3D Map Building with Mobile Robots

Levente Tamas and Lucian Cosmin Goron

Abstract— **This paper presents a feature-based registration for 3D environments using mobile robots. The developed 3D laser scanner with custom hardware setup is able to scan both indoor and outdoor. For the map registration a nonlinear variant of the Iterative Closest Point (ICP) algorithm was used with initial alignment from the correspondences given by the features of the scenes. The initial alignment was determined using a set of key-points and the features of the keypoints in order to reduce the computational time and to ensure a robust estimation. Considering the increasing interest in 3D navigation for mobile robots, our aim was to use the created maps for both indoor and outdoor navigation purposes. Several maps were built by merging point clouds while our method was tested for a wide range of datasets including urban and office environments.**

I. INTRODUCTION

The demand for 3D perception and action for mobile robots increased rapidly in the recent years. There are several applications in which the representation of complex environments is essential including architecture, automotive, mining and piping systems inside factories [?], [?]. Off-line 3D data is required for applications such as architecture or factory design. These datasets can be conveniently acquired using 3D scans from multiple views. Other applications like autonomous navigation may require real-time data from the environment [?].

Several variants for the spatial perception of environments exist today including stereo cameras, laser scanners, time-of-flight cameras, ultrasonic rangers or the recently adopted structured light sensors [?]. Each of these sensors has its own limitation regarding the range, precision or robustness of measurements. Although computer vision techniques increased in performance, including stereo image processing [?], the data collected from laser range finders is more robust and texture independent.

Planar laser scanners are popular in the field of robotics due to their high speed and precision. They are mainly used for 2D mapping and navigation purposes, although by augmenting the two-axial motion with an additional degree of freedom, an accurate 3D scanner can be obtained [?]. Several variants may be used for the sweeping of the 2D laser, including the moving of the base platform during the scan [?] or by means of a mechanical actuator which is controlled independently from the vehicle displacement [?]. For applications in the field of navigation, the scans taken from different positions are useful only if registered, i.e. merged in a common coordinate frame in order to have a map of the surrounding environment. The registration can be interpreted also as an optimization problem of the spatial transformations among different overlapping scans [?].

The registration process is highly dependent on the characteristics of the measured data including noise, sparseness and robustness. Thus several techniques were developed for registration in different environments such as geometric feature-based registration for urban scenes [?] or raw point cloud registration for cluttered environments [?], [?]. The registration is also possible based on the position of the moving robot, but due to the uncertainty in the robot's localization the error accumulated during the displacement may corrupt the registration. A common approach is to use distinctive features for a set of points between consecutive scans for initial alignment and the ICP algorithm [?] for the fine-tuning of the registration. A similar version of this approach was also adopted for the paper at hand.

In this paper several types of data were tested for the registration problem with different key-points and feature descriptors. The main scope of the paper was to determine a suitable setup for the map registration benchmarking different combination of key-points and feature descriptors. In Section II the hardware and software details regarding the custom 3D laser scanner are presented. Section III describes the feature-based registration techniques used for the merging of scans and testing of different techniques. We conclude in Section IV and present our future research directions.

II. 3D LASER SCANER ON A MOBILE ROBOT

This section presents the design and construction details regarding the 3D laser scanner module mounted on a mobile robot platform. This module is based on a commercial Sick LMS200 2D laser for which an auxiliary mechanical part was constructed in order to earn a 3^{rd} degree of freedom. The actuated laser scanner is mounted on a *P3-AT* mobile robot and data acquisition is performed using the *ROS* [?] environment.

A. Actuated 2D Laser Range Finder

The key component of the 3D sensor is the 2D commercial laser scanner for which a custom rotary platform was designed. There are several possibilities to rotate the laser scanner, i.e. around the yaw, pitch or roll axis, thus achieving a yawing, pitching or rolling 3D sensor [?]. Each of these three setups has its own advantage and disadvantage. As for mobile robots the most common approach is the pitching scan, which was adopted for the current system. The mechanical design and prototype are presented in Figure 1. The design shown has two parts: one fixed containing the

Robotics Research Group, Technical University of Cluj-Napoca
71-73 Dorobantilor Street, 400609 Cluj-Napoca, Romania
{Levente.Tamas, Lucian.Goron}@aut.utcluj.ro

driving servo motor (left) and the rotation encoder (right); and the mobile rotary on which the Sick LMS200 is placed. The prototype was built using an iron frame both for the fixed and mobile part.

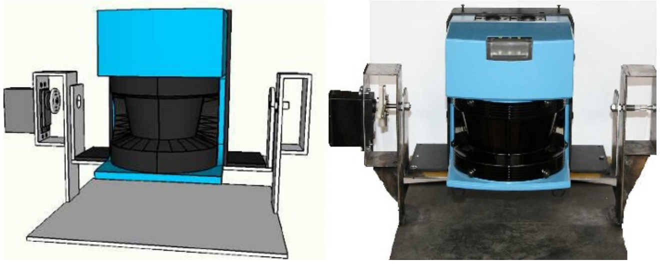


Fig. 1. The design and prototype of the actuated 3D sensor.

For the driving motor a Hitachi $12V$ servo motor was chosen with a minimum rotation step of $0.45°$, while for the rotation sensor a high resolution encoder was considered. The motor control and serial connection to the PC were solved using an AVR micro-controller based on the Cerebot2 electronic board. This type of board as well as the other mechanical and electrical components of the prototype are low cost products. The Sick LMS200 has a depth resolution of $1cm$ and an angular resolution of $0.25°$, $0.5°$, or $1°$ depending on the configuration. The scanning cone of the device can be set to either $100°$ or $180°$ depending on the actual needs, while the maximum range of readings is $80m$. The scanning time is around $15ms$ and additional time is required to send the data to the PC at 9600, 19200, 38400 or 500000 kb/s. Thus a complete 3D scan may require seconds depending on the actual configuration used for the scanning process.

For a scanner with pitching actuator the 3^{rd} dimension of a point is given from the pitch angle. The coordinates of one 3D point result from the distance to the surface, the yaw angle of the beam, and the pitch angle of the actuated mechanical part. Thus a scanned point can be represented as a tuple of the form $(\rho_i; \theta_i, \gamma_i)$ where ρ_i represents the depth information from the laser scanner and θ_i, γ_i the yaw and pitch measurements. The forward kinematic transformation taken as original coordinate system to the laser base link is given by:

$$p = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{pmatrix} \times \begin{pmatrix} \rho\cos\theta \\ \rho\sin\theta \\ 0 \end{pmatrix} \quad (1)$$

where p is a point in the Cartesian space with the coordinates x_n, y_n and z_n.

In (1) the displacement between the center of the robot and the 3D sensor was not taken into account. This can be introduced into the mathematical model by means of an additional translation term. Also the error induced by the misalignment between the rotation axis of the laser mirror and the pitching axis is not taken into account. This introduces a systematic error which can be detected by experimental identification and compensated by a constant term in (1). A more detailed discussion regarding the error budget can be found in [**?**].

B. Software Framework for the 3D Scanner

When dealing with a wide variety of experiments it is important to have a well designed software architecture. This can reduce substantially the development time by ensuring code reuse and robustness of software modules. First the general concepts are presented followed by the design phase, then the involved components and details of the integration stage.

Main idea for the design was to obtain a balance between the maintainability of the packages and the flexibility for code reuse. The adapted solution in this case was the Domain Driven Design (DDD) described in [**?**].

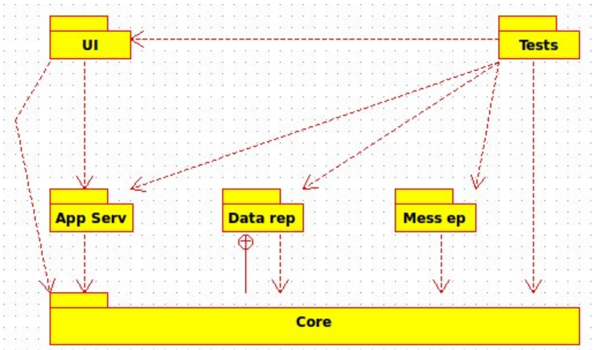


Fig. 2. Overview of the 3D scanner system.

The different layers and the relationship between them is presented in Figure 2. The domain layer contains the core of the package which for the map registration is the registration algorithm itself (e.g. non-linear ICP). It implements the interfaces to the application layer, data repositories and message endpoints. The main role of the application service layers is the delegation and execution of tasks. It is a middle layer between the core and the user interface without the knowledge about the data representation and the communication details within the package. The communication with external data sources is done via the data repository layer, while the message endpoints specify the communication internally and externally to the package. All layers have their own test unit defined. The adopted solution for unit tests is based on the test doubles, i.e. stub objects, which act as the other packages during the test. This may be useful for larger projects to separate the testing phase for different layers [**?**].

Several components were developed for the experiments in different computer languages like C++, Python or Matlab, thus it was important to have cross platform libraries which could be merged into a single application. The main programming language was C++ on a Linux platform within the Robotic Operation System (ROS) [**?**] proved to be a good choice for transparently integrating the different hardware and software modules.

III. MAP REGISTRATION

Several range scans are necessary in order to build a 3D elevation map of the environment. To use these scans as a coherent dataset, they have to be unified in a common coordinate frame. Unless the position and orientation of the mapping robot is accurately known, the range scan registration needs to be done using specialized algorithms. Since in our case the robot position could not be determined with a sufficient accuracy between the measurement steps, the registration algorithms were employed for creating the elevation maps.

A. 3D Data Acquisition

The laser scanner presented in Figure 1 was mounted on a P3-AT mobile robot in order to perform both indoor and outdoor scans as it can be seen in Figure 3. The scan area was of $180°(h)$ x $100°(v)$ with a horizontal resolution of 361 and a vertical one with 200 steps. This configuration ensured an optimal resolution and data density in the point cloud for further data processing. The scanning of environment with the mobile robot was performed in a stop-scan-go fashion, a single scan taking up to 60 seconds depending on the used configuration. All the measured data was integrated in the ROS environment where each logging was timestamped for an easier off-line processing.

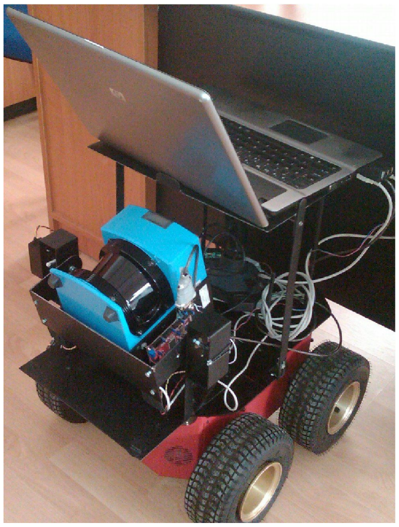


Fig. 3. The 3D scanner on the mobile robot platform.

B. ICP-Based Registration

The registration problem can also be viewed as the optimization of a cost function describing the quality of alignment between different scans. The algorithm determines the rigid transformation which minimizes this cost function [?]. The type of algorithm applied for the frame alignment strongly depends on the measured dataset type. For the 3D laser scans the Iterative Closest Point (ICP) and derivatives are popular in the field of robotics [?], [?], [?]. The ICP computes the rigid transformation which minimizes the distance among two point sets by associating a point from one frame to the closed point in the target frame. The transformation between two independently acquired sets of 3D points consists of two components, a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. Correspondence points are iteratively searched from the model set of points M (with $|M| = N_m$) in the dataset D (with $|D| = N_d$). In case of a valid correspondence we need the transformations $\mathbf{R}$ and $\mathbf{t}$ which minimize the distance between the two points as follows:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})||^2 \qquad (2)$$

where $w_{i,j}$ is assigned 1 if a valid correspondence is found between the ith point from M denoted with $\mathbf{m}_i$ and the jth point from D denoted with $\mathbf{d}_j$.

Different variants were developed in order to increase the robustness and the performance of the algorithm especially for computing the rotational transformation term, which introduces a non-linear term in the minimization problem. A comprehensive overview and qualitative evaluation of different approaches for the registration problem can be found in [?].

A common approach for boosting the ICP robustness is the augmentation of the points with additional features like point color, geometric features or point histograms [?]. This transposes the optimization problem in a higher order dimensional space search. These features are usually computed only for a certain subset of interest points from the original point cloud, i.e. keypoints in order to reduce the computational effort and enhance robustness. The use of keypoints is to enable the efficient comparison between different data regions. Our approach for the data registration is based on the correspondence estimation for the extracted keypoint features.

C. 3D Keypoints and Descriptors

There are several possibilities for extracting interest points and descriptors from 2D images including the popular SIFT (Scale Invariant Feature Transform) [?] or the SURF (Speeded Up Robust Features) [?] features. Unfortunately, these rely on local gradients from a unique orientation and therefore are not directly applicable for our approach with 3D data, however some concepts may be inherited from the 2D domain.

In this paper the Normal Aligned Radial Feature (NARF) [?] keypoints were adopted for the extraction of interest points from range images. This type of keypoint takes into account the information about the boarders and surfaces, ensures the detection from different perspectives and the stability for the descriptor computation. The most important parameter for the NARF extraction is the support size, i.e. the diameter of the sphere in which the interest point characteristics are determined [?]. In our case several values for this parameter were tested in order to gain a sufficient number of keypoints for different types of datasets.

After the selection of keypoints, the specific properties are determined, meaning the descriptors for the set of extracted keypoints. The role of the descriptors is to efficiently compare for discrimination between two selected points. There are several approaches for the descriptors, a part of them being invariant to the rotation around the normal, like in the case of the NARF descriptor [?] or even complete 3D orientation invariant such as the Fast Point Feature Histogram (FPFH) [?].

For our approach we used the optimized version of the FPFH in order to augment the three dimensional space with pose-invariant local features and also tested the NARF descriptors with Manhattan metrics for the same set of keypoints. To compare the two set of descriptors, the runtime (T) and the initial alignment fitness score (S) was computed for indoor (Id) and outdoor (Od) datasets. The result of the comparison is summarized in the Table I.

TABLE I
FEATURE DESCRIPTOR COMPARISON

$Dataset$	T_{NARF}	T_{FPFH}	S_{NARF}	S_{FPFH}
$Id_{cluttered}$	0.19	45	0.071	0.032
Id_{plane}	0.12	12	0.094	0.057
Od	0.11	26	0.083	0.044

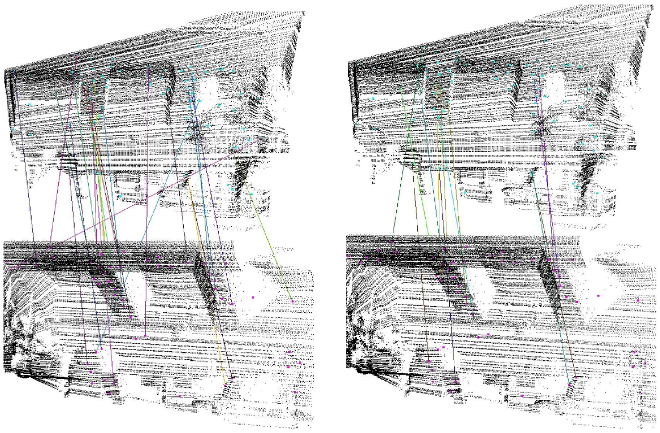
The tests were performed on datasets containing around $10K$ points for which the extracted number of keypoints was in the magnitude of $0.1K$. For computing the runtime the average values were considered for 10 consecutive runs on an Intel Pentium 4 single core laptop running Ubuntu Linux. Although the run-time of the proposed algorithm is higher than some custom scenario based approaches like the one presented in the work [?], the degree of generality of the current approach is higher.

As observed, NARF descriptors are computed with several orders of magnitude faster than FPFH descriptors, but the latter approach is more robust in terms of estimating correspondences. This would be also the case for scenes which present less clutter or variation, thus having less discriminative features, where the FPFH features ensured a better correspondence between points.

D. Correspondence Estimation

The next step after determining the keypoints and the descriptors is the estimation of correspondences between the two sets of keypoints with descriptors. There are several methods for the correspondence estimation, such as one-to-one, back and forth or sample consensus based [?].

In our approach the correspondence estimation was performed based on the geometric constrains of the selected points. Thus the nearest point in the high dimensional descriptor space was searched by using a *kd-tree* for enhancing the search speed [?]. Unfortunately, the brute force search does not ensure a coherent result for the correspondence estimation problem having in many cases a large number of false positives. For improving the estimation results, the filtering based on sample consensus was adopted. This ensures that after performing the one-to-one search for descriptors, only those correspondences are kept which satisfy a geometrical transformation constrain.



(a) (b)

Fig. 4. Initial correspondences (a) and filtered correspondences (b).

The comparison of the unfiltered and filtered set of correspondences is shown in Figure 4 on an indoor dataset. This dataset contains two scenes, the original one and the one rotated with 45°. As it can be observed, the initial, unfiltered set of correspondences contains a large number of false positives, which are eliminated, yielding a more consistent estimation. The number of final correspondences depend on the parameters used as a threshold for the sample consensus rejection.

The complete ICP-based algorithm can be found in the works [?], [?], therefore only a short overview is given, with emphasis on the additional descriptor information for the points. The ICP with initial alignment is described in Algorithm 1. It has two input point clouds, P_s for the source and P_t for the target. Step 1 and 2 extract the FPFH of the source and target clouds (these two steps can be substituted with arbitrary point cloud feature search), while in Step 3 the initial alignment t^* is determined after the correspondence filtering. In the while statement in each iteration a set of associations A_d is taken for which the best transformation is determined. The loop exit conditions are related to the error variation or to the maximum number of iterations both specified as tuning parameters for the algorithm. Finally, the algorithm returns the computed transformation between the two datasets. Further details regarding the implementation of the ICP with initial alignment based on sample consensus can be found in [?].

Algorithm 1 ICP with initial alignment

```
Require: $P_s$, $P_t$
1: $F_s$ = ComputeFPFH($P_s$);
2: $F_t$ = ComputeFPFH($P_t$);
3: $(t^*, A_f) = InitialAligment(F_s, F_t)$;
4: while $(error_{diff} < \epsilon)$ $or$ $(iter < iter_{max})$ do
5:     $A_d = getClosestPoints(t^*, P_s, P_t)$;
6:     $t^* = argmin\left(\frac{1}{|A_d|}\sum_{j \in A_d} w_j \left|t(p_s) - p_t\right|^2\right)$;
7: end while
8: return $t^*$
```

Based on the discriminative power of the descriptors, the initial alignment for the ICP can be solved using a sample consensus based approach for the correspondence estimation [?]. Further on, the initial alignment ensures a faster convergence for the ICP, with less possibilities to get into a local minimum at the optimization phase. This is important from the multiple scan merging perspective.

The aligned map for the indoor environment is presented in Figure 5, while results for the outdoor registration are shown in Figure 6 and Figure 7. In both cases the registration was performed using a pair alignment approach and the FPFH descriptors for the computed NARF keypoints. The initial alignment of the scans was performed based on the filtered correspondences of the FPFH descriptors. This alignment was then used for the ICP refinement, computed on the last pair of data in the alignment loop.

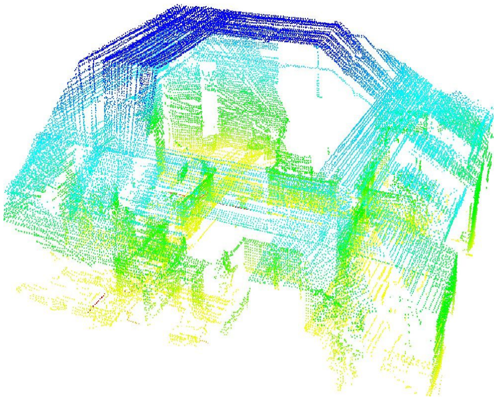


Fig. 5. Example of registered indoor map.

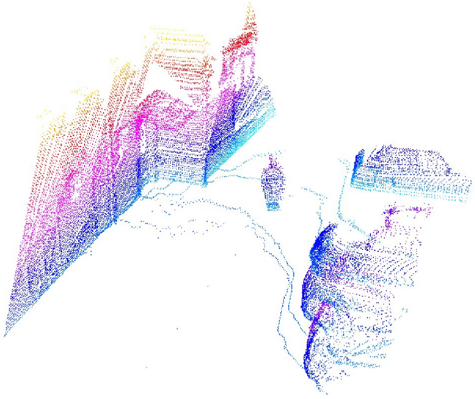


Fig. 6. An outdoor environment map with human obstacle.

For both scenarios the error convergence of the ICP algorithm was monotonically decreasing, a suitable registration error was achieved in less than 100 iterations. This scenario was obtained by considering the maximum distance between two neighbor points to be less than $1m$.

IV. CONCLUSIONS

This paper presents a custom 3D laser finder for indoor and outdoor environment mapping using a mobile robot.

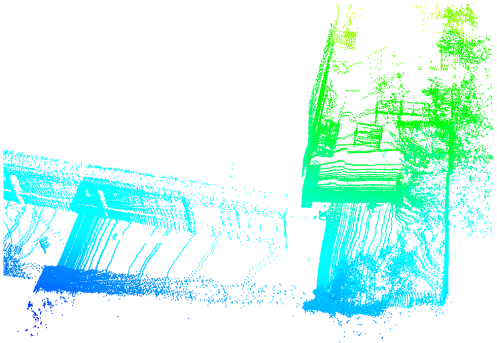


Fig. 7. An large scale outdoor environment map.

We built environment maps using a pairwise registration of 3D point clouds based on an enhanced ICP approach using different keypoints and local descriptors for the keypoints. The correspondences obtained are filtered in order to obtain a reliable initial alignment for the initialization of the ICP registration phase.

As future extensions, we plan to combine the current approach from the laser scanner information with data from stereo camera and from projected light sensor. For the registration part, we propose to implement the global optimization within a Simultaneous Localization And Mapping (SLAM) application.

ACKNOWLEDGMENTS

This paper was supported by the project ”Develop and support multidisciplinary postdoctoral programs in primordial technical areas of national strategy of the research - development - innovation” 4D-POSTDOC, contract nr. POSDRU/89/1.5/S/52603, and by the project ”Doctoral studies in engineering sciences for developing the knowledge based society - SIDOC” contract no. POSDRU/88/1.5/S/60078, both projects co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.