

Reducing spatial error in mobile laser scanning by real-time uncertainty visualization and human-machine interaction

M. Trzeciak, C.J. Burgoyne and I. Brilakis

Department of Engineering, University of Cambridge, United Kingdom

E-mail: mpt35@cam.ac.uk, cjb19@cam.ac.uk, ib340@cam.ac.uk

Abstract –

Scanning is a key element for many applications in the AECO industry. It provides point clouds used for construction quality assurance, scan-to-BIM workflows and construction surveys. However, data acquisition using static laser scanners or photogrammetry methods is lengthy and requires even lengthier subsequent processing. A quick and apparent escape from this problem might be mobile mapping solutions mainly based on lidars. However, current hand-held scanners suffer from drift, skewing point clouds and thus, increasing their spatial error. In this paper, we present a novel, real-time and fully explainable method exploiting human-machine interaction to increase the correctness of produced point clouds. Our method progressively reconstructs the scanned scene and predicts the regions of a potentially high error with a 95% confidence level. The user can then revisit these parts of the scene, which adds additional constraints on the underlying probabilistic graphical model, thus reducing the drift and increasing the confidence in the correctness of these regions. We build a prototypic lidar-based mobile scanner, implement our method and test it in a case study. The results show that the areas identified with a relatively high spatial error indeed suffer from it, while predicted areas with relatively higher correctness do have a smaller spatial error.

Keywords –

Mobile mapping; SLAM; Digitization; Pose graph; Uncertainty visualization; Uncertainty propagation; Human-machine interaction

1 Introduction

In this paper, we propose a novel mobile scanning technique aiming at reducing a spatial error in point clouds based on the user-scanner interaction. As the user traverses a scene, our real-time method visualizes the uncertainty related to the correct position of points in a

progressively-built point cloud, hence informing the user on the potentially increasing spatial error. The user can then take corrective actions on-the-fly by revisiting places with a potentially higher spatial error, thus imposing additional constraints on the underlying optimization problem and hence reducing the error.

Digitizing the geometry of existing assets is a key element for many use cases in the Architectural, Engineering, Construction and Operation (AECO) industry. However, data acquisition using static laser scanners or photogrammetry methods is lengthy and requires even lengthier subsequent processing [1], [2]. A quick and apparent escape from this problem might be mobile mapping solutions, mainly based on lidars.

However, the current state-of-practice mapping devices do not allow for scanning with high accuracy [3], [4]. Therefore, it is not uncommon that there is a mix of static and mobile scanners on construction sites depending on the requirements of the use cases at hand [3].

We define a spatial error as a distribution of distances between points by a mobile scanner and their corresponding ground truth. This corresponds with the geodetic “correctness” of point clouds [5], “absolute accuracy” [6] or, simply, “accuracy” [1]. We will use these notions interchangeably in this paper. With that in mind, the problem statement is that point clouds produced by current mobile scanners suffer from relatively higher spatial error because of drift increasing over time in Simultaneous Localization and Mapping (SLAM) systems [7], [8].

We propose a novel and fully explainable real-time scanning method based on the user-scanner interaction. As the user traverses the scene, our system propagates the uncertainty of odometric inter key-pose constraints in the underlying pose graph. Then, with a high confidence level, it computes the largest variability related to each of the key poses and visualizes it on the progressively-built point cloud using colours. Since the colours displayed on the 3D points are based on comparing the variability to accuracy levels/bands from surveying standards, our method is dedicated specifically to construction use cases.

The user can thus see which regions of the point cloud are likely to suffer from higher spatial error and can revisit them, which adds additional constraints onto the pose graph, thus reducing the uncertainty in the key poses.

The proposed system can be then integrated into the practical mobile scanning procedures in the following way. Before scanning starts, the user chooses the top and bottom levels of point cloud correctness they are willing to accept. As the user traverses the scene, the progressively-built point cloud gradually changes colour from green to red, indicating that the spatial error of the red regions of the scene is beyond the set lower level. After the desired parts of the scene have been scanned, the user revisits these parts of the scene that are coloured red, ideally linking them with green areas.

We build a prototypic scanner consisting of a lidar and a laptop, implement our method and test it in a case study. With a 95% confidence level, the results show that our system correctly predicts the regions of both relatively high correctness as well as those suffering from high spatial errors.

Before we proceed to the specifics of our method, however, the subsequent section introduces the reader to the state of the art (SOTA) pose-graph SLAM systems and uncertainty propagation in their underlying models.

2 Background

2.1 Pose-graph SLAM

According to [9], there are two probabilistic SLAM paradigms: 1) filtering and 2) smoothing. The former focuses on the estimation of the most current pose of the scanner given all the measurements of its sensors. It is useful in the case of robots that must determine their position in real-time as accurately as possible. The latter, in turn, focuses on the estimation of all the key poses comprising the trajectory of the scanner. Given the fact that the correctness of the produced point cloud is a function of the trajectory, its wrong estimation will yield a skewed point cloud resulting in its higher spatial error. Therefore, the smoothing SLAM paradigm is of interest in this paper because it focuses on the estimation of the whole trajectory, and hence the correctness of the produced point cloud as a whole.

There exist a number of SLAM methods under the smoothing paradigm, some of which are described by [9], [10] and [11]. Although these authors name their methods differently, their approaches share the same core. They model this problem using a probabilistic graph and then turn it into a problem involving minimizing non-linear least squares. Even with a relatively good initialization, there is no guarantee that such a problem can be executed in constant time due to iterations during the optimization and is generally considered

computationally expensive [12]. However, in the light of increasing computing power of mobile devices and recent scientific advancements in effective factorization methods [13], solving graph-based SLAM in near real-time has become increasingly possible.

The type of sensors used for SLAM also affects how the problem should be modelled. In cases where lidar is involved, it is not uncommon to avoid modelling an explicit map of the scanned scene and instead focus on the trajectory of the scanner only [14] since a stream of lidar points can yield relatively good odometric constraints between key poses [9]. Modelling the problem this way is otherwise known as pose-graph SLAM and involves only the mentioned odometric constraints and loop closures. The latter is important since they provide constraints that allow creation of a globally consistent trajectory, and therefore a globally consistent map.

2.2 Uncertainty propagation in non-linear least squares

On one hand, pose-graph SLAM under the smoothing paradigm still suffers from drift, which cannot be eliminated [15]. On the other hand, information about uncertainty of the key poses can help to localize those parts of the trajectory that suffer from drift. In this vein, the uncertainty/error of the inter-pose constraints can be propagated through the pose-graph so that the joint probability of the key poses can be computed [16]. From there, uncertainty on the individual poses can be calculated through the means of marginal covariances. Since the smoothing paradigm of SLAM can be viewed as a more general non-linear least-squares minimization problem, it is of interest to investigate the methods that propagate uncertainty through such systems. Some of them are presented in [17], [18] and [19].

2.3 Gaps in knowledge

On one hand, SLAM frameworks still suffer from growing drift, resulting in worsening spatial error of point clouds. They are not as accurate as the workflows based on static scanners and surveying, and there is no way to remove the drift in a user-aware manner. On the other hand, there are also ways to propagate the uncertainty in SLAM systems; however, they are either (1) rather theoretical, yielding faster and faster approximate methods of recovering marginal covariances, or (2) have applications in the detection of loop closures where more exact marginal covariances allow for fewer candidates among key-poses to be searched for to find the best match.

2.4 Research objectives

The point of this paper is to provide a real-time method for uncertainty visualization in pose-graph SLAM so that the user can be aware of regions of potentially higher spatial error and they can take corrective actions on-the-fly (during scanning).

3 Proposed solution

3.1 Scope & assumptions

Our solution is designed for SLAM systems working on pose-graphs. We assume that our inter key-pose and loop-closure constraints do not fail in any way during the execution of our system. The spatial error predictions are based on the translational part of marginal covariance matrices Σ_i .

3.2 Overview

The core of our idea is to pass the uncertainty encoded in marginal covariance matrices Σ_i of the estimated trajectory X onto the lidar points PC_i^{lidar} that are associated with the key poses x_i , and scene-reference the points. See Figure 1 for reference. Each key pose comprises a 3D rotation matrix (3×3) and a translation vector (3×1) defined in the coordinate system of the scene. The user can then intuitively see what areas of the progressively-built scan suffer from a potentially high spatial error after scene-referencing the lidar points. As explained in section 2.2, there are SOTA real-time methods to propagate the uncertainties $\{\Omega_1, \Omega_2, \dots\}$ of individual inter key-pose transformations $\{z_1, z_2, \dots\}$ in such a way that the joint probability of all the key poses can be computed. There are also real-time methods to recover the marginal covariances of the key poses after the propagation. Our idea is to pass these marginal covariances Σ_i in real-time onto the scene-referenced lidar points.

More specifically, our method is presented in Figure 2. As the trajectory of a mobile scanner grows, we progressively compute the greatest variance for each marginal covariance matrix Σ_i associated with the new key pose x_i . This will ensure that we find the maximal translational error for the new key pose. Given that the errors along each of the three axes might be correlated, we compute eigenvalues λ_k of the covariance matrix according to Equation (1) and pick the largest, marked as λ_L in processes (a) and (b) in Figure 2 respectively. v_k is an eigenvector associated with the corresponding eigenvalue.

$$\Sigma v_k = \lambda_k v_k \quad (1)$$

However, such computed variance itself is not a

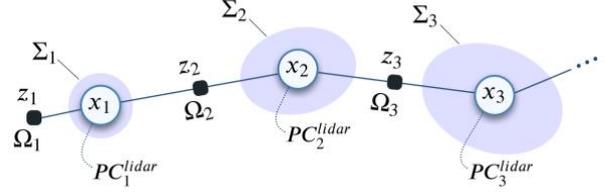


Figure 1. Part of progressively-built trajectory X consisting of a set of key poses $\{x_1, x_2, x_3, \dots\}$ represented as a pose graph. Black rectangles represent constraints Z including inter key-pose odometric transformations $\{z_1, z_2, z_3, \dots\}$. Each odometric transformation comes with an information matrix Ω_i representing uncertainty on these transformations. Pose uncertainties are marked in purple and are represented by marginal covariance matrices $\{\Sigma_1, \Sigma_2, \Sigma_3, \dots\}$

useful statistic in practice. Therefore, we convert it into a standard deviation and multiply it by 2. We thus obtain a 95% confidence level on the maximal translation error of the current pose. We choose a 95% confidence level to comply with land surveying guidance documents such as those by the Royal Institution of Chartered Surveyors [20] or specifications of Levels of Accuracy (LOA) [5] where this level of certainty is de facto a standard.

Next, we compare the two standard deviations to the accuracy levels or bands defined by USBSD or RICS respectively. Before, scanning starts, the user chooses which guidance document they want to comply with. Our system then compares $2\sqrt{\lambda_L}$ for each key pose against the accuracy levels according to the chosen standard.

We propose to colour-code the scene-referenced lidar points according to the accuracy levels. The most restrictive accuracy level (for example LOA 10 by [5]) is coloured in green while the bottom level (for example LOA 50) is red. All levels in between are then colorized according to hues in between these two, such as yellow, amber and orange. It is likely that current mobile devices will be unable to meet the requirements of the most stringent accuracy levels. Hence, we propose to cap the highest level to the one picked by the user, for example, LOA 30. In such a case, all the poses whose $2\sqrt{\lambda_L}$ are smaller than the maximal error associated with LOA 30, will remain green. A similar cap can be imposed for the bottom level. For example, if $2\sqrt{\lambda_L}$ is greater than the error associated with LOA 40, then the points related to this pose will be red.

Finally, for each new key pose, we transform lidar points in the lidar coordinate system PC_i^{lidar} to the scene coordinate system PC_i^{scene} according to Equation (2) and process (e) in Figure 2, with x_i , PC_i^{lidar} and PC_i^{scene} stored in homogenous coordinates. This way, the user scanning a scene can see what the predicted correctness

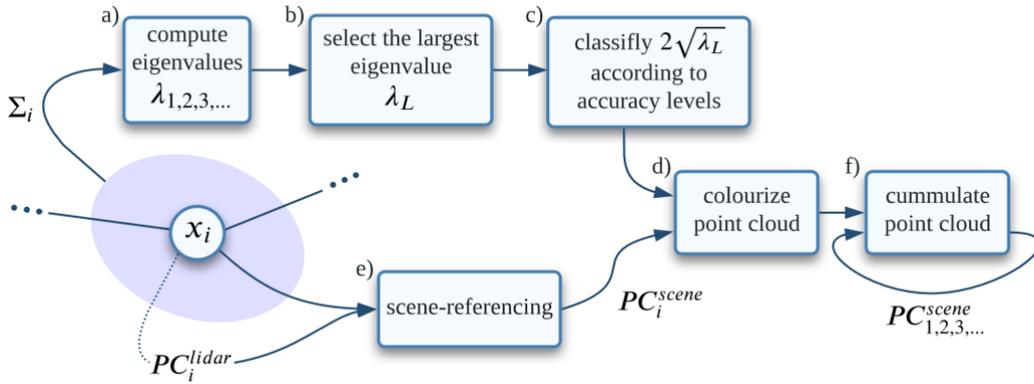


Figure 2. For each key pose x_i the following processes presented in this figure are executed so that the marginal covariance Σ_i is passed onto a scene-referenced pointcloud PC_i^{scene} and visualized to the user in real time.

of the progressively growing point cloud is in real-time.

$$PC_i^{scene} = x_i PC_i^{lidar} \quad (2)$$

The visualization system presented above can then be used in the following way. Before scanning starts, the user chooses the top and bottom levels of point cloud correctness they are willing to accept. As the user traverses new parts of the scene, the progressively-built point cloud gradually changes colour from green to red, indicating that there is a 95% chance that the spatial error of the red regions of the scene is beyond the set lower level. After the desired parts of the scene have been scanned, the user revisits these parts of the scene that are coloured red. By revisiting them, the user closes loops in the underlying pose graph representing the trajectory of the scanner, hence adding additional constraints to the graph. The underlying SLAM optimization process then shifts the trajectory, and hence the lidar points to a more correct position thanks to these additional constraints. This increases the confidence level in the correctness of the key poses and scene-referenced lidar points.

3.3 Hypothesis

For hand-held lidar-based scanners with pose-graph SLAM, real-time predictions on the correctness of the produced point cloud can enable the user to make informed corrective actions during the scanning process, thus increasing the correctness of the point cloud. The corrective actions are revisiting previous parts of the scene in an informed way.

4 Research methodology

4.1 Data collection

We built a prototypic scanner consisting of a

Velodyne VLP-16 lidar and a MacBook Pro laptop, both plugged into a portable power unit as shown in Figure 3. We coded up software for the scanner on Linux Ubuntu 20.04 with Robot Operating System [21] (version Noetic) using many own and publicly available repositories and frameworks.

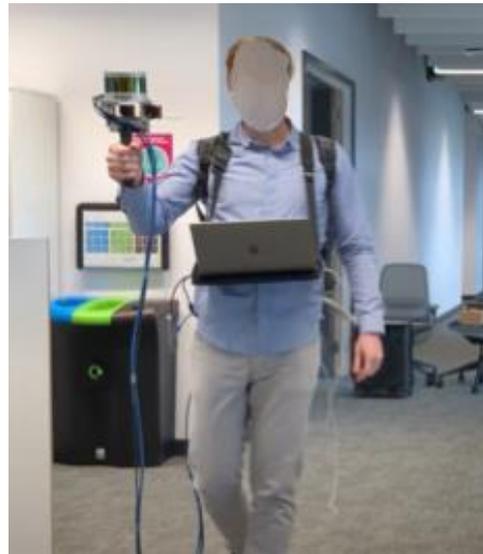


Figure 3. Our prototypic scanner connected to a laptop during scanning.

Next, we went to one of the colleges and scanned its Front Court with the scanner to test our method. During scanning, we walked along the four walls so that the façade was captured, and we returned to the place we started scanning to close a loop. The estimated trajectory of the scanner can be seen in Figure 4. In addition, we used a FARO Focus 3D terrestrial scanner [22] to provide a ground truth scan. The scanner was placed in the middle of the court so that it could cover all four façades.

4.2 Methods

We follow a Maximum A Posteriori (MAP) incremental non-linear pose-graph optimization approach by [13], to find such key poses of trajectory X that their errors for the odometric inter key-pose and loop closure transformations z_i are the smallest (Equation (3)).

$$X^{MAP} = \min_X \sum_i \|h_i(x_i) - z_i\|_{\Omega_i^{-1}}^2 \quad (3)$$

In Equation (3), $h_i(\cdot)$ are non-linear functions transforming key poses stored in the coordinate system of the scene to the coordinate system of the previous pose in which z_i is measured. Computation of the odometric constraints is based on the Lidar Odometry And Mapping technique by [7].

We estimated the information matrices Ω_i in such a way that after the loop closure event shown in image d) of Figure 4, the most uncertain part of the point cloud (around the top corner) is coloured red and the other parts are green for LOA 10 to 50.

5 Results & Discussion

5.1 Raw results

In Figure 4, we present the progressively created point cloud with overlaid uncertainty information. At the start of the reconstruction (image a), all the points in the scene are green, indicating prediction in their relatively correct position. As we traverse the scene, the drift increases and so does the estimated spatial error in the newly accumulated points. This is shown first in amber and then in red in image (b), predicting that the relative error in these places is around 3-4 times higher than in the bottom part of the scene. Having this information and following our method, we decided to return to the green area to close the loop and potentially increase the correctness of these red points. However, as we go towards the place we started the scanning, the newly reconstructed parts of the scene on the right side of the scene turn even more red (image c), indicating that the error there might be 5 times higher than in the green areas. Finally, we close the loop in image (d), and the points around this area turn green. The system predicts that their spatial error decreased significantly. However, the points in the corner at the top part of the scene remain amber.

We will work on the point cloud shown in image (d) of Figure 4 and investigate what the actual spatial error in the amber region (top corner) is and contrast it with the spatial error in the top right corner, which should have a relatively lower error according to the predictions of our system.

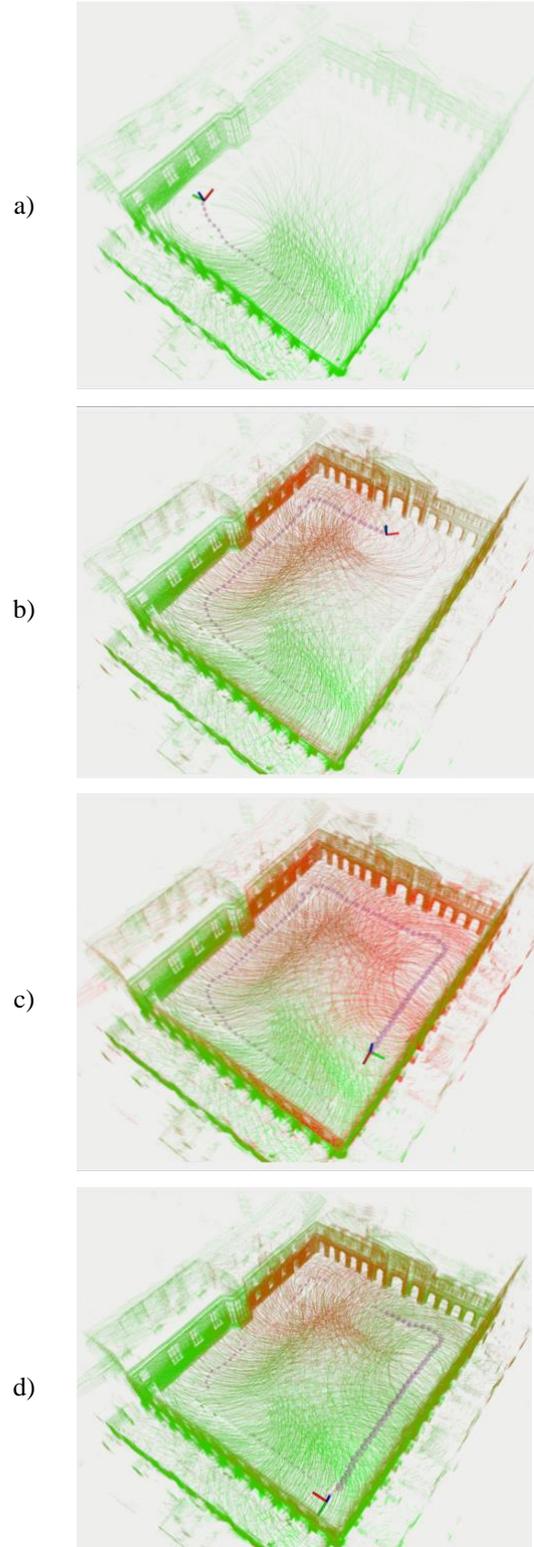


Figure 4. Progressively-built point cloud in real-time with overlaid uncertainty. Green indicates high point cloud correctness while red stands for predictions for a relatively higher spatial error.

5.2 Analysis

We segmented the point cloud by our scanner into three regions (all marked in blue in Figure 6): the one at the bottom of the figure was used to register the whole point cloud by our mobile scanner to the point cloud by the FARO Focus 3D scanner; the one at the top of the figure was predicted by our system as having a bigger spatial error (see Figure 4 d); finally, the region in the top right of the figure was predicted for a relatively lower spatial error than the previous region after the loop closure (again, see the corresponding Figure 4 d), although initially, the error there was even higher than that at the top corner (Figure 4 c).

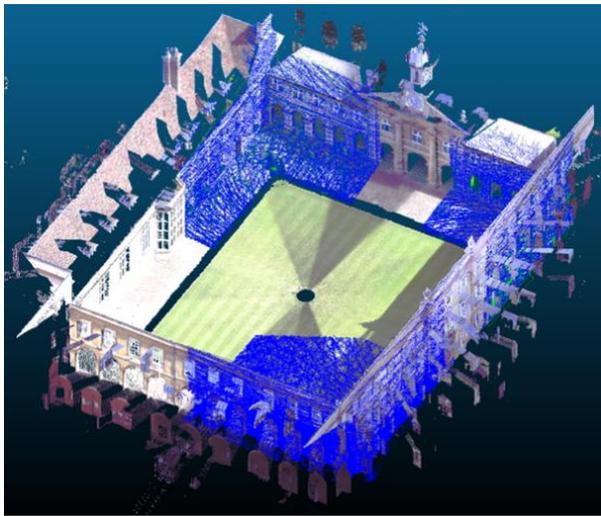


Figure 6. A scan from a terrestrial FARO Focus 3D scanner was used to provide ground truth (colourful point cloud). The segmented blue point cloud comes from our mobile scanner.

For the top two regions, we computed the distances to the ground truth scan using Cloud Compare. Next, we binned them and present them as histograms in Figure 5. On these distances, we also computed three statistics: 1) a mode (most likely value), a mean (average value) and a 95-percentile (see Table 1).

Judging by the modes in Table 1, the blue region at the top of Figure 6 has almost 5 times higher most likely spatial error than the point cloud in the top right corner (47 and 10 mm respectively). It also has around 39% higher mean error, and its 95-percentile is larger by 9 mm. These three statistics seem to confirm that our system correctly predicted the regions of relatively higher spatial error.

In addition to measuring the spatial error above, we also took a closer look at the loop closure event that occurred right before image (d) in Figure 4 was taken. In Figure 7, the initial key poses comprising the odometric trajectory (marked with a dashed light blue line) had very

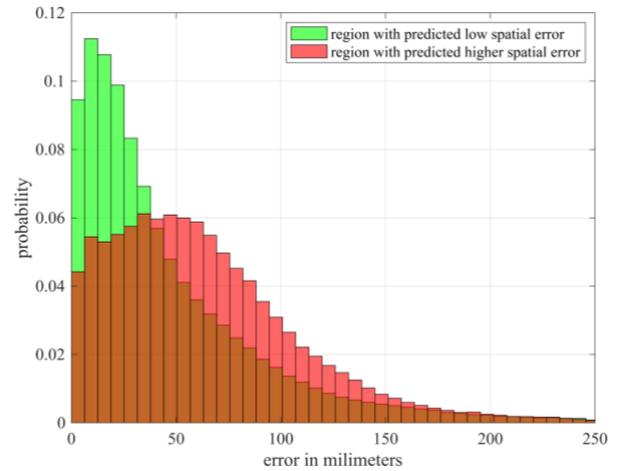


Figure 5. Distribution of error between the ground truth and the regions of a point cloud identified by our scanner as having a low spatial error (green bins) and a high one (red).

large marginal covariances (purple spheres), indicating that the position of the related point clouds was significantly off. After computing the relative transformation between the two key poses connected by the red line, the trajectory has been recomputed which resulted in an updated trajectory marked with a green line. The updated key poses now have significantly smaller marginal covariances indicated by the much smaller spheres. In fact, these spheres are as small as those at the start of scanning (Figure 4 a). The updated trajectory and its marginal covariances on the key poses seem to be in line with other studies on this topic such as those by [23]. This shows that we have implemented our method correctly.

Table 1. The actual spatial errors corresponding to the two segmented regions of our point cloud shown in Figure 6.

Statistics	Region with predicted low spatial error [mm]	Region with predicted higher spatial error [mm]
Mode	10	47
Mean	46	64
95-percentile	140	149

In this case study, the top corner marked in red in Figure 4 (d) still has lower accuracy compared to the green part. The question then is how to improve it? The informed user could walk up to the corner while scanning, diagonally through the lawn, and close the second loop. This would further reduce the uncertainty, especially in

that region, hence decreasing the error. We are planning to show this and demonstrate the impact of closing the second loop (and maybe more loops) in the following paper in the future. The reason for not demonstrating this here is the fact that more loop closures demand even tighter integration of odometry and SLAM systems than we have currently implemented. However, it is a problem related to the architecture of our software and the method described in this paper will still hold.

5.3 Our contributions

We presented a novel fully-explainable method for real-time predictions of areas with potentially high spatial error in progressively created point clouds by lidar-based mobile scanners. The method exploits the human-machine interaction and is suitable for SLAM systems based on pose graphs. We implemented it and tested its applicability in a case study which confirmed that the identified areas of relatively higher spatial error, indeed suffer from it. To the best of our knowledge, it is the first such system in the world.

Our system can help the construction industry to reduce the effort (time and money) put generally into scanning construction sites. Since the user is aware of the regions of potentially high spatial error during scanning, they can take corrective actions on-the-fly, and not after the data collection and processing have been completed. This, in turn, might eliminate the potential risk of re-scanning.

6 Conclusions

In this paper, we presented a real-time fully-explainable method exploiting human-machine interaction to increase the correctness of point clouds produced by lidar-based mobile scanners. With a 95% confidence level, our method predicts areas of relatively higher spatial error in a progressively created point cloud. We built a mobile scanner, implemented our method, and tested it in a case study. The results show that the areas identified with a relatively high spatial error indeed suffer from it, while areas with relatively higher correctness predicted, do have a smaller spatial error in reality.

The method presented here has the potential to increase the accuracy of point clouds produced by lidar-based SLAM systems operating on pose-graphs. This, in turn, might unlock demanding use-cases such as engineering surveying or high accuracy measured building surveys, which so far have been unreachable by mobile scanners due to systematic spatial errors in point clouds caused by drift. Moreover, our method might contribute to reducing the effort put into scanning by giving real-time predictions on the quality of produced point clouds, hence eliminating the need for potential rescanning.

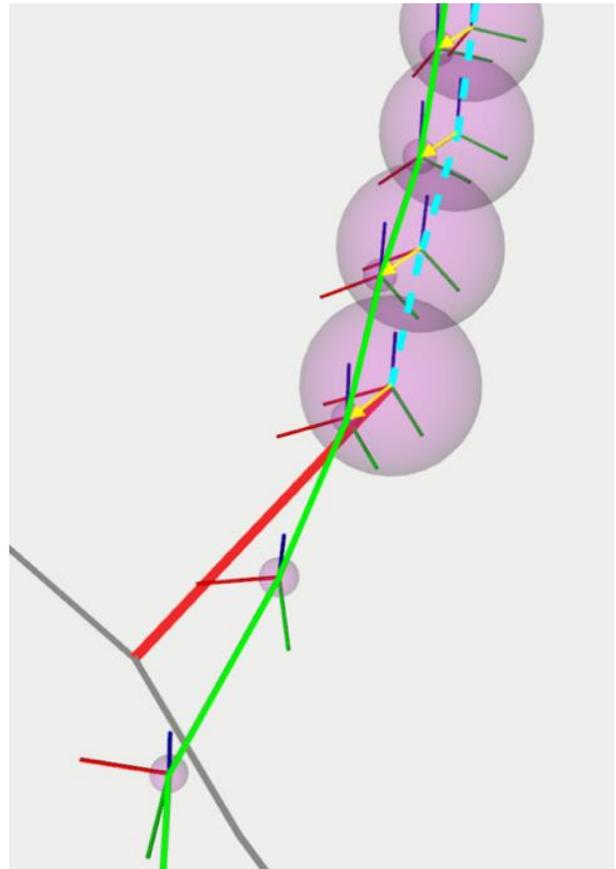


Figure 7. Correction of the odometric trajectory (dashed blue line) resulting in the SLAM trajectory (green line) in the 3D space of the scene. The red line connects the two key poses used for computing the loop closure with yellow arrows showing how the corresponding key poses shifted. Purple spheres represent marginal covariances.

In the future, we are planning to test our method in more case studies and extend it so that it yields predictions in cases where many loops have been closed.

Acknowledgements

The research leading to this paper received funding from BP, GeoSLAM, Laing O'Rourke, Topcon and Trimble. We would like to thank these companies for making our research possible. We also thank Emmanuel College for allowing access to their grounds. We gratefully acknowledge the collaboration of all the industrial partners. Any opinions, findings, conclusions or recommendations expressed in this material are ours and do not necessarily reflect the views of the companies mentioned above.

References

- [1] BIM task group, “Client guide to 3d scanning and data capturing.” 2013.
- [2] T. S. Kalyan, P. A. Zadeh, S. Staub-French, and T. M. Froese, “Construction Quality Assessment Using 3D as-built Models Generated with Project Tango,” *Procedia Engineering*, vol. 145, pp. 1416–1423, Jan. 2016, doi: 10.1016/j.proeng.2016.04.178.
- [3] NavVis, BIM+, Geo Week News, Lidar News, and GIM International, “State of Mobile Mapping: Survey,” 2021. [Online]. Available: <https://f.hubspotusercontent40.net/hubfs/3339696/State%20of%20Mobile%20Mapping%202021.pdf?hsCtaTracking=2cf01e48-9803-4273-a9cb-28a5a72bfb88%7C1003557b-fe59-47bb-a928-e17d0470e054>
- [4] M. Trzeciak and I. Brilakis, “Comparison of accuracy and density of static and mobile laser scanners,” Apr. 2021, doi: 10.17863/CAM.66857.
- [5] USIBD, “USIBD Level of Accuracy (LOA) Specification Guide.” U.S. Institute of Building Documentation, 2016.
- [6] L. Lachauer, “Get real answers about mobile scanning accuracy.” <https://www.navvis.com/blog/get-real-answers-about-mobile-scanning-accuracy> (accessed Feb. 25, 2022).
- [7] J. Zhang and S. Singh, “LOAM: Lidar Odometry and Mapping in Real-time,” 2014. doi: 10.15607/RSS.2014.X.007.
- [8] C. Cadena *et al.*, “Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
- [9] M. Kaess and F. Dellaert, “Factor Graphs for Robot Perception,” 2017. <https://ieeexplore.ieee.org/document/8187520?bkn=8187520> (accessed Feb. 02, 2022).
- [10] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, “A Tutorial on Graph-Based SLAM,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, winter 2010, doi: 10.1109/MITS.2010.939925.
- [11] Cyrill Stachniss, *Robust Least Squares for Graph-Based SLAM (Cyrill Stachniss)*, (Apr. 06, 2020). Accessed: Feb. 25, 2022. [Online Video]. Available: <https://www.youtube.com/watch?v=z60RbiY18I8>
- [12] T. D. Barfoot, *State Estimation for Robotics*. Cambridge: Cambridge University Press, 2017. doi: 10.1017/9781316671528.
- [13] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3281–3288. doi: 10.1109/ICRA.2011.5979641.
- [14] A. Jurić, F. Kendeš, I. Marković, and I. Petrović, “A Comparison of Graph Optimization Approaches for Pose Estimation in SLAM,” in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Sep. 2021, pp. 1113–1118. doi: 10.23919/MIPRO52101.2021.9596721.
- [15] L. Carlone, *The Future of Robot Perception: Recent Progress and Opportunities Beyond SLAM* | Luca Carlone, (Jul. 15, 2021). Accessed: Feb. 02, 2022. [Online Video]. Available: <https://www.youtube.com/watch?v=j5g3efgdjRg>
- [16] “Factor Graphs and GTSAM,” *GTSAM*, May 22, 2019. <http://gtsam.org/tutorials/intro.html> (accessed Feb. 25, 2022).
- [17] G. D. Tipaldi, G. Grisetti, and W. Burgard, “Approximate Covariance Estimation in Graphical Approaches to SLAM,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007, pp. 3460–3465. doi: 10.1109/IROS.2007.4399258.
- [18] M. Kaess and F. Dellaert, “Covariance recovery from a square root information matrix for data association,” *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1198–1210, Grudzie 2009, doi: 10.1016/j.robot.2009.06.008.
- [19] V. Ila, L. Polok, M. Solony, P. Smrz, and P. Zemcik, “Fast covariance recovery in incremental nonlinear least square solvers,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4636–4643. doi: 10.1109/ICRA.2015.7139841.
- [20] RICS, “Measured surveys of land, buildings and utilities. RICS guidance note, global. 3rd edition.” Royal Institution of Chartered Surveyors, 2014.
- [21] M. Quigley *et al.*, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2, p. 5.
- [22] FARO, “Technical Specification Sheet for the Focus3D X 30/130/330 and X 130/330 HDR,” *FARO® Knowledge Base*, Dec. 20, 2016. https://knowledge.faro.com/Hardware/3D_Scanners/Focus/Technical_Specification_Sheet_for_the_Focus3D_X_30-130-330_and_X_130-330_HDR (accessed Jan. 19, 2021).
- [23] D. M. Cole and P. M. Newman, “Using laser range data for 3D SLAM in outdoor environments,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1556–1563. doi: 10.1109/ROBOT.2006.1641929.