sorted eigenvalues, with $\lambda_1 < \lambda_2 < \lambda_3$, using

$$\boldsymbol{W} = \boldsymbol{N}\boldsymbol{\Lambda}\boldsymbol{N}^T \Rightarrow \boldsymbol{W}^{-1} = \boldsymbol{N}\boldsymbol{\Lambda}^{-1}\boldsymbol{N}^T.$$

The decomposition can be inserted inside the cost function and reformulated as three point-to-plane errors using a projection for each of the eigenvector. Dropping the summation and the indices for clarity, we obtain for a single pair of points

$$\text{J}_{\text{p-g}} = \boldsymbol{e}^T \boldsymbol{N}\boldsymbol{\Lambda}^{-1}\boldsymbol{N}^T \boldsymbol{e} \tag{4}$$
$$= \boldsymbol{e}^T \begin{bmatrix} \boldsymbol{n}_1 & \boldsymbol{n}_2 & \boldsymbol{n}_3 \end{bmatrix} \text{diag}\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}\right) \begin{bmatrix} \boldsymbol{n}_1 & \boldsymbol{n}_2 & \boldsymbol{n}_3 \end{bmatrix}^T \boldsymbol{e}$$
$$= \underbrace{\frac{1}{\lambda_1}\left(\boldsymbol{e}^T \boldsymbol{n}_1\right)^2}_{\text{J}_{\text{p-n}}} + \frac{1}{\lambda_2}\left(\boldsymbol{e}^T \boldsymbol{n}_2\right)^2 + \frac{1}{\lambda_3}\left(\boldsymbol{e}^T \boldsymbol{n}_3\right)^2, \tag{5}$$

where λ_i is an eigenvalue and $\boldsymbol{n}_i$ is its associated eigenvector. Thus, point-to-Gaussian can be used with any point-to-plane minimizer. In fact, point-to-plane is a special case of point-to-Gaussian, where the first eigenvalue λ_1 is small enough compared to λ_2 and λ_3. This formulation can be used to also minimize Gaussian-to-Gaussian by setting $\boldsymbol{W}$ to the sum of the uncertainty of point $\boldsymbol{q}$ with the rotated uncertainty of its associated $\boldsymbol{p}$.

3.2 Adding Penalty Terms to ICP

ICP mapping creates crisp maps by taking into account local geometric characteristics contained in each new point cloud. Therefore, global consistency is not enforced. On the other hand, GNSS provides globally consistent positioning, but yields low local precision, especially when compared to ICP in forested areas. Furthermore, there is a disproportion between the altitude, latitude and longitude positioning components, the altitude being the least precise. By fusing ICP, GNSS and IMU information, we propose to compensate for the ICP drift.

Penalties are a natural way to add a constraint to the minimization step of ICP. They can be seen as imaginary points added to the point cloud during minimization for which the association is known. The minimization problem thus becomes:

$$\widehat{\boldsymbol{T}} = \arg\min_{\boldsymbol{T}} \underbrace{\frac{1}{M}\sum_{m=1}\left(w\boldsymbol{e}^T\boldsymbol{W}^{-1}\boldsymbol{e}\right)_m}_{\text{point clouds}} + \underbrace{\frac{1}{K}\sum_{k=1}\left(\boldsymbol{e}^T\boldsymbol{W}^{-1}\boldsymbol{e}\right)_k}_{\text{penalties}}, \tag{6}$$

where $\boldsymbol{e}_k$ and $\boldsymbol{W}_k$ are respectively the error and the covariance of the kth penalty, M is the number of matched points and K the number of penalty added. The penalty error $\boldsymbol{e}_k$ consists of two points added to their respecting frames of reference

$$e_k = {}^{\mathrm{M}}q_k - {}^{\mathrm{M}}_{S}\breve{T}\,{}^{\mathrm{S}}p_k, \tag{7}$$

where ${}^{\mathrm{M}}q_k$ is the position of the kth penalty in the map frame, ${}^{\mathrm{M}}_{S}T$ is the transformation from the scan frame (S) to the map frame (M) at the current iteration and ${}^{\mathrm{S}}p_k$ is the position of the kth penalty in the scan frame. For instance for the GNSS penalty, ${}^{\mathrm{M}}q_k$ is the GNSS's global position and ${}^{\mathrm{S}}p_k$ is the origin of the scan frame. Effects of adding penalty points are presented in Sect. 5.1.

Since the GNSS penalty points come in the form of a Gaussian distribution and point clouds contain plane normal information, equation (6) is approximated using a constant scale s in our implementation. This constant ensures the conversion from projected distances to Mahalanobis distances by assuming that $\frac{1}{\lambda_1}$ is constant for all points and neglecting $\frac{1}{\lambda_2}$ and $\frac{1}{\lambda_3}$ in (5). In this setting, the final cost function to be optimized becomes:

$$\mathrm{J_{p\text{-}g}} \approx \frac{s}{M}\sum_{m=1}\left(w(e^T n_1)^2\right)_m + \frac{1}{K}\sum_{k=1}\left(e^T W^{-1} e\right)_k. \tag{8}$$

3.3 Iterative Closest Point mapping

The mapping was achieved using a modified version of `ethz-icp-mapping` [26]. The mapper performs the following steps: 1) Move the scan to the initial estimate, 2) register the scan with the map using ICP, and 3) insert the scan inside the map. The initial estimate $\breve{T}$ is composed of a translation increment based on the GNSS positioning and change in orientation based on the IMU. The IMU heading is corrected by the GNSS positioning as long as the platform moves forward. Justification of this correction and a possible alternative are discussed in Sect. 5.1. Since this initial estimate $\breve{T}$ is utilized in an incremental manner, the mapping can diverge over time. As for the construction of the global map $\mathcal{Q}$, the whole scan $\mathcal{P}$ is not directly concatenated. Rather, only points that are farther than ε from any points in $\mathcal{Q}$ are inserted. This helps in keeping the global map uniform, without sacrificing registration precision. As the robot explores the environment, the complexity of registration grows linearly with the number of points in the map due to the KD-tree structure updates. To stabilize the mapping complexity, a scan $\mathcal{P}$ is not registered against the whole map $\mathcal{Q}$, but only against a subsection of the map within a radius r_{max} equal to the maximum range of the lidar. The effects of this optimization is shown in Sect. 5.3.

4 Experimental Setup

4.1 Data Acquisition Platform

For our experiments, we developed a rugged data acquisition platform which can withstand snow and sub-zero temperatures. It comprises an Xsens MTI-30 IMU, a Robosense RS-16 lidar, and a REACH RS+ GNSS antenna powered by two 20Ah 12V AGM batteries (10 h battery life). A small, low-power computer (AIV-APL1V1FL-PT1) records the sensor data using the Robotic Operating System (ROS) framework. This platform can be attached to most of mobile vehicles (see Fig. 2). The rotation axis of the lidar sensor is at an angle of 27° from the vertical. This orientation has been chosen for two reasons: 1) the lidar does not see the mobile vehicle nor its operators as long as they are in front of the platform, and 2) as mentioned in [27] a lower incidence angle with the ground reduces the odometry drift. The GNSS antenna is coupled with a fixed station (also a REACH RS+ antenna) mounted on a tripod to provide a Real-Time Kinematic (RTK) solution.

Fig. 2: The data acquisition platform mounted on a sleigh behind the snowmobile (*left*). This configuration limited transfer of engine vibrations to the sensors. *Right*: close-up of the GNSS receiver and the RS-16 lidar.

The Université Laval owns the *Forêt Montmorency*, the largest research forest in the world with over $412 \, km^2$ of boreal forest (Fig. 1). For our experiments, we collected data along three large loops (see Fig. 3). Two of them (i.e., lake and forest) consist of a mix of narrow walkable trails and wider snowmobile trails, while the last one (i.e., skidoo) followed exclusively a wide snowmobile trail.

More specifically, the dataset lake was recorded by mounting our platform to a snowmobile (see Fig. 2) and then having two operators pull the sleigh through a pedestrian trail. The snowmobile drove through a cross-country skiing trail, which is an open area with good GNSS coverage (A to B in Fig. 3). The pedestrian trail was a dense forest path (B to C), where the platform suffered from poor GNSS reception due to the tree canopy. The overlap between scans diminished abruptly each time branches came near the sensor. Similarly, the dataset forest consists of a pedestrian trail and a cross-country skiing trail. In the first part of the trajectory, a pedestrian trail in the dense forest was traversed with a sleigh (D to E), followed by an untapped path through an even denser forest (E to F) and finally the sleigh was attached to a snowmobile and driven back to the starting point (F to D). Lastly, the dataset skidoo follows a 2 km long snowmobile trail. The data were gathered

during a light snow fall. This loop was the easiest to map because it provided clear GNSS signal and was relatively flat from beginning to end (G to H to G).

Fig. 3: The three large loops completed in the Montmorency forest in order to collect the lake (0.8 km), forest (1.3 km) and skidoo (2 km) datasets. The map was adapted from ©Mapbox and ©OpenStreetMap contributors.

5 Field Results

5.1 Effects of Adding GNSS Penalty to ICP

In order to inject the GNSS positioning information as a constraint into the ICP algorithm, it is necessary to find the correct transformation between the ICP map coordinate frame and the local tangent East-North-Up (ENU) frame. The translation, roll and pitch angles are directly observable from the GNSS receiver and the IMU. The yaw angle (heading) measurement provided by the magnetometer part of the IMU is, however, affected by soft- and hard-iron errors induced by the mobile platform itself and by local deviations of the Earth's magnetic field. A practical solution to this problem was to observe a short initial portion of the GNSS trajectory to estimate the magnetometer heading offset. This way, the typical error of between $15°$ and $20°$ could be reduced to $3°$ or less, which led to a satisfactory initial alignment. Another approach would be attaching a second GNSS receiver antenna to the mobile platform and estimating the heading angle from the relative positions of the two antennas. Both approaches, however, require a precise RTK GNSS solution. The standard uncorrected GNSS operating under the tree canopy yields excessive error and cannot be used for this purpose.

We first only applied a single penalty point to the ICP, based on the GNSS positioning. As the green trajectory in Fig. 4 demonstrates, this approach does not provide satisfactory results. On a short straight trajectory, we see that the single-point penalty forces the ICP to follow the GNSS reference, however, the orientation estimate drifts leading to a malformed map. In the Fig. 4, this effect manifests itself as a slow rise in the pitch angle.

To fully constrain the ICP and avoid both orientation and position drift, we increased the number of penalty points to three, still following (8). The additional two points lie on the gravity and on the heading vectors as follows: In the map frame, one point lies below the GNSS position in the direction of the z axis. The second point lies in the x-y plane, in the direction of the current heading as indicated by the IMU and GNSS. In the scan frame, these two new points are accordingly projected

Fig. 4: Effect of adding penalty points to the ICP. *Top*: side view of a straight trajectory segment along with orientations represented by arrows. *Bottom*: corresponding pitch angles, unavailable for GNSS. The test was done in an open area with good GNSS coverage.
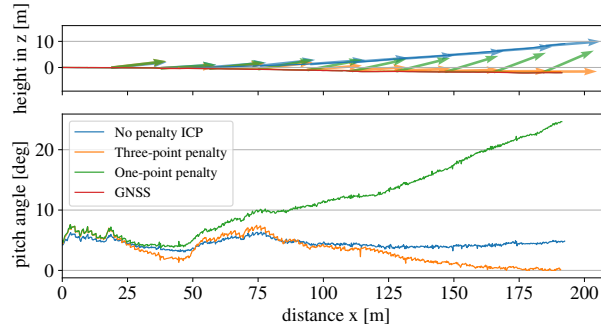
using the IMU orientation information. In the ideal case, all three penalty points in the scan frame coincide with their map counterparts. Otherwise, the penalty is forcing the ICP solution towards the ideal state. The effect is demonstrated in the Fig. 4 by the orange trajectory; the ICP output follows the GNSS positioning while keeping the correct orientation as well.

5.2 The Forêt Montmorency Dataset Results and Discussion

For each dataset, three mapping configurations were evaluated: GNSS+IMU (i.e., prior), ICP with penalties (i.e., penalty) and ICP without penalty (i.e., baseline). When processing, the `ethz-icp-mapping` was used with $r_{max} = 100$ m and $\varepsilon = 5$ cm for `lake` and `forest`. The `skidoo` dataset uses $\varepsilon = 10$ cm due to its size and memory requirements. The resulting maps are shown in Fig. 5. **Lake** – In this dataset, the baseline map looks similar to the penalty map when observed from top. However, a side view clearly shows that the buildings at the start and the end of the trajectory do not match. This effect is avoided by applying the penalty. For each configuration, we highlighted a pine tree in the top part of the map. While the tree is crisper in the baseline, the penalty map's tree is clearly sharper than the one from the prior map. **Forest** – The particularity of this map is the rough trail at the bottom of the map (see subfig. `III` and `IV` of Fig. 6). It suffers from large circular artifacts caused by the platform being immobile and by major changes of orientation along that trail. Again, the penalty is less clear than the baseline, but quite an improvement compared to the prior map. The circular-shaped building at the top part of the map is quite blurry in both the prior and penalty map. This lack of crispiness is caused by the start and end of the trajectory not correctly matching. **Skidoo** – The baseline map is the most bent of all three datasets. This bending has two probable causes: 1) the map was created with lower density $\varepsilon = 10$ cm contrary to the other maps 2) the trajectory is twice as long as the `lake` dataset. Otherwise, the behavior is similar to that of the other datasets with the similar circular-shaped artifacts in the prior map.

None of the baseline maps manages to close the loop, they all drift and bend over time. Because of the magnetometer-based heading, the prior maps show large cir-
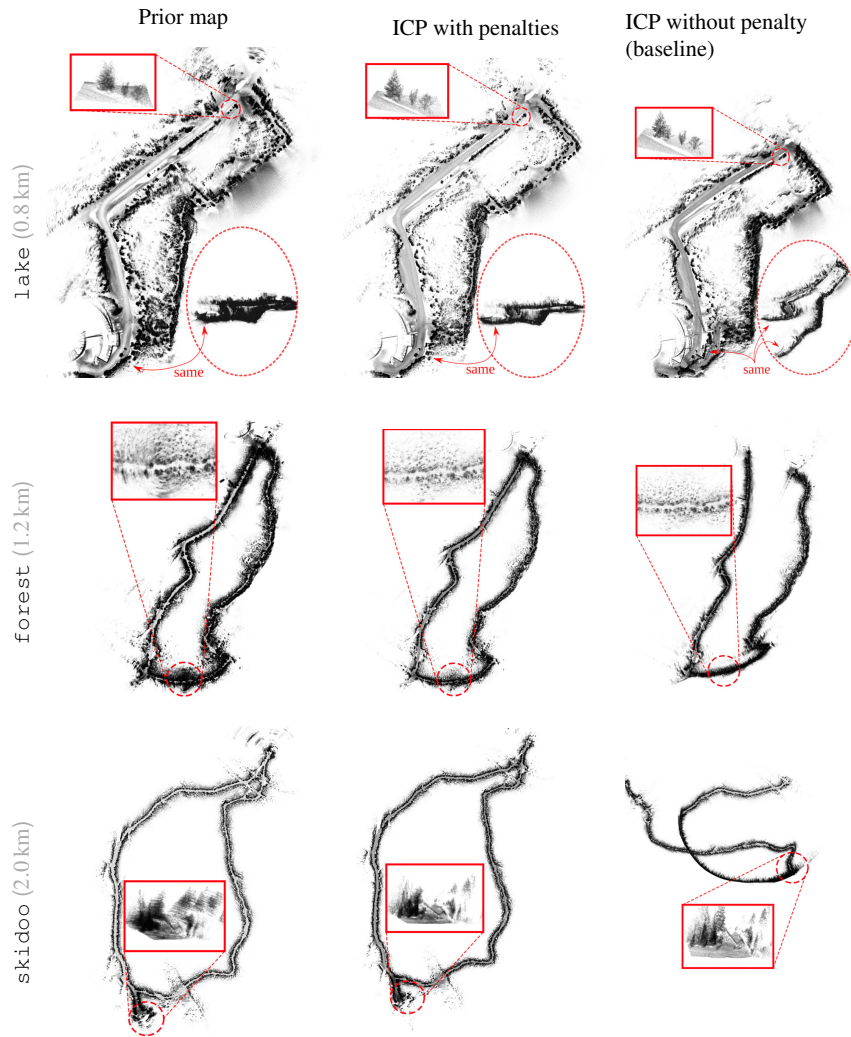
Prior map   ICP with penalties   ICP without penalty
                                                (baseline)

Fig. 5: Top view of the point cloud maps of the three datasets; prior, penalty and baseline configurations. For the lake dataset, a side view of the map (red dashed ellipse) shows the misalignment between the start and end. Red insets show the local (in)consistencies otherwise not visible at the full scale. Some trees in the insets are up to 15 m in height.
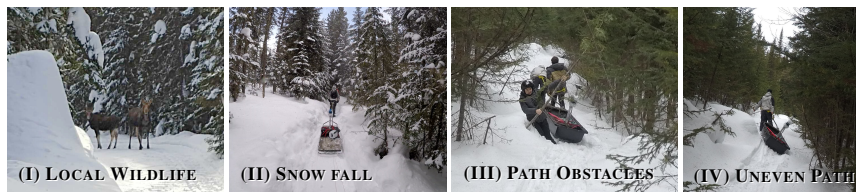

(I) Local Wildlife   **(II) Snow fall**   **(III) Path Obstacles**   **(IV) Uneven Path**

Fig. 6: Aspects of mapping a subarctic boreal forest. (II) shows snow fall caused by the snow in the tree branches. (I) was taken in skidoo, (II) in lake, and (III)-(IV) in forest.

cular artifacts at several locations. These artifacts are especially noticeable in zones where the platform stops moving—the heading cannot be corrected by GNSS in this case. All penalty and prior maps closed the loop in a similar fashion. Moreover, the penalties manage to achieve a trade-off between the global and local consistency.

The field trials at the Forêt Montmorency presented a number of challenges. As `(I)` of Fig. 6 shows, local wildlife might hinder your experiments. We had a pair of moose blocking one of our trajectories and the experiment had to be rescheduled for another day. Also, wild birds used our static GNSS antenna as a perch. Another challenge is snow fall, as even a light snow fall will be visible in the map. Furthermore, even when our experiments were done on a clear day, they were still affected by snow falling from the trees (see subfig. `II` of Fig. 6). In the `forest` trajectory, we had to pass through an untapped trail with trees blocking our way `(III)`. Because of the roughness of that trail, the platform almost tumbled over multiple time `(IV)`. The snowmobile trails, on the other hand, were easy to pass through.

Cold can cause hindering issues with lidars that are not properly rated for low temperatures. We tested a lidar rated for a minimum of $-10\,°C$, with exterior temperature during our field experiments varying between $-17\,°C$ and $-7\,°C$. The lidar started to malfunction when the temperature dropped below $0\,°C$, producing spurious measurements to a level where the environment could not be seen anymore. Providing an exterior source of heat could mitigate the problem temporary.A second lidar rated for $-20\,°C$ was used for our final experiments. Through the development process, we observed that sun glare was more apparent at low temperatures. More tests are required to fully understand the impact of cold on lidar, but one should be careful regarding the temperature ranting of sensors deployed as it can cause serious safety issues.

5.3 Steps towards Real-Time Large-Scale Lidar Mapping

The large-scale point cloud maps bear several problems that complicate real-time deployment on autonomous vehicles. The processing time required to register the scans and update the map may limit the agility of the vehicle. Moreover, memory management needs to be taken into consideration. For example, the $\sim$23 700 000 data points of the `forest` final prior map consumed 1.8 GB when stored in RAM. To improve the mapping speed, we have implemented the r_{max} cut-map radius as defined in Sect. 3.3. As shown in Fig. 7, the mapper execution time is reduced. One can observe a flattening in the number of points in the cut-map around 250 s to 500 s. This situation occurs when the mobile platform is immobile.

Finally, in order to achieve globally consistent maps, we used the RTK GNSS solution consisting of two receivers, one static and the other attached to the mobile platform. The precise positioning information is obtained by combining the information from both receivers. In our case, it was done during post-processing of the dataset. For real-time deployment, it is necessary to reliably transmit the static receiver information to the mobile vehicle, which may be difficult in dense vegetation.
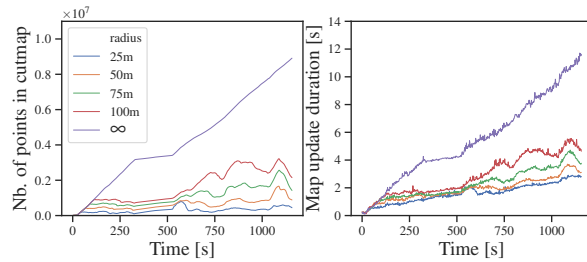
Fig. 7: Effect of the cut-map radius r_{max} on the complexity over time for the `lake` trajectory. *Left*: number of points used as a reference point cloud for the ICP. *Right*: computation time used by the mapper for each new lidar scan.

6 Conclusion

In this paper, we explored the process of creating large globally consistent maps of subarctic boreal forests by adding external constraints to the ICP algorithm. The maps remained crisp even through 4.1 km of narrow walkable and snowmobile trails. We also discussed problems encountered with the environment and the lidar sensor during the field trials. Moreover, we introduced a computation optimization for very large maps, allowing real-time deployments. Encouraged by the results, this opens the door to further comparison with Normal Distributions Transform (NDT) and GICP using better experimental validation and external tracking systems. Furthermore, studying the impact of penalties within ICP against graph minimization would lead to a better understanding of their pros and cons.

References

[1] R. Zlot and M. Bosse. Efficient large-scale three-dimensional mobile mapping for underground mines. *JFR*, 31(5):758–779, 2014.

[2] M. Pierzchała, P. Giguère, and R. Astrup. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Computers and Electronics in Agriculture*, 145:217–225, 2018.

[3] S. Williams, L. T. Parker, and A. M. Howard. Terrain Reconstruction of Glacial Surfaces : Robotic Surveying Techniques. *RA Magazine*, 19(4):59–71, 2012.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256, 1992.

[5] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets: Open-source library and experimental protocol. *Autonomous Robots*, 34(3):133–148, 2013.

[6] S. Thrun and M. Montemerlo. The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *IJRR*, 25(5-6):403–429, 2006.

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *T-RO*, 32(6):1309–1332, 2016.

[8] D. Landry, F. Pomerleau, and P. Giguère. CELLO-3D: Estimating the Covariance of ICP in the Real World. In *ICRA*, 2019.

[9] J. H. Lever, A. J. Delaney, L. E Ray, E. Trautmann, L. A. Barna, and A. M. Burzynski. Autonomous GPR Surveys using the Polar Rover Yeti. *JFR*, 30 (2):194–215, 2013.

[10] G. Jagbrant, J. P. Underwood, J. Nieto, and S. Sukkarieh. LiDAR Based Tree and Platform Localisation in Almond Orchards. In *FSR*, pages 469–483, 2015.

[11] S. Williams and A. M. Howard. Developing Monocular Visual Pose Estimation for Arctic Environments. *JFR*, 71(5):486–494, 2009.

[12] M. Paton, F. Pomerleau, and T.D. Barfoot. In the dead of winter: Challenging vision-based path following in extreme conditions. *STAR*, 113:563–576, 2016.

[13] M. W. McDaniel, N. Takayuki, C. A. Brooks, P. Salesses, and K. Lagnemma. Terrain Classification and Identification of Tree Stems Using Ground-based LiDAR. *JFR*, 29:891–910, 2012.

[14] L. Wallace, A. Lucieer, Z. Malenovsk, D. Turner, et al. Assessment of Forest Structure Using Two UAV Techniques: A Comparison of Airborne Laser Scanning and Structure from Motion (SfM) Point Clouds. *Forests*, 7(3), 2016.

[15] J. Tian, Reinartz P., P. d'Angelo, and M. Ehlers. Region-based automatic building and forest change detection on cartosat-1 stereo imagery. *ISPRS*, 79:226 – 239, 2013.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.*, 2013.

[17] F. Pomerleau, F. Colas, and R. Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. and Trends in Rob.*, 4(1):1–104, 2015.

[18] N. Ohta and K. Kanatani. Optimal Estimation of Three- Dimensional Rotation and Reliability Evaluation. *ECCV*, pages 175–187, 1998.

[19] R. S. J. Estépar, A. Brun, and C.-F. Westin. Robust Generalized Total Least Squares Iterative Closest Point Registration. In *MICCAI*, 2004.

[20] L. Maier-hein, A. M. Franz, T. R. Santos, M. Schmidt, M. Fangerau, et al. Convergent ICP Algorithm to Accomodate Anisotropic and Inhomogenous Localization Error. *TPAMI*, 34(8):1520–1532, 2012.

[21] S. D. Billings, E. M. Boctor, and R. H. Taylor. Iterative Most-Likely Point Registration (IMLP): A Robust Algorithm for Computing Optimal Shape Alignment. *PLoS ONE*, pages 1–45, 2015.

[22] A. V. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *RSS*, 2009.

[23] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, pages 2724–2729, 1992.

[24] P. Babin, P. Giguère, and F. Pomerleau. Analysis of Robust Functions for Registration Algorithms. In *ICRA*, 2019.

[25] R. Balachandran and J. M. Fitzpatrick. Iterative Solution for Rigid-Body Point-Based Registration with Anisotropic Weighting. *SPIE*, 7261:1–10, 2009.

[26] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart. Long-term 3D map maintenance in dynamic environments. In *ICRA*, pages 3712–3719, 2014.

[27] J. Laconte, S.-P. Deschênes, M. Labussière, and F. Pomerleau. Lidar measurement bias estimation via return waveform modelling in a context of 3d mapping. In *ICRA*, 2019.