Open source robotic 3D mapping framework with ROS - Robot Operating System, PCL - Point Cloud Library and Cloud Compare

Conference Paper · August 2015

DOI: 10.13140/RG.2.1.4869.9605

5 authors, including:

Janusz Bedkowski
Mandala

Michał Pełka
Przemysłowy Instytut Automatyki i Pomiarów

Karol Majek
Institute of Mathematical Machines

Jacek Naruniec
Warsaw University of Technology

Open source robotic 3D mapping framework with ROS - Robot Operating System, PCL - Point Cloud Library and Cloud Compare

Janusz Będkowski, Michał Pełka, Karol Majek, Tresya Fitri
Laboratory of Image Processing, Modeling and Simulation Department
Institute of Mathematical Machines
Warsaw, Poland
j.bedkowski@imm.org.pl

Jacek Naruniec
Faculty of Electronics and Information Technology
Warsaw University of Technology
Warsaw, Poland
J.Naruniec@ire.pw.edu.pl

***Abstract*—We propose an open source robotic 3D mapping framework based on Robot Operating System, Point Cloud Library and Cloud Compare software extended by functionality of importing and exporting datasets. The added value is an integrated solution for robotic 3D mapping and new publicly available datasets (accurate 3D maps with geodetic precision) for evaluation purpose Datasets were gathered by mobile robot in stop scan fashion. Presented results are a variety of tools for working with such datasets, for task such as: preprocessing (filtering, down sampling), data registration (ICP, NDT), graph optimization (ELCH, LUM), tools for validation (comparison of 3D maps and trajectories), performance evaluation (plots of various outputs of algorithms). The tools form a complete pipeline for 3D data processing. We use this framework as a reference methodology in recent work on SLAM algorithms.**

Keywords—3D mapping, mobile mapping, data registration

I. INTRODUCTION

The classic problem of mobile robotics is finding the pose of a moving robot in the surrounding environment. Thus, the mobile robot should have ability for self-localization. In a known environment, precise information about it's pose enables the robot to plan further movements and activities [1]. Thus, in an unknown environment, a pose estimate is required to build a model of this environment. Assuming a wheeled robot that moves on a plane, the pose is defined as the position and orientation $\boldsymbol{X}_R = [x_r\ y_r\ \theta_r]^T$ with respect to an external coordinate system. If the map is given a priori, the robot can self-localize by matching characteristic features of the environment observed at the given moment to similar features in the known map. A feature suitable for self-localization is a salient object, or part of an object, which can be described with respect to some co-ordinate frame. The acquisition of a precise, complete and up-to-date environment model for self-localization is a tedious task. Thus, keeping such a map up-to-date is a demanding task, as the environment may change over time. Moreover, a map surveyed by hand can be perceptually incompatible, i.e., it may not properly reflect all features in the environment perceived with the given sensing modality. Therefore, the robot should be capable building its own model of the environment. This leads to the following the chicken or the egg dilemma: for self-localization, the robot requires a map, but to build such map, the pose of the robot must be known. A solution to this problem is to build the map while computing a pose estimate, which is known as SLAM [2]. In this paper we show the framework for 6DSLAM, which extends SLAM to 3D environments and allows digitizing large environments fast and reliably and solves the simultaneous localization and mapping problem. Robot motion has to cope with parameters of position and rotation, therefore the pose is given in six mathematical dimensions, including translation (x, y, z) coordinates and yaw, pitch and roll angles. 6DSLAM can be used both in indoor and natural outdoor environments. We can distinct two main representations of the map - dense and sparse. Dense maps are related to 3D laser data [3] obtained typically with 3D laser scanners. Sparse maps are related to sparse features extracted from images or also from 3D data. Dense representation is related with the dense map that is composed of the registered local maps obtained typically by the 3D range finder or the depth camera. Local maps are typically composed of 3D points. An interesting extension - an elevation map is shown in [4].

Common approach for data registration is Iterative Closest Point (ICP) algorithm introduced by Besl and McKay in [5]. The goal of ICP algorithm is to find the transformation matrix that minimizes the sum of distances between the corresponding points in two different datasets, therefore two important aspects have to be solved:

- nearest neighbour search (NNS),
- choosing the proper optimization technique for the minimization of the mentioned function (estimation of the 3D rigid transformation).

There are several approaches for the NNS for the purposes of ICP algorithm. An approach from [6] utilizes the regular grid decomposition [7], that allows for a very fast and simple processing. Another approach is kd-tree [8] that divides the space by the subsequent axes. Another promising approach - an octree based NNS is shown in [9].

Modern GPU processors can be used to efficiently compute NNS, therefore they can improve the performance of ICP. The NNS procedure is a bottleneck of the most of the ICP algorithms, therefore many of the researchers are trying to optimize the time of its execution. Complete 6DSLAM algorithm that includes loop closing was evaluated in [10]. This algorithm was embedded into Point Cloud Library [11]. Currently, most of the robotics community chooses 6DSLAM methods from two different implementations – 3DTK[12] or PCL[13]. Thus, for the purposes of algorithms evaluation we released datasets for mobile robotics community in [17].

Proposed framework is a solution for 3D data processing, in which widely used open source projects were integrated into one framework: ROS[14], Cloud Compare[16] and PCL[11]. ROS (robot operating system) is an open source framework for academic and industrial robotic applications. It combines large number of features like low-level driver for sensors, universal transport layer and ready to use implementation of the latest state of the art algorithm for data fusion, 2D and 3D mapping, path planning and robots autonomy. It offers a set of well-tested visualization and debugging solutions. Currently ROS is available only for Linux platforms.

Cloud Compare is an open source, Qt-based application for point cloud and mesh manipulation, visualization and processing. It is widely used by professionals, because of its stability and possibility to import or export almost any known point cloud format. It can be easily used for visualization of large datasets. Plug in-based architecture allows for easy integration with new features, such as our approach for importing datasets from the robot.

Point Cloud Library is an open source, template based, multiplatform point cloud processing library. The advantage of the PCL library is ready to use implementation of the latest state of the art point cloud and mesh processing algorithms. Best methods of data registration, relaxation, down sampling, up sampling and more are implemented. At this moment there are also ROS packages that use PCL, and PCL is a part of ROS framework for on-line data processing.

The paper is organized as follows: at first we presented mobile mapping system, then we proposed various mobile mapping scenarios. In the subsequent sections we presented the 3D data registration framework overview and new datasets. Finally we demonstrated experiment showing use case of proposed framework. In the last section we summarized presented work.

II. MOBILE MAPPING SYSTEM

Figure 1 shows mobile mapping system – mobile robot equipped with 3D laser measurement system Ladybug 3 spherical camera. System provides 3D colored point clouds with range up to 60 meters. It covers our first scenario of a robotic mapping system presented in section III.

System is a combination of mobile platform (Clearpath Husky), spherical camera (Pointgrey LadyBug 3), 3D laser scanner (SICK LMS- 500 with rotating unit), SICK LMS-100 2D horizontal rangefinder, IMU and GPS unit. The most interesting parts of the mounted system are spherical camera and 3D laser unit rotated around Z-axis of device. RGB camera is mounted above laser scanner unit and is static, but the scanning head is able to rotate constantly, without stopping. Minimal velocity of the head rotation is 10 rpm, maximal - 35 rpm, and it allows for adjustment of the resolution of the produced point cloud. Field of view of a laser is a full sphere affected by robot chassis, while the field of view of spherical camera is $360^{\circ} \times 320^{\circ}$.

System is currently integrated with ROS (Linux) and Windows, but ROS was used due to integration with other systems. The camera has closed source SDK for Windows system, but, fortunately, data acquisition from the camera can be performed in Linux platforms. Currently post-processing, and data fusion can be performed only using Windows operating system. The rotating unit with laser rangefinder is fully integrated with ROS, using in-house developed open-source driver. Device is compact and it can be integrated with large number of platforms, including classic cars. That was achieved due to interesting mechanical solution: motor, encoder, slip ring, motor controller and other equipment is integrated in the rotating head. This way there is no space needed under device. Closed loop PI DC motor controller allows achieving low, steady rotation velocity. Additionally the device is equipped with special torque limiter. The proposed solution allows to safely operate in difficult environment. Presented sensor can be exposed to rapidly changing linear and rotation velocity, and scanning head can be blocked by obstacles (eg. vegetation). Large mass of rotating elements creates large momentum of inertia. This inertia momentum can produce large torque during platform's rapid movement. This moment is limited by torque limiter, so unit's gear-box cannot be damaged. For the sake of simplicity, the rotating unit with the laser range finder needs only power supply and one Ethernet connection for operation. Spherical camera requires FireWire connection. At minimal rotation speed, the system can produce about 800 000 points in a single scan. Mapping pipeline is as follows:

1. Teleoperation mapping using ROS(Robot Operating System) and described sensors. Data is only saved to PCD XML metamodel. Spherical camera stores the video stream in proprietary PGR format.
2. Data fusion using PCL(Point Cloud Library) and LadyBugSDK - as the result we obtain colored point cloud.
3. Data subsampling.
4. Registration using ICP(Iterative Closest Point) or NDT(Normal Distribution Transform[19]).
5. Data relaxation using ELCH(Explicit Loop Closing[20]) or LUM(Lu/Milios like SLAM[10]).
6. Export "ready to use" model using Cloud Compare.

7. Import model to production software (eg. *Autodesk 3DStudio Max*).

Fig. 1: Mobile-mapping system – mobile robot equipped with 3D laser measurement system and Ladybug 3 spherical camera.

III. Mobile Mapping Scenarios

A. *Scenario 1 – robotic data processing*

In this scenario we propose a 3D acquisition procedure in a ROS-based robotic system node. The node listens to topics, combining it with current transformation and saves it. Typical application is a 3D laser scanner with EKF filtered odometry data. Stored data is ready to use in the framework. Note, that at this state data can be visualized with Cloud Compare. Cloud Compare can be used for removing corrupted data (such as incomplete scans), or manual registration if localization systems results are poor.

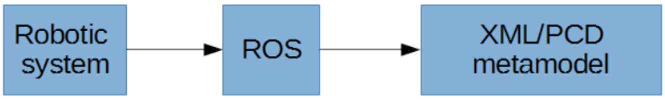


Fig. 2: First scenario of using proposed framework.

B. *Scenario 2*

ROS allows to record and playback so-called BAG files. Those files contain information (topics), which were transported in system. This allows using any BAG file in our framework.

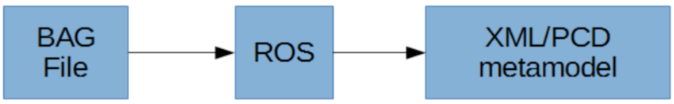


Fig. 3: Second scenario of using proposed framework.

C. *Scenario 3*

In the scenario data can be loaded from almost any dataset and any device. In that way data from geodetic scans, industrial scanning heads or airborne photogrammetry can be used. Typically data from few viewports are imported to Cloud Compare, manually aligned and registered. Core feature of the framework is a defined format of the data, which deals with multi-scan models. Model is a combination of XML structure, which keeps metadata (for eg. transformation matrices, timestamps) and PCD files, which holds point cloud data. Point clouds are in local sensor's coordinate systems and are modified only when it is essential (subsuming). Multiple XML files can be attached to the same point cloud data. Point clouds are kept in the PCD format. This concept allows executing multiple iterations of algorithms without copying data. XML also stores the data about last execution, thus results can be easily compared.

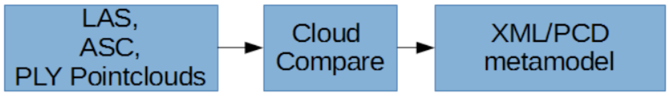


Fig. 4: Third scenario of using proposed framework.

IV. 3D Data Registration Framework Overview

Data registration is based on 6DSLAM algorithm shown in figure 5. It is composed of three components: data registration, loop detection and graph optimization. This algorithm was extended by ELCH method [18] to improve initial graph recalculation. PCL extends this algorithm by providing NDT data registration framework introduced in [19]. In this paper we propose complete 3D mapping framework integrating all 6DSLAM and extension components.

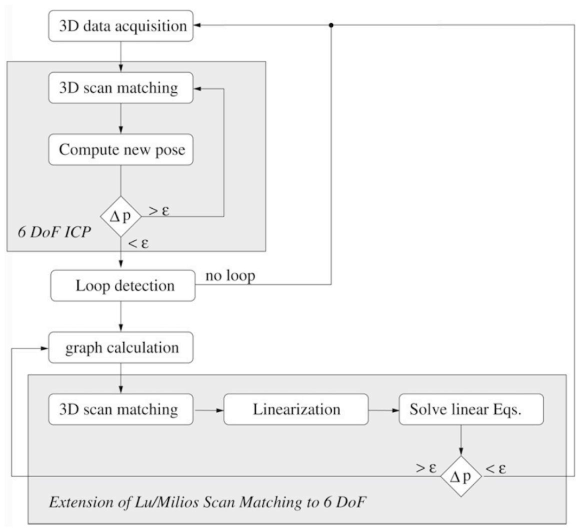


Fig. 5: 6DSLAM algorithm described in [10].

The 3D data registration framework is composed of following programs:

- *filtering* (reduction of noisy data),
- *subsampling* (result: equal density over 3D cloud of points),
- *icp* (data registration with Iterative Closest Point),
- *ndt* (data registration with Normal Distribution Transform),
- *elch* (explicit loop closing procedure),
- *lum* (graph optimisation after loop closing),

- *nn-overlap-plot* (visualisation of overlap between neighbouring scans),
- *icp-overlap-viewer* (visualisation of nearest neighbours between two scan),
- *elch-viewer* (visualisation of edge in the graph related with explicit loop closing),
- *lum-viewer* (visualisation of the graph being optimised),
- *model-viewer* (visualisation of 3D model),
- *compute normals* (calculation of normal vectors),
- *result-plot* (visualisation of algorithm results such as computation time, fitness-score etc...),
- *trajectory-viewer* (visualisation of the trajectories modified by 6DSLAM components).

V. DATASETS

We propose the new datasets for research and evaluation in 3D cloud processing (Fig. 6). The advantage of this dataset is high accuracy of the reference models, scanned with geodetic laser scanners. In the proposed framework large datasets are organized in hierarchical structure. Point cloud data is kept in PCD (Point Cloud Data) format, which is natural choice for PCL applications. Each point cloud is always in sensor's coordinate system, without any transformations. Transformations are kept in an XML file. The operations such as registration or loop closing in fact does not affect the data in PCD files, they only modify affine transforms accumulated in XML file. Algorithms that need to modify point cloud data, like subsampling and filtering, store the data in new location, therefore we do not loose any information during data processing. XML file contains information about scan localization, as reference path to XML file, model global transformation and every point clouds' local transformation. If dataset is generated directly from the data, which was captured by mobile mapping system, accumulated transforms will be robot's poses (odometry) during scanning process.

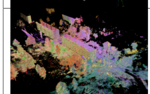
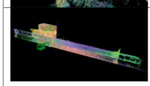
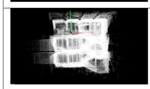
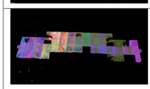
Image	Name	Scans	Sensor
	Indoor Outdoor	232	Rotating SICK LMS100
	Klomino Geodetic	18	ZF Imager 5010
	Stairs Geodetic	24	ZF Imager 5010
	Stairs Robot	41	Rotating SICK LMS100
	Underground Parking Geodetic	13	ZF Imager 5010
	Underground Parking Robotic	79	Rotating SICK LMS100

Fig. 6: Datasets available in [17].

VI. EXPERIMENT – FRAMEWORK DEMONSTRATION

Our first step was to evaluate the dataset using plot from figure 6. It shows nearest neighborhood overlap iteratively for each pair of scans. In this case the minimum value is above 20%, thus this dataset can be registered. Figure 7 shows all trajectories from all steps of registration: odo_s.xml (odometry subsampled data), odo_s_icp.xml (trajectory after iterative closest point procedure), odo_s_icp_2.xml (trajectory after second iterative closest point procedure with decreased radius of nearest neighborhood search), odo_s_icp_2_elch.xml (trajectory after explicit loop closing procedure), odo_s_icp_2_elch_lum.xml (trajectory after global relaxation with LUM method). After iterative closest point we perform explicit loop closing procedure [20] (fig. 8) that finds loop end closes by finding transformation between the scans and redistribution error over all scans in this loop. Finally we perform global map relaxation with LUM method [10]. For this reason our framework plots graph used for optimization (figure 9). Final 3D model is shown on figure 10. Finally our framework can be used for evaluation purpose, thus we provide accurate reference model obtained with terrestrial laser scanner Z+F IMAGER 5010 and registered with geodetic precision (fig. 11). Figure 12 shows histogram of errors measured by quantity of distances between points from reference model and result of proposed framework. The average error is 30cm, however the model is consistent. This confirms that the framework allows user to successfully generate 3D model from raw data, thus forming a complete pipeline. We are using this framework as a reference methodology for our research in 3D mapping applications. Datasets used in this experiment are available in http://lider.zms.imm.org.pl/downloads. We have also published the instructions how to perform this experiment. The complete framework presented in this paper is available here
https://github.com/LIDER-MSAS/data_registration_pcl.
The Cloud Compare plug in is available here:
https://github.com/michalpelka/qDataFramework.
The ROS driver for 3D unit [15] is available here https://github.com/mandalarobotics/m3dunit_install.

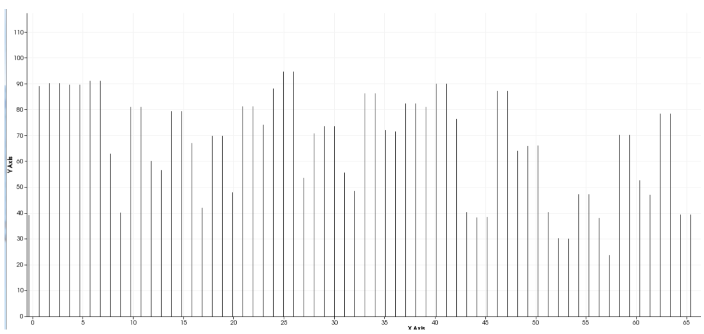


Fig. 6: Plot for evaluation of dataset. It shows nearest neighborhood overlap iteratively for each pair of scans.

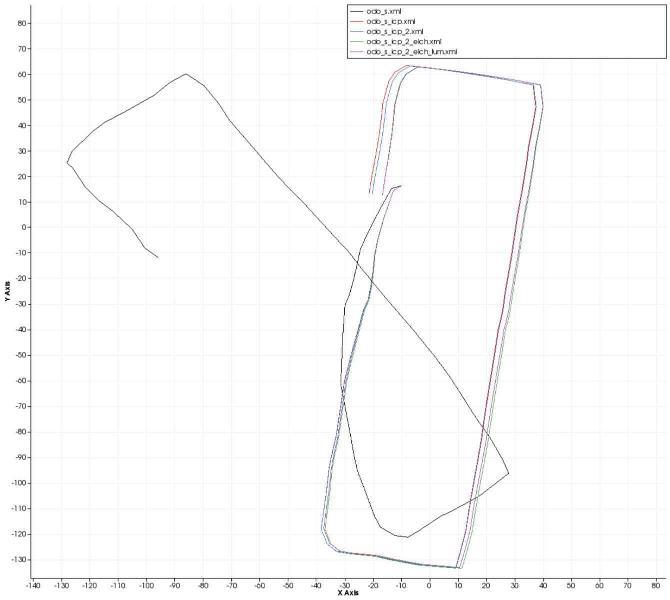


Fig. 7: Trajectories from all steps of registration: odo_s.xml (odometry subsampled data), odo_s_icp.xml (trajectory after iterative closest point procedure), odo_s_icp_2.xml (trajectory after second iterative closest point procedure with decreased radius of nearest neighborhood search), odo_s_icp_2_elch.xml (trajectory after explicit loop closing procedure), odo_s_icp_2_elch_lum.xml (trajectory after global relaxation with LUM method).

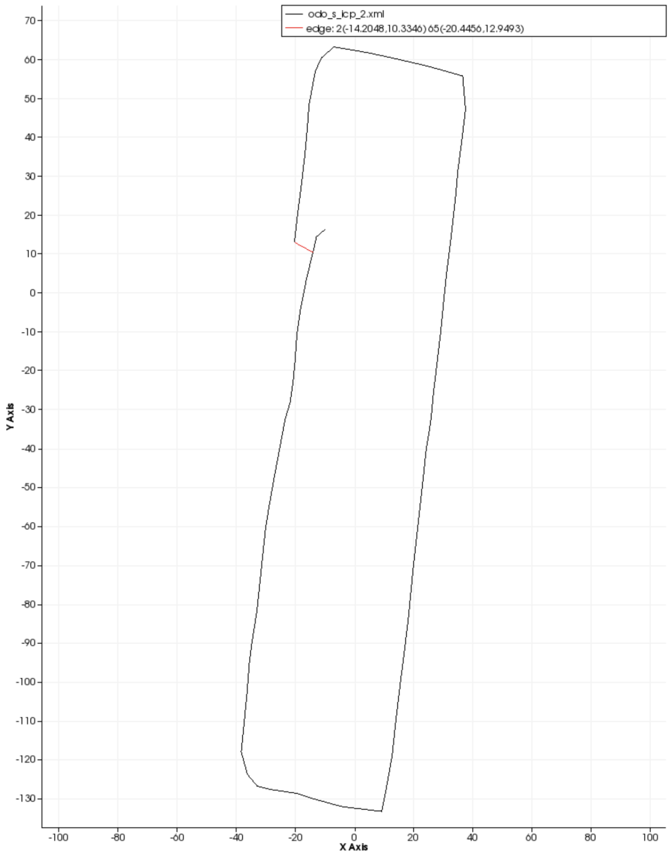


Fig. 8: Loop closing procedure[20] that finds loop end closes it by finding transformation between these scans and redistributing error over all scans from this loop.

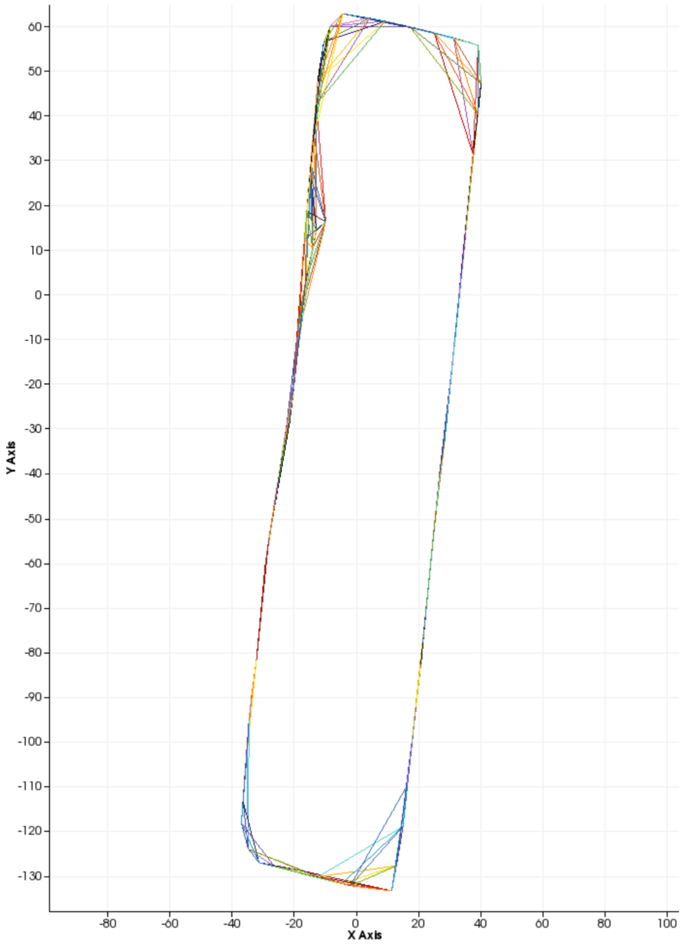


Fig 9: Graph for final map relaxation with LUM[10] procedure.

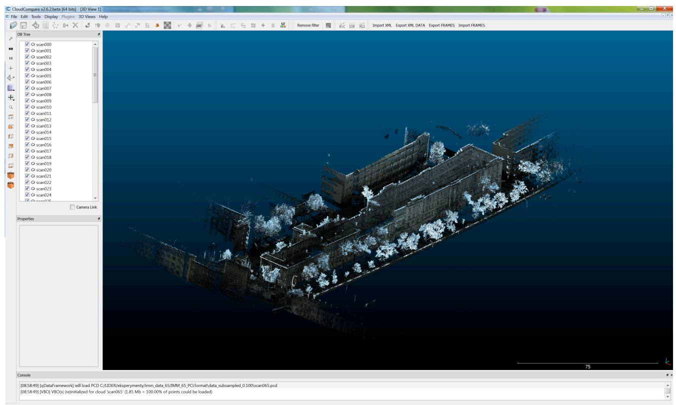


Fig 10: Final 3D model obtained with this experiment.

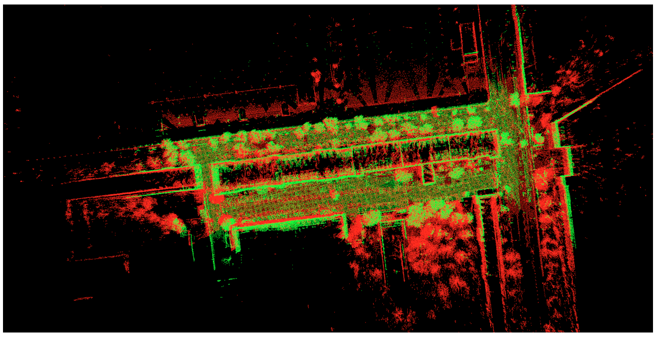


Fig 11: Comparison of reference model (red) and result of proposed framework (green).

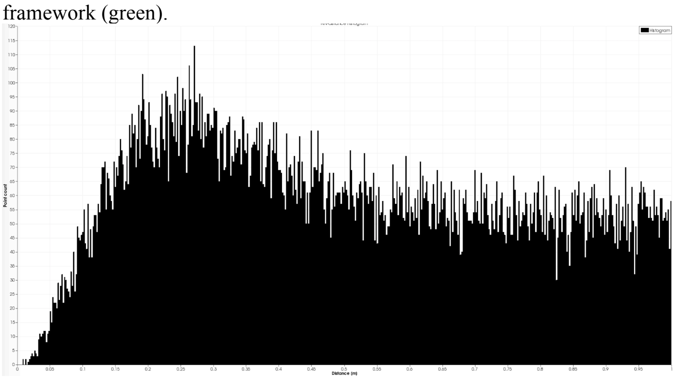


Fig 12: Histogram of errors measured by quantity of distances between points from reference model and result of proposed framework. The average error is 60cm.

VII. Conclusions

In this paper we have proposed an open source robotic 3D mapping framework with ROS (Robot Operating System), PCL (Point Cloud Library) and Cloud Compare. The system is capable of mapping a 3D location. The result of our research is available publicly in the form of open-source software and datasets with accurate reference 3D models. The framework is composed of tools for dataset pre-processing (filtering, subsampling), data registration (ICP, NDT), graph optimization (ELCH, LUM), tools for validation (comparison of 3D maps and trajectories), performance evaluation (plots of different outputs of algorithms). These tools are designed for 3D mapping pipeline. We introduce new dataset composed of several locations. The added value for robotics community is that each location is mapped with geodetic survey, thus the reference 3D models are highly accurate (typically less than 1cm). Our robots are equipped with commercially available 3D lasers and were used for 3D survey in those locations. Our contribution to robotic community is XML based data organization, implementation of metascan as an extension of pairwise registration and software tools for qualitative and quantitative evaluation of 3D models on each registration step. We believe, that this work can help robotic users in understanding critical issues concerning 3D mapping.

Acknowledgment

This work is done with the support of NCN (Polish National Centre of Science) project "Methodology of semantic models building based on mobile robot's observations" nr: DEC-2011/03/D/ST6/03175. This work is done with the support of NCBiR (Polish Centre for Research and Development) project: "Research of Mobile Spatial Assistance Sys- tem" Nr: LIDER/036/659/L-4/12/NCBR/2013.

References

[1] P. Skrzypczyński, Uncertainty models of the vision sensors in mobile robot positioning, International Journal of Applied Mathematics and Computer Science 15 (1) (2005) 73–88.

[2] P. Skrzypczyński, Simultaneous localization and mapping: A feature-based probabilistic approach, International Journal of Applied Mathematics and Computer Science 19 (4) (2009) 575–588.

[3] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, S. Thrun, 6D SLAM with an Application in Autonomous Mine Mapping, in: In Proceedings of the IEEE International Conference on Robotics and Automation, 2004, pp. 1998–2003.

[4] P. Pfaff, R. Triebel, W. Burgard, An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing, Int. J. Rob. Res. 26 (2) (2007) 217–230.

[5] P. J. Besl, N. D. McKay, A Method for Registration of 3-D Shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.

[6] J. Będkowski, A. Masłowski, G. de Cubber, Real time 3D localization and mapping for USAR robotic application, Industrial Robot 39 (5) (2012) 464–474.

[7] T. Rozen, K. Boryczko, W. Alda, GPU bucket sort algorithm with applications to nearest-neighbour search, WSCG 16 (1-3) (2008) 161– 167.

[8] D. Qiu, S. May, A. Nüchter, GPU-Accelerated Nearest Neighbor Search for 3D Registration, in: Proceedings of the 7th International Confer- ence on Computer Vision Systems, ICVS09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 194–203.

[9] J. Elseberg, D. Borrmann, A. Nüchter, Efficient processing of large 3D point clouds, in: 2011 XXIII International Symposium on Information, Communication and Automation Technologies (ICAT), 2011, pp. 1–7.

[10] O. Wulf, A. Nüchter, J. Hertzberg, B. Wagner, Benchmarking urban six-degree-of-freedom simultaneous localization and mapping, J. Field Robot. 25 (3) (2008) 148–163.

[11] R. B. Rusu, S. Cousins, 3d is here: Point cloud library (pcl), in: IEEE International Conference on Robotics and Automation (ICRA), 2011, IEEE, 2011, pp. 1–4.

[12] http://slam6d.sourceforge.net

[13] http://pointclouds.org

[14] http://www.ros.org

[15] http://mandalarobotics.com

[16] http://www.danielgm.net/cc/

[17] http://lider.zms.imm.org.pl/downloads

[18] J. Sprickerhof, A. Nüchter, K. Lingemann, J. Hertzberg, An Explicit Loop Closing Technique for 6D SLAM, in: I. Petrovic, A. J. Lilien- thal (Eds.), Proceedings of the 4th European Conference on Mobile Robots, ECMR 09, September 23-25, 2009, Mlini/Dubrovnik, Croatia, KoREMA, 2009, pp. 229–234.

[19] M. Magnusson, T. Duckett, A Comparison of 3D Registration Algo- rithms for Autonomous Underground Mining Vehicles, in: Proceedings of European Conference on Mobile Robots ECMR, 2005, pp. 86–91.

[20] Jochen Sprickerhof, Andreas Nüchter, Kai Lingemann, Joachim Hertzberg. An Explicit Loop Closing Technique for 6D SLAM, In Proceedings of the 4th European Conference on Mobile Robots (ECMR '09), Mlini/Dubrovnic, Croatia, September 2009