



Low-cost 3D Laser Design and Evaluation with Mapping Techniques Review

L. Bauersfeld, Guillaume Ducard

► To cite this version:

L. Bauersfeld, Guillaume Ducard. Low-cost 3D Laser Design and Evaluation with Mapping Techniques Review. 2019 IEEE Sensors Applications Symposium (SAS), Mar 2019, Sophia Antipolis, France. pp.1-6, 10.1109/SAS.2019.8706006 . hal-02143353

HAL Id: hal-02143353

<https://hal.science/hal-02143353>

Submitted on 16 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Low-cost 3D Laser Design and Evaluation with Mapping Techniques Review

L. Bauersfeld, G. Ducard

Abstract—This paper presents a design of a low cost 3D laser scanner. It was developed by adding an additional axis of rotation to a planar (2D) laser scanner. By tilting the laser scanner with respect to the additional axis of rotation, a more even scan point distribution can be achieved. This paper provides an analysis of the 3D scanning performance that can be expected from such a configuration. Then a review of methods that could be appropriate for 3D mapping with such a low-cost 3D laser scanner design is provided. Experiments show that this setup can be used onboard small vehicles to perform simultaneous localization and mapping in three dimensions.

Index Terms—Laser scanner, SLAM, ICP-alignment

I. INTRODUCTION

Over the last couple of years, small autonomous ground and aerial vehicles have become incredibly popular. For autonomous navigation, those vehicles rely almost completely on (differential) GPS, inertial sensors and sometimes a camera. While GPS allows for a very precise localisation in a global reference frame, it does not provide information on the environment. Cameras, on the other hand, provide detailed information on the surrounding, however, vision-based localization algorithms are computationally demanding. This usually requires powerful computers, which can be deployed as a ground station, but which are rather unsuited aboard a moving autonomous vehicle.

As an alternative to sense the environment, laser scanners have become popular in academia and industry. Laser scanners directly provide accurate data on the depth of a scene and are hence ideally suited for autonomous navigation. Scanners that only scan in a plane are called 2D laser scanners and a number of affordable and light-weight devices are available. However, often a two-dimensional representation of the surrounding is not enough for navigation and obstacle avoidance in 3D-space making a 3D scanner a necessity. The commercially available systems are heavy (*e.g.* velodyne), expensive (*e.g.* Hukoyo) and require much energy.

This paper presents a low-cost and light-weight 3D laser scanner that was developed by adding an additional axis of rotation to a 2D scanner. First, the mechanical design is discussed. Second, an analysis of the system performance based on simulations is provided. Third, an overview of approaches for simultaneous localization and mapping (SLAM) is given. Last, experimental results with a ground vehicle are shown.

L. Bauersfeld is with Institute for Dynamic Systems and Control, ETH Zurich, Switzerland, leonardb@ethz.ch

G. Ducard is with I3S, University Cote d'Azur, CNRS, Sophia Antipolis, France, ducard@i3s.unice.fr

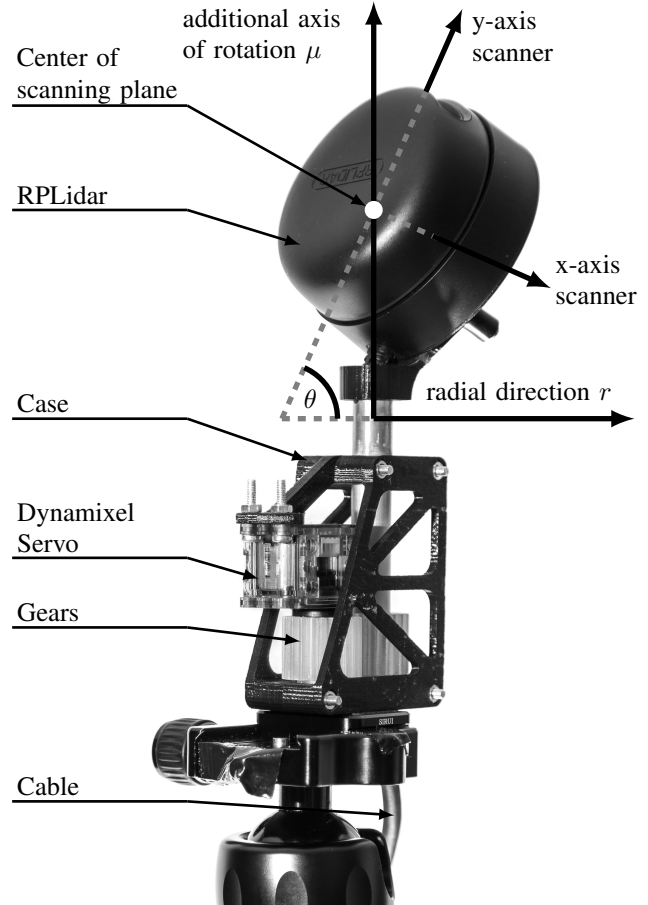


Fig. 1. Mechanical design of the 3D laser scanner. The upward direction is defined along the additional axis of rotation (referred to as μ -axis) and the scanning plane of the 2D laser scanner is spanned by the x- and y-axis. This coordinate system will be used throughout the paper.

II. DESIGN OF THE SCANNER

In Fig. 1, the design of the developed 3D laser scanner is shown. To turn the scanner around the additional axis of rotation μ , a servo is used. The laser scanner is mounted with an inclination angle θ with respect to the horizontal direction on a hollow aluminum axis, which is connected to a hollow gear. This allows for optimal cable routing.

A. Hardware

Due to its small form factor and low price, the RPLidar A2M4 has been chosen as a 2D laser scanner. It has a

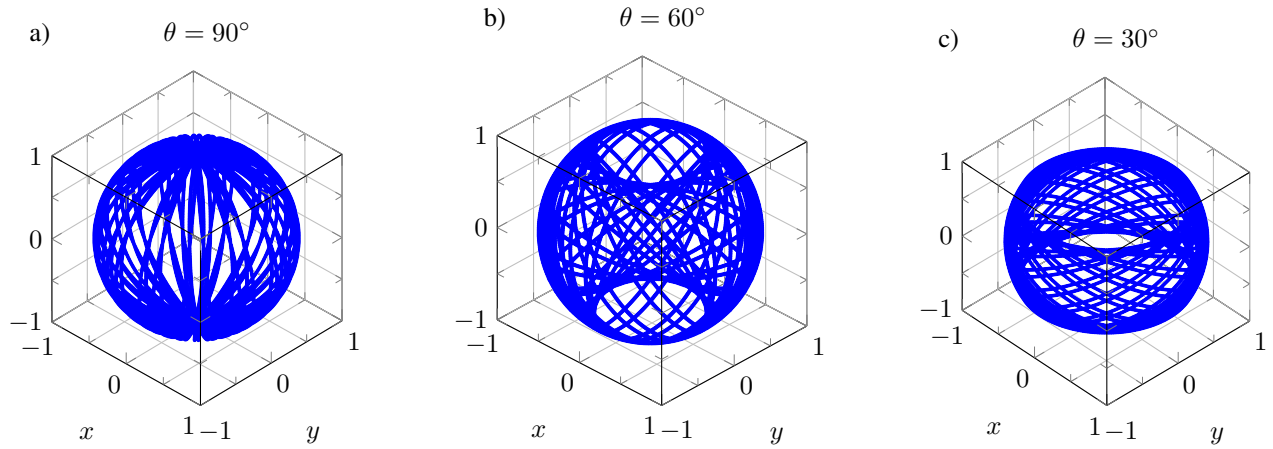


Fig. 2. Comparison of the scan point distribution when the inclination angle θ is changed. The axis μ is aligned with the z-direction of the 3D plots above. The simulation of the point distribution ignores any obstruction of the view due to the mechanical design.

range of 6 m, a resolution of 1° and a frequency of 10 Hz. While this is good for planar scanning, it will result in a rather low resolution, when turned into a 3D scanner. A *Dynamixel XL320* servo is used to actuate the additional axis of rotation. This servo is particularly well suited, because it is a continuous rotation servo that measures its own position. A typical operating speed of the servo is 0.5 Hz. The 3D scanner then scans the environment in 3D with 1 Hz. The factor of two is because the scanning plane passes each point twice per revolution of the servo. Both the scanner and the servo can be easily interfaced with the Robot Operating System (ROS).

B. Scanner Inclination

Ideally, a 3D laser scanner has an even point distribution, meaning that an equal amount of scanning points lies in each spherical sector. Only then, the resolution of the 3D scanner is independent of the object location. Since this is not achievable in practice, one tries to get a high point-density in the primary plane of motion, which is often horizontal.

Case 1, $\theta = 90^\circ$: If the axis μ lies in the scanning plane (*i.e.* $\theta = 90^\circ$), the point density will be very high along μ and very low in the radial direction as shown in Fig. 2 case a). Here the axis μ is pointing upwards. Eventhough this setup allows to scan the full sphere, this distribution is not ideal. The highest resolution is achieved directly above and below the scanner. The view ‘down’ is obstructed by the servo and the vehicle, resulting in a large number of scan points to be filtered out. More importantly, most vehicles move primarily horizontally (x-y plane in Fig. 2 case a)), but there the point density is the lowest.

Case 2, $\theta < 90^\circ$: By tilting the scanner by an angle ($\theta < 90^\circ$), its scanning plane does not contain the axis of rotation anymore which changes the scan point distribution as shown in Fig. 2 case b) and c). The 3D scanner has now a limited field of view ($\pm\theta$ degrees latitude) but an increased scan resolution in the horizontal plane. The smaller the inclination, the higher the scan resolution in the x-y plane gets. A setup like shown Fig. 2 b) is suited for autonomous vehicles since walls and

other obstacles are well resolved in the scan. Even smaller inclination angles (e.g. 5°) are possible to increase the scan

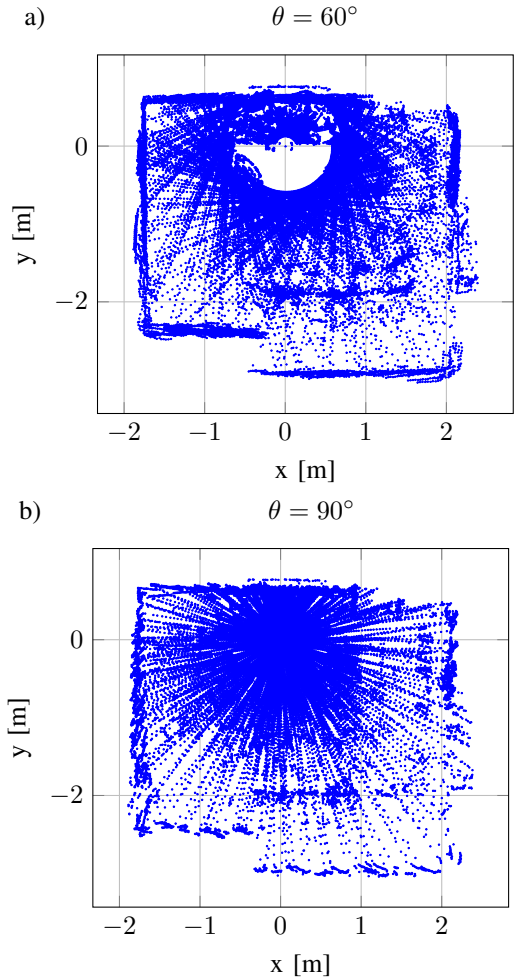


Fig. 3. Comparison of two real pointclouds representing the same object. Pointcloud a) was obtained using a tilted scanner ($\theta = 60^\circ$) and b) with no tilt ($\theta = 90^\circ$), but both consist of the same number of points. The double cone of unscanned space and the higher point density along the walls are clearly visible in image a) compared to b).

resolution close to the $x - y$ plane but such a device is not a true 3D scanner anymore.

When looking at a real-world scene, the advantage of a tilted scanner becomes clearly visible. Figure 3 shows two scans with different inclination angles of the same room. When the inclination is set to 90° (Fig. 2 b)), the walls are not resolved well since most of the points fall on the ceiling. With the tilted scanner, the important features of the scene (*i.e.* obstacles) are scanned with a higher resolution and thus can be recognized better.

The inclination θ should generally be chosen as small as possible to increase the resolution in the $x - y$ plane. The smallest inclination is given by the largest acceptable cone of unscanned space. Therefore, the optimal choice of the scanner tilt depends on the problem at hand.

III. LOCALIZATION AND MAPPING - REVIEW

An autonomous vehicle must be able to locate itself in an unknown environment and incrementally construct a map of it, without any external aids. To do so, multiple steps are necessary as shown in Fig. 4. This section briefly reviews common approaches for each step and details the feature detection and matching process, since this is strongly influenced by the resolution of the data and hence the choice of the laser scanner is important.

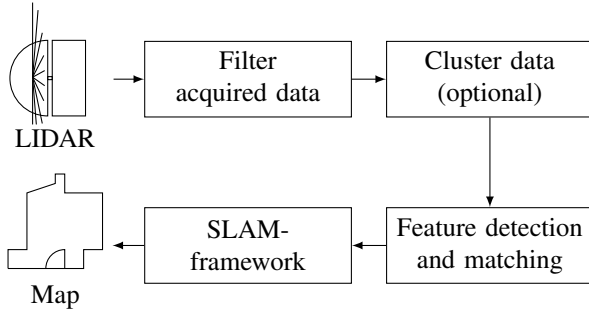


Fig. 4. Steps that any vehicle with autonomous navigation needs to perform to get from the laser scan to the map. First the data is preprocessed (filtered and clustered) and then the laser scans obtained at different times need to be matched. Last, the map is updated by the SLAM-framework.

A. Filtering and Clustering

In this first step, the incoming raw data from the laser scanner is filtered and then output. Whenever a physical quantity is measured, there is measurement error. The noise of laser scanners typically consists of two types: 1) there is a normally distributed measurement noise and 2) there is a significant number of outlier points, where a completely ‘wrong’ value was measured. This noise-type is often referred to as salt-and-pepper noise and it can be removed with a simple threshold or with a non-linear filter. The computationally lightweight thresholding technique can be used, when the scanner produces a salt-and-pepper noise with values out of the specified scanner range. Otherwise, a non-linear filter needs to be used. Linear filters have a bad performance when used to

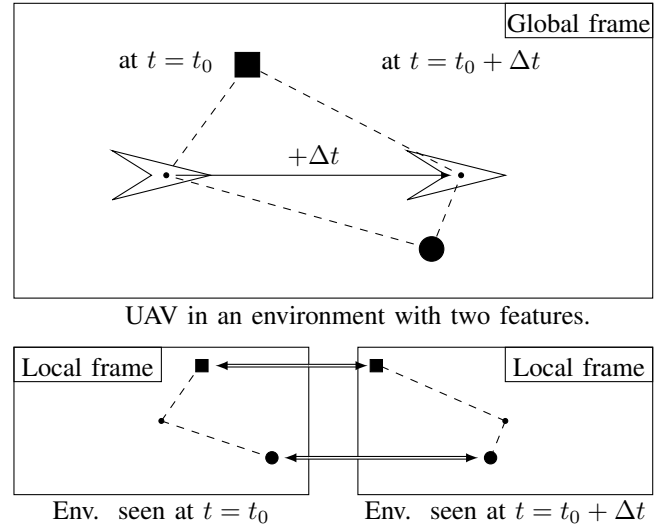


Fig. 5. The top image shows the situation in a global, stationary reference frame. The two pictures at the bottom show the location of the features in the UAVs local frame. To determine its position the UAV needs to establish correspondances between physical objects over time.

remove salt-and-pepper noise, because they cannot ignore the wrong information from outlier points. The most common non-linear filter is the median filter or one of the derivatives, *e.g.* the adaptive median filter [1]. An alternative to the median filter is the nearest-neighborhood smoothing [2], which preserves corners better.

Clustering or segmentation is the process of grouping points together that are similar according to certain criteria. This step is optional, since not all feature detection and matching techniques profit from segmented data. The points are assigned labels indicating their cluster affiliation. In the context of 3D pointclouds, clustering is performed on a per-scan basis with respect to whether points belong to the same physical surface / object. For more information, the reader is referred to “Clustering techniques: The user’s dilemma” [3].

B. Feature Detection and Matching

In order to perform simultaneous localization and mapping, the pose of the vehicle must be known with respect to fixed, physical objects as shown in Fig. 5. This relative position can be estimated by finding out which points – in two different laser scans obtained at different times and positions – correspond to the same physical object. One can imagine this as aligning the current laser scan with a map of the environment by applying a proper rigid transformation to it. The map is iteratively built and stores information about the environment obtained from the previous scans, *i.e.* the map is often the collection of all past scans.

With the newly developed 3D scanner, the scan alignment is a challenge. It has a low range and a low and uneven resolution. To align laser scans, either all scan points are matched (brute-force approach) or only certain keypoint points (feature-based approach) are considered. A keypoint or a feature is a very distinctive point (and the region around it) of a

physical object, *e.g.* a corner. To identify such keypoints, so-called detectors are used. Once the keypoints are identified, a descriptor is applied to calculate the feature vectors. The vectors are a mathematical and compact representation of what makes the selected point distinctive. Feature matching is about finding out which keypoints correspond to the same physical object. It is done by comparing the feature vectors.

1) *Feature-based Approach:* With a feature-based approach, the feature points need to be identified first. The most popular detector in two- and three- dimensional images is the *Harris corner and edge detector* proposed by C. Harris and M. Stephens in 1988 [4]. It was originally meant only for 2D images, but extensions to 3D data exist. [5]

1.1) *Detectors:* Figure 6 shows how the original detector works. Square A marks an area that is not distinctive by any means, since moving it slightly to the left or right, top or bottom, does not change anything. Area B is an edge, and therefore only distinctive in the x-direction. Nothing changes if B is moved in the y-direction. Only area C is distinctive, since any slight movement in any direction causes a drastic change. Thus C is a good feature. This means that a point is a *keypoint*, if any shift of the filter window causes its content to change significantly. The above can be expressed in a mathematical way:

Performing a Taylor expansion around the center point (x, y) of the filter window gives an approximation of the gradients I_x and I_y in the x and y direction. Let $I(i, j)$ denote the intensity value of the pixel in the i -th column and j -th row and pixel size Δ :

$$I_x = \frac{I(i, j+1) - I(i, j-1)}{2\Delta} \quad (1)$$

$$I_y = \frac{I(i+1, j) - I(i-1, j)}{2\Delta} \quad (2)$$

When this is repeated for all possible shifts (u, v) of the filter window, the matrix M can be calculated:

$$M = \begin{bmatrix} \sum_{u,v} I_x^2 & \sum_{u,v} I_x I_y \\ \sum_{u,v} I_x I_y & \sum_{u,v} I_y^2 \end{bmatrix} \quad (3)$$

Harris and Stephens [4] then define a *corner-response function* R with a tunable parameter k . The corner response function is large and positive for corners and large but negative for edges. Otherwise it is small.

$$R = \text{Det}(M) - k\text{Tr}(M)^2 \quad (4)$$

A possible extension to three dimensions, which is used in the Point Cloud Library is described in [5]. The image gradient (in 2D) is replaced with the surface normals which have the same meaning in 3D. They are estimated over a neighborhood of tunable size. The corner response function must also be replaced. This can be done differently, but one approach presented in [5] is the Lowe-method. There the corner-response function R is given by:

$$R = \frac{\text{Det}(\text{Cov}(x, y, z))}{\text{Tr}(\text{Cov}(x, y, z))^2} \quad (5)$$

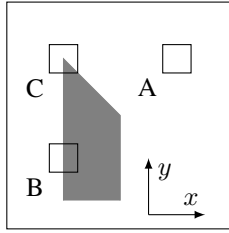


Fig. 6. Schematic illustrating the Harris Detector

The Harris Detector is known to perform well for a wide input range of scenes. For specific data (*e.g.* only face recognition) often more specific detectors perform better.

Next to the Harris Detector, there is a second very prominent detector that can be used for any scene. The SIFT detector was proposed in 2004 by David Lowe [6] for 2D images. Flint et al. presented an extension to three dimensions in 2007 [7]. Both the 2D and 3D version are widely used for non realtime applications, since the SIFT detector is very accurate but computationally extremely demanding. In a comparison between the Harris3D detector and the SIFT3D detector, it has been found that the Harris3D detector is about ten times faster than the SIFT3D detector while providing comparable accuracy [5].

1.2) *Descriptors:* The keypoint detector returns a set of highly distinctive points for each scan analyzed. Now the task is to find out, which keypoints correspond to the same location on a physical object. This is done by using a descriptor to compute the feature vectors of the keypoint and its surrounding. Those feature vectors are then typically compared in a nearest-neighbor fashion. One of the very first true 3D descriptors proposed (*i.e.* not just an extension of a 2D descriptor to 3D), is the Johnson Spin Image [8] descriptor. First, a point centered, partial cylindrical coordinate system is defined as shown in Fig. 7. Let the z -axis be parallel to the estimated surface normal \vec{n} and passing through the considered point p_0 . Now a radial coordinate r can be introduced as the perpendicular distance of a point p to the z -axis. A second coordinate h can be defined as the distance of p to a plane with normal vector \vec{n} passing through p_0 . The elevation is h . The angular coordinate is not introduced, as it can not be defined robustly.

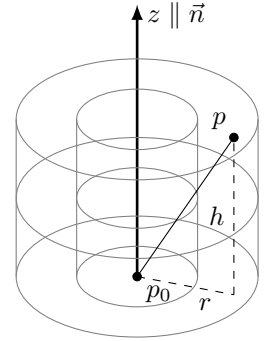


Fig. 7. The local coordinate system is used by the Spin-Image descriptor.

Then, the cylinder is split into multiple sections (called bins) radially and vertically. In a next step, every point in the support region is considered and it is calculated into which bin it falls. Finally one can write all bin values into the feature vector in a predefined order. The Spin Image descriptor is invariant to rotation, because the feature vector does not change, if the input image is rotated. This is because the azimuthal coordinate is omitted in the coordinate frame constructed. It becomes very clear, when thinking about the descriptor in a graphical way: it does spin the image around the surface normal at the considered point p_0 and thus reduces the number of degrees of freedom by one.

This descriptor is suited for low-resolution data, because only the surface normal needs to be estimated. Compared to techniques like the principal component analysis (PCA) this requires less dense data.

Other feature descriptors like the *Spherical Spin Images* [9] or the *3D Div* [10] exist. They are, however, less suitable for low resolution scan data or computationally expensive. When

TABLE I
COMPARISON OF DESCRIPTORS FOR 3D LASER SCAN MATCHING.

Descriptor	Sensibility to keypoint quality	Sensibility to scan resolution	Computational complexity
SIFT 3D	low	low	high
Spin Img	low	medium	low
Sph. Spin Img	low	high	medium
3D-Div	high	medium	low
ICP	none	low	high

the feature detection and matching approach is used, scan alignment is done by considering only the matched keypoints. Therefore, this approach is fast and possibly accurate at the same time.

2) *Non-feature Based Approach*: A non-feature based brute force approach has no need for a detector or a descriptor but directly aligns a laser scan with a given reference. The most common technique is the *iterative closest point* (ICP) algorithm. First, it looks at all points in the new scan and searches for the nearest neighbors in the reference. Then, the new scan is rigidly transformed such that the sum of squared distances between all nearest neighbors is minimized. Both steps are performed multiple times and hence the alignment is refined iteratively [11].

Because ICP considers all points and not only the special keypoints, it is robust even with a low-resolution scanner. However, it is subject to local minima and can converge to the wrong solution when the initial guess is bad (*i.e.* ICP is not guaranteed to be globally optimal). Table I compares the above-mentioned descriptors according to their sensibility, to the quality of the data, and their computational complexity.

IV. EXPERIMENTS

First, the experimental setup is briefly described from a hard and software point of view. Then a feature-based scan alignment and a brute-force scan alignment are compared.

A. Experimental Setup

The laser scanner is mounted on a ground vehicle which can move freely in two dimensions and rotate. To interface the *RPLidar A2M4* laser scanner and the *Dynamixel XL320* servo, an *Intel UPBoard* is used. This small form-factor computer features a 1.9GHz quad core processor and is capable of running an Ubuntu-Linux with ROS. A car battery is used as a power supply, but smaller batteries would suffice, since the total power consumption is typically below 10W. The institute's corridor is used as the test environment and the setup is pulled by hand, as shown in Fig. 8.

On a software level, the processing and alignment of the laser scans is implemented in a ROS node relying on the widely used *Point Cloud Library*. This library also includes implementations of many feature detector and descriptor algorithms (*e.g.* Harris3D, Spin Images, SIFT, Iterative Closest Point) and can be easily integrated into the ROS framework.

B. Experimental Results - Feature-Based Approach

The goal of the experiments was to verify that the developed 3D laser scanner has sufficient performance (resolution, update

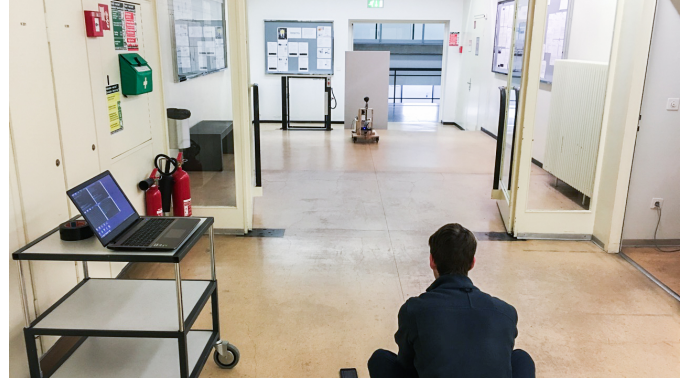


Fig. 8. The 3D laser scanner is mounted on a ground cart, which is pulled with a string

rate etc.) for a SLAM application. Since a feature-based approach is common in computer vision for robotics applications, this approach has been tried first. The keypoints are extracted with the Harris3D keypoint detector and then the Spin Image descriptor is used for the matching part. However, this approach was not successful, and three issues were observed:

- Only very few keypoints were identified.
- The descriptor needed a large support region.
- Corners were not uniquely described by the descriptor.

1) *Few keypoints*: The Harris3D detector uses surface normals to identify edges in the three-dimensional space. To reduce the effect of noise and outliers, the surface normals are estimated from more than three points. To robustly detect a corner, each enclosing wall should consist of more than 5 points. This corresponds to an angular diameter of roughly 30° in each direction. Features smaller than that, will not be detected by the Harris3D algorithm.

2) *Large support region*: Because of the low scan density, the descriptor needs a large support region. The support region is the amount of space that is described by the feature vector since all points within the support region are considered by the descriptor, when the feature vector is calculated. A large support region also means that small details have lower relative importance. Hence, the feature vectors outputted by the descriptor are less distinctive.

3) *Not unique corners*: From the two issues mentioned above, one might conclude that the ideal environment for this approach is a room with large features (tables, couch etc.) that is not cluttered with small furniture. However, this is not true due to the invariance to rotation of the Spin Image descriptor. Every rectangular corner has the same feature vector, no matter how it is oriented. Keypoints are matched using a nearest-neighbor search in a multi-dimensional space. When the feature vectors of all corners are similar, no reliable matches can be established.

From the above, we conclude that the feature-based approach is not suitable for such a low-resolution scan. Increasing the scan resolution by a factor of 5-10, would likely suffice to make the feature-based approach feasible.

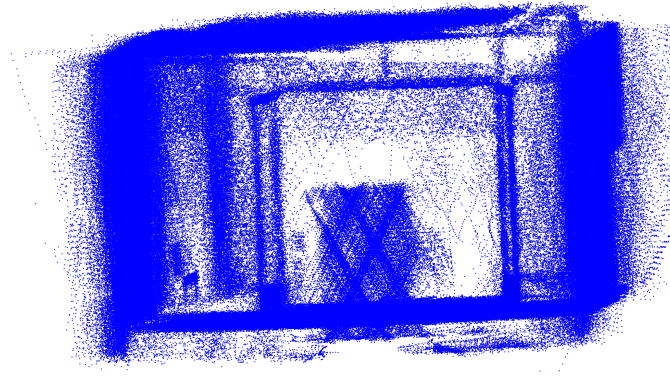


Fig. 9. Resulting 3D map of corridor shown in Fig. 8. It was generated using the non-feature based ICP approach and the low-cost 3D scanner designed in this work.

C. Experimental Results - Non-Feature-Based Approach

The non-feature-based ICP approach was implemented next. Scan resolution is not much of an issue since all points are used for alignment anyway, but one needs an accurate (≈ 30 cm) initial guess. When used on a ground vehicle or an aerial vehicle that moves slow (< 0.25 m/s) compared to the update rate of the scanner, the last known position can be used as an initial guess.

Figure 9 shows the result of the ICP algorithm, when the ground vehicle was pulled with a speed of 5 m/min. Due to the accumulative nature of the map, even details far below the individual scan's resolution (doorframe, chair on the left) become visible.

Running on the small onboard computer, a real-time implementation was not successful. It ran with a speed of about 0.5 realtime. This is rather close to real-time and hence small form-factor computers of the near future could make a real-time 3D SLAM with ICP possible.

V. CONCLUSION

This paper introduces a low-cost, light-weight and versatile 3D laser scanner. With its low power consumption and its small form-factor, it is ideally suited for onboard applications on autonomous ground robots. Aerial applications are problematic because of the slow update rate. By using a faster 2D laser scanner, the servo velocity could be increased resulting in a higher overall update rate without sacrificing resolution.

Experiments have successfully demonstrated the simultaneous localization and mapping capabilities (see Fig. 9) of this device. The laser scanner was mounted on a ground cart and a 3D iterative closest point alignment was used to generate the map. The workflow that led to the 3D map shown in Fig. 9 is summarized in Fig. 10. This map also shows small details that are not visible in the individual scans. The algorithm we used for 3D iterative closest point scan matching is currently not able to run in real time. However, this likely will be possible

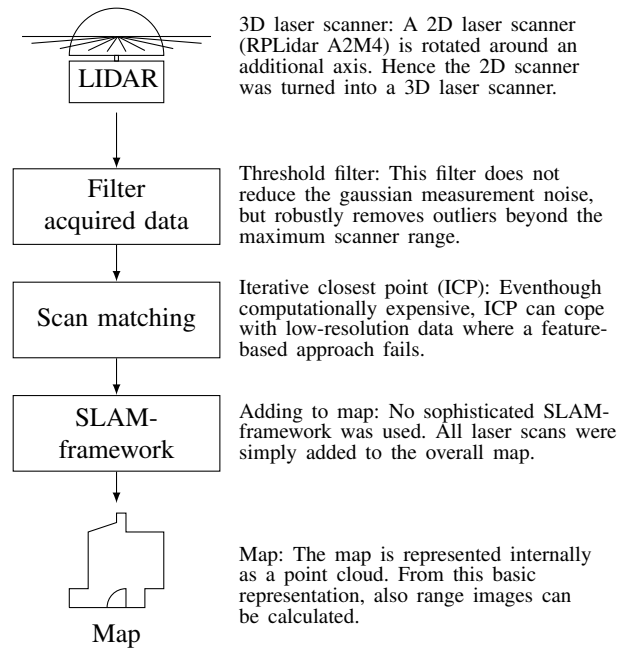


Fig. 10. Steps used to obtain a 3D map. The newly developed 3D laser scanner was mounted on a moving ground cart and slowly pulled along the institute corridor. Then the steps shown above were performed to obtain the final result shown in Fig. 9. All steps are implemented within one ROS node to achieve good computational performance.

with more powerful hardware (CPUs with higher FLOP/s per core) in the near future.

REFERENCES

- [1] X. Han and Q. Wang, "An adaptive median filtering algorithm for range image," in *Symposium on Photonics and Optoelectronics (SOPO)*. IEEE, 2012, pp. 1–3.
- [2] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 608–620, 1987.
- [3] R. Dubes and A. K. Jain, "Clustering techniques: The user's dilemma," *Pattern Recognition*, vol. 8, no. 4, pp. 247–260, 1976.
- [4] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988, pp. 147–152.
- [5] S. Filipe and L. A. Alexandre, "A comparative evaluation of 3D keypoint detectors in a rgb-d object dataset," in *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, vol. 1. SciTePress, 2014, pp. 476–483.
- [6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [7] A. Flint, A. Dick, and A. van den Hengel, "Local 3D structure recognition in range images," *IET Computer Vision*, vol. 2, no. 4, pp. 208–217, 2008.
- [8] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, 1999.
- [9] A. Frome, D. Huber, R. K. Malik, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3023, pp. 224–237, 2004.
- [10] S. A. A. Shah, M. Bennamoun, F. Boussaid, and A. A. El-Sallam, "3D-Div: A novel local surface descriptor for feature matching and pairwise range image registration," in *20th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2013, pp. 2935–2938.
- [11] J. Sprickerhof, "Pcl :: Registration," in *Karlsruhe ICRA 2013*. PCL, 2013, presentation. [Online]. Available: <http://www.pointclouds.org/assets/icra2012/registration.pdf>