Classification within Indoor Environments using 3D Perception

Lucian Cosmin Goron, Levente Tamas, Gheorghe Lazea
Robotics Research Group, Department of Automation, Technical University of Cluj-Napoca
{lucian.goron, levente.tamas, gheorghe.lazea}@aut.utcluj.ro

Abstract—**Making sense out of human indoor environments is an essential feature for robots. In this paper we present a system for the classification of components inside these environments, starting from our robotic platform to a simple yet robust labeling process. Our method starts by acquiring multiple point clouds which are then registered into one single dataset. An estimation of principle axes is performed and the planar surfaces are segmented out. Further on, quadrilateral-like shapes are estimated for each detected plane, by making use of edges. And finally, since our classification approach relies on physical features, the method analyses the relationship between the previously mentioned shapes, as well as their physical sizes. To validate our approach, we tested the method on different datasets, which were recorded inside our office environment.**

I. Introduction

Modern robots are now capable of performing the most intrinsic of tasks when it comes to indoor environments. They can navigate and find their way around, grasp and pull handles to open doors and cupboards, manipulate different objects of daily use, and even interact with humans. All of these relatively simple tasks require a considerable effort in terms of computation. And since the current systems are not always reliable and straightforward, there is still room for improvements.

The work at hand is describing an integrated system for classifying basic components of human indoor environments based on their physical features. To be more concise, our system records 3D point clouds and after processing assigns a label to each point according to the class it belongs to. For achieving this task we make use of a P3-AT[1] *(Pioneer 3 All Terrain)* robot, our 3D laser scanning system and a laptop. In the following we give a comprehensive overview of the proposed system.

We scan a certain scene from a couple of different viewpoints. Then we register these point clouds using the ICP *(Iterative Closest Point)* algorithm applied in two stages: (i) initial alignment: only for filtered set of correspondences; and (ii) refined alignment: using the complete datasets. We assume that the registered cloud is not axis aligned, thus having a random coordinate system. Since our approach relies on accurate alignment with real-world axes, our method transforms the cloud in two steps: (i) initial guess: using normals of dominant planes to compute the axes; and (ii) correct alignment: where basic features from indoor environments are used to determine the final axes. During this process the planar surfaces are segmented and the boundary points for each plane are computed. Quadrilateral shapes are then fitted to each set of boundaries. These shapes will tell us the positioning of walls and components such as doors and windows. After inspecting the sizes of these rectangle-like shapes and determining the relationships between them, the method can start assigning a class to each point.

This system can be useful in different areas of the robotic research. One would be the interpretation of registered clouds taken from human indoor environments. Also it can be useful to know where basic components, e.g. doors or windows, are situated for certain SLAM *(Simultaneous Localization And Mapping)* algorithms.

The main contributions presented in this material are:

- the construction of the 3D laser scanning system;
- a straightforward and reliable way of estimating the principle axes of 3D indoor datasets;
- a procedure for estimating quadrilateral-like forms;
- a simple set of rules for classifying indoor components.

In the following section we give a short overview of related works. Then we describe our robotic platform in Section III, followed by the registration based on features in Section IV and our classification method in Section V. We conclude in Section VI and present our future research directions.

II. Related Work

There are many possibilities to acquire 3D information from the surrounding environment. The measurement methods can be divided into three major categories based on applied sensor and sensing technology: stereo vision with two or more cameras, active triangulation and time-of-flight measurements [1]. In order to get information from the third dimension, the standard 2D laser scanners are often used with an auxiliary rotary mechanical system. The 2D laser is then mounted on the aforementioned system, obtaining the third degree of freedom for the laser beam. Such an approach based on a servo actuator system was used in [2].

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 according to different scenarios such as geometric feature-based registration for urban scenes [3] or raw point

[1] `www.mobilerobots.com/researchrobots/p3at.aspx`

cloud registration for cluttered environments [4]. A common approach is to use distinctive features for a set of points between consecutive scans for initial alignment and the ICP algorithm [5] for the fine-tuning of the registration.

In [6] the authors propose a so-called scene interpretation method for labeling the walls, floor, and ceiling. This approach relies only on the segmentation of planar patches using the RANSAC *(Random Sample Consensus)* [7] algorithm. Further identification of indoor components is obtained using a rather complex recognition pipeline.

The authors of [8] are presenting a similar approach to ours. They also use 2D quadrilaterals to detect rectangular shapes, but under the assumption that the points are axes aligned, making it easier to determine the floor and ceiling. Also, from the article we understood that the so-called cuboid fitting was applied only for that particular kitchen datasets, casting doubts about its reliability and accuracy.

Regarding to door detection there has been done some very interesting work, such as [9] where authors use computer vision to classify pixels using a somehow smaller set of classes that the one presented in this paper. Although the results look good, the performance can be easily affected by changing light conditions. Also by using only image processing, robotic applications which require 3D information cannot be supported. Another work on door detection is [10] where the developed system is accurately detecting and opening doors. Unfortunately, the authors have applied it only on doors and handles that are conform to ADA *(American Disability Act)* U.S. law, thus suggesting that the system might not comply for other types of doors.

In addition, there was very little to be found on indoor window detection, this application being more common for outdoor processing of buildings. Nevertheless in [11] a method is presented for detecting windows from outdoor scans, using indoor points. The authors also rely on the detection of dominant planes, being considered most representative. Using the indoor points, meaning the points which lie inside the buildings, and which were obtained by the laser beam going through the windows into the building, the authors estimate the positions of windows.

III. 3D Laser Range Finder 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 product for which an auxiliary mechanical part was constructed in order to earn a 3^{rd} degree of freedom. The actuated laser scanner was mounted on a P3-AT mobile robot, and the data acquisition was performed using the ROS[2] *(Robot Operating System)* 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 and roll axes, thus achieving a yawing, pitching or rolling of the 3D sensor [1]. Each of these setups has its own advantage/disadvantage. Since for the mobile robots the most common approach is the pitching scan, it was adopted for the current design. The mechanical design and the product prototype are presented in Figure 1. The mechanical design shown has two parts: one fixed, containing the driving servo motor (left) and the rotation encoder (right); and one mobile, on which the Sick LMS200 is placed. The prototype was built using an iron frame both for the fixed and for the mobile part.

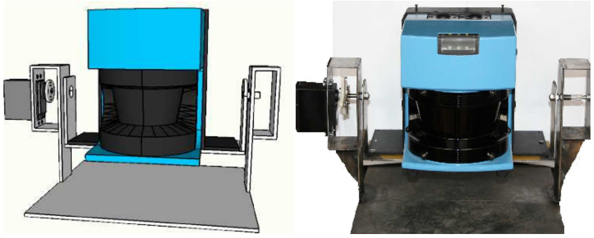

Fig. 1. The CAD model [**left**] and final prototype [**right**] of the actuated 3D sensing device used for collecting data.

As the driving motor a Hitachi $12V$ servo motor was chosen, having a minimum rotation step of $0.45°$, while for the rotation sensor a high resolution encoder was considered. The motor control and the serial interface to the PC were solved using a Cerebot2 electronic board with an AVR micro-controller. 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 up to $80m$. 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.

For a scanner with pitching actuator the 3^{rd} information about a point is from the pitch angle information. The coordinates of a 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 scan 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 taking as original coordinate system 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.

[2] http://www.ros.org/wiki/

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 tests and eliminated by considering a constant term. A more detailed discussion regarding the error budget can be found in [12].

IV. Feature Based Map Registration

Several range scans are necessary in order to build a 3D 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 registration algorithms. Since in our case the robot position could not be determined with sufficient accuracy between the measurement steps, the registration algorithms were employed for creating the elevation maps.

A. 3D Data Acquisition

The data acquisition was performed with the laser scanner presented in Figure 2 which was mounted on a P3-AT mobile robot in order to perform both indoor and outdoor scans. The scan area was of $180°(h)$ x $100°(v)$ with a horizontal resolution of 361 and a vertical one of 200 steps. This configuration ensured an optimal resolution and density of the point cloud for further processing of the data. The scanning of the environment with the mobile robot was performed in a stop-scan-go fashion, a single scan taking up to 30 seconds depending on the used configuration. All the measured data was integrated in the common software environment where each logging was timestamped for an easier off-line processing.

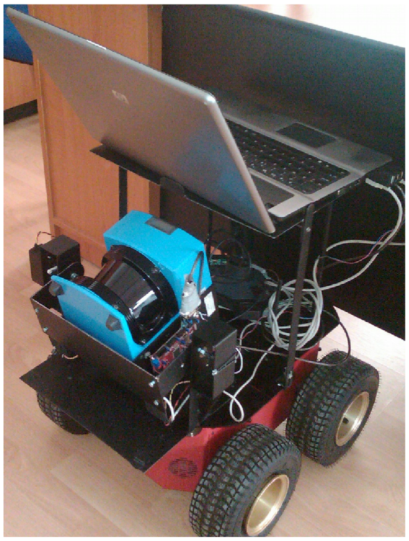

Fig. 2. The 3D laser scanning system mounted on our P3-AT robot and ready to record point clouds.

B. Multiple Scan Merging

The structure of the adopted registration algorithm follows the 3D mapping techniques commonly used in similar applications in the way it tackles the frame-to-frame matching. As the number of scans to be merged increases, so does the computational complexity [13].

Thus several problem-specific solutions were adopted in order to have an acceptable compromise between speed and accuracy. Such solutions include the scanned data filtering, key-point detection, and in some cases the odometry information fusion for the initial alignment. In our approach, the voxel grid filter was used with a downsampling factor of 5 to 10 for different datasets. Within this range the best compromise between the computational speed and merging accuracy was obtained. The initial pairwise alignment, for the two datasets, and the refined alignment were computed using the downsampled dataset and the estimated correspondences.

The initial alignment is performed using ICP only for the filtered set of correspondences. This step therefore involves only a few points to be aligned and is performed relatively fast. For the refined alignment, please see Figure 3, the whole point clouds were used as input for the ICP, thus this step of the registration needs several times more computational effort than the initial alignment phase.

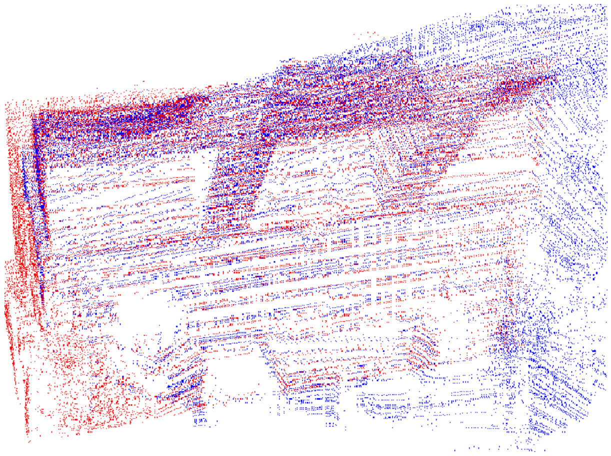

Fig. 3. Registration based on features: result of refined alignment. The point cloud represents an office area where three windows are present.

The initial alignment in this case is done with a rather good accuracy, although the refined alignment reduces the fitness score of alignment 5 times. The alignment errors are especially visible at the corners of the rectangles in the dataset, e.g. in the case of windows. The computational time of the refined alignment can be reduced by tuning the registration parameters, i.e. the maximum distance in the search space and the error threshold for the stopping criterion.

Further optimization of the registration can be performed by taking the global optimization of the registered point clouds or by using the loop closure, whenever it is possible, in order to reduce the accumulating errors during the multiple scan registration [14].

An example of the pairwise registration output is shown in Figure 4. This map is obtained combining several scans from different view-points.

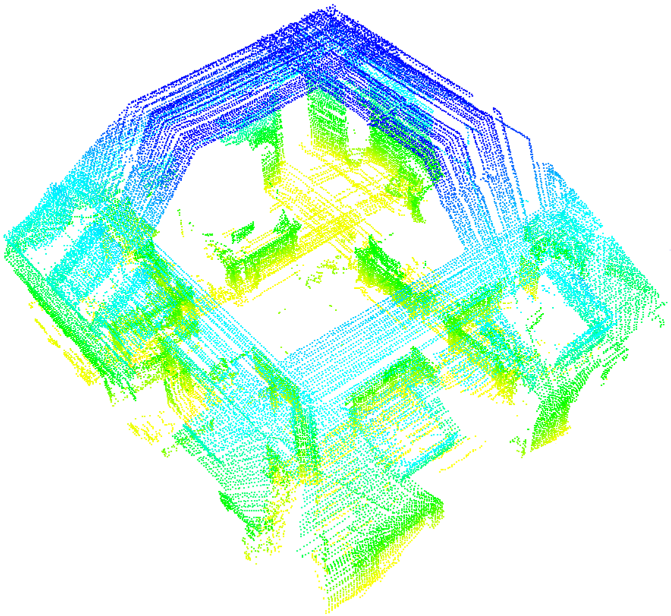

Fig. 4. Pairwise registered map of our office.

For outdoor datasets the geometric constrains were also included in the initial alignment phase, i.e. during the incremental registration the consecutive scans could be aligned only within a certain region of the global scan. In the worst cases, when the initial alignment failed due to less distinctive features, the same dynamics of motion was performed between the last pair of frames, as in the case of the previous frame registration.

V. Classification Process

With the method described in this section we implemented a simple yet robust classification process which can be easily extended with additional component classes if necessary. We also wanted for the method not to rely on any training process but rather use common sense knowledge to label 3D points. To give you some examples of what we understand as common sense about indoor human environments please read the following list:

- in the majority of cases the walls inside houses and buildings form rectangular shapes;
- by considering also the floors and ceilings we would get cuboid shapes;
- doors are always positioned on the edge of a wall, very close to the floor;
- and if closed, a door is always parallel to its supporting wall; the same for windows;
- windows are always located higher from the floor, and closer to the ceiling;
- furniture pieces are usually placed right against the walls, thus being also aligned accordingly.

For the time being, our classification procedure uses eight classes, which are also called labels: (1) *floor*; (2) *ceiling*; (3) *walls*; (4) *door frame*; (5) *door*; (6) *window frame*; and (7) *window*. It might seem redundant to have such similar classes, e.g. (4) with (5), and (6) with (7), but some components are not rigid and are constantly manipulated. Therefore, it is useful to have a system which can determine the places of those components. The method might not detect the actual door or window, but will identify the location where it should be when closed.

A. Estimation of Principle Axes

In this subsection we describe our technique for estimating the principles axes of indoor point cloud data. First we fit planes using the RANSAC [7] algorithm, by segmenting the inliers of each plane out of the point cloud. For an example of the final fitted planes, please look at Figure 5. This is repeated until the remaining points in the cloud fall under a certain threshold. The mentioned threshold is set to an empirical value of 2500 point inliers.

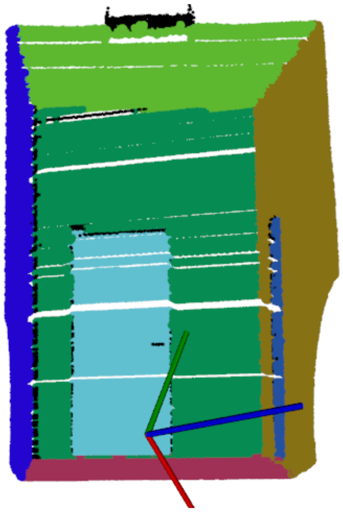

Fig. 5. Segmentation of planar surfaces. We can observe that all major planes are segmented correctly, including the two doors. Please also notice that the coordinate system is not correct, thus having the axes (X [**red**], Y [**green**], Z [**blue**]) randomly tilted in 3D space.

As expected, RANSAC always finds the dominant planes first, meaning the planes with the biggest number of inliers. Also in many indoor environments most of the surface normals coincide with one of the three main axes of the room. This is because most of the walls and furniture façades are parallel and perpendicular to each other. So the idea behind estimating the principle axes is to find three planes, which can form a right angle between each others normals. We start by comparing the normal of the most significant plane with the normal of the second most significant plane and so on. When we find a match, we mark those plane normals as axes for the new coordinate system. After we found those three planes, and their corresponding normals, we transform the point cloud in the new coordinate system. This was the initial guess stage of our estimation procedure.

Although now the points are aligned with the real-world Cartesian system of coordinates, the orientation of the three vectors representing the axes are most probably incorrect, as it can be seen in Figure 6. But the method will correct the axes in the second step, which is described in the following subsection. Also, it is important to mention that we normally need only two normal planes which are at a right angle, since the cross product between the two mentioned normals returns the third axis.

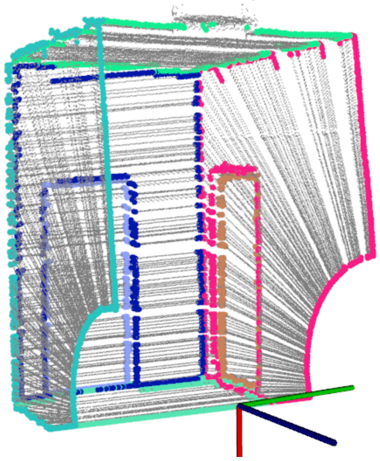

Fig. 6. Axes are aligned with the real-world, but the coordinate system is not correct, meaning that the Z-axis [**blue**] for example should point upwards, and not point towards the reader. In this figure we can also visualize the computed boundary points for each planar surface.

B. *Quadrilateral Detection and Classification Reasoning*

Here we present our strategy for detecting quadrilateral-like shapes and explain the reasoning behind the classification process used.

Our method uses these quadrilaterals to classify points which belong to doors, windows, or frames, based on their physical sizes and positioning inside the scene. By using this approach, the classification can be effortlessly extended for other rectangular-shaped components, e.g. furniture pieces, radiators, or trash bins. But in order to perform quadrilateral fitting, the method would need to execute these steps:

1. compute boundary points for each segmented plane previously found;
2. detect line models inside the sets of boundary points using the RANSAC algorithm;
3. analyze line segments to find candidates which can form rectangular-like shapes.

We estimate the planar boundary points by using the PCL[3] *(Point Cloud Library)* project. For visualizing the boundaries please take a look at Figure 6, where each plane has its boundaries colored differently. Then we continue by fitting lines much similar to the plane segmentation routine presented in the previous subsection. Whereas here the threshold for lines is set to 25 point inliers, which is also a value deduced empirically.

Quadrilateral fitting is performed by comparing lines between each other to determine if formations of rectangular-like shapes are possible. Finding quadrilateral-like forms which have right angles is done in an iterative fashion. Each iteration, a new line segment is checked and if it satisfies certain conditions, it is added to a quadrilateral configuration. A line is added to a current shape, i.e. rectangular, if two conditions are fulfilled:

(a) at least one, if not both, of the segment's ends is in proximity to either one of the shape's ends;

(b) the angle between the newly added segment and the existing one, is of approximately 90°, give or take 5°.

A quadrilateral configuration is obtained when a maximum of four line segments are found making up a rectangular shape. On the other hand, shapes which have less then three line segments are rejected. This routine stops when there are no more line segments to be analyzed. For a better understanding please consider Algorithm 1. Naturally, the results of this routine are influenced by the point cloud density, hence more points result in a better accuracy.

Algorithm 1 Estimating the quadrilateral-like shapes

```
Input: $N$                                // Number of fitted lines
Input: $S = \{l_1, l_2, ..., l_N\}$       // Set of line models
Input: $T = \emptyset$                    // Empty quadrilateral set
repeat
    $l = S(0)$                            // Start with first line
    $q = \emptyset$                       // Create empty quadrilateral
    repeat
        if $(q == \emptyset)$ then        // First edge of quadrilateral
            add $l$ to $q$
            delete $l$ from $S$
            update $N$
            $l = S(0)$                    // Restart at first line
        else
            if $((a)\ and\ (b))$ then     // Check the two conditions
                add $l$ to $q$
                delete $l$ from $S$
                update $N$
                $l = S(0)$                // Restart at first line
            else
                $l = S(l+1)$              // Continue with next line
            end if
        end if
    until $((l \leq N)\ or\ (q \leq 4))$
    if $(q > 2)$ then                     // More than two lines
        add $q$ to $T$
    end if
until $(S \neq \emptyset)$                // No more lines in set
Output: $T = \{q_1, q_2, ..., q_M\}$      // Detected quadrilaterals
```

After we find the 2D rectangular shapes, the method checks their sizes, to see if there are any doors or windows. Usually, the doors we encountered so far where around $2000mm$ by $800mm$, and windows around $1200mm$ by $600mm$, whereas

[3] http://pointclouds.org/

walls have a height of around $3000mm$. Also, this is not the only criterion by which rectangles are classified as doors or windows. There is also the positioning of those rectangles in their supporting quadrilateral, thus being usually the walls. Therefore, our method checks also for the relationships between rectangles, i.e. the relative position in comparison with one another. With the help of this information we can determine the correct alignment – please see Figure 7 – of our coordinate system.

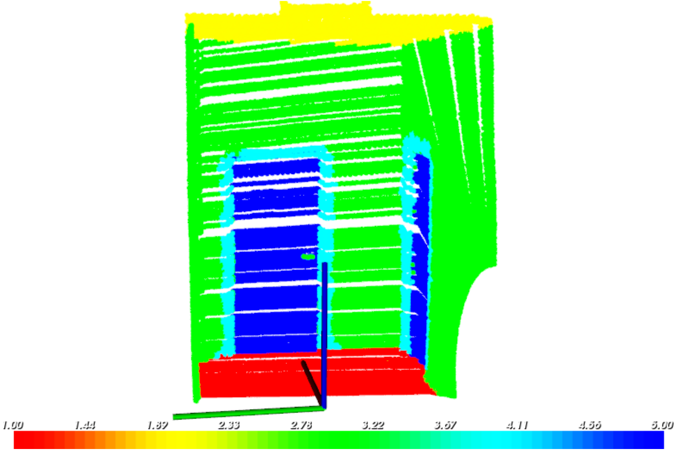

Fig. 7. Classification results where we can see the floor [**red**], ceiling [**yellow**], walls [**green**], door frames [**cyan**], and doors [**blue**]. There is also a color scaling visible. Also please notice that the axes (X [**red**], Y [**green**], Z [**blue**]) are now aligned as in the real-world environment.

As an example, think of a smaller rectangle, which is very close to one of the edges of a bigger rectangle. If that smaller rectangle within the bigger rectangle has a size close to the system's thresholds, then we can assert that the smaller rectangle is a door, which lies on the floor, and that the Z-axis should be oriented upwards in the door's direction.

If we then consider the normal of the door, as for example Y-axis, which is at a right angle with our newly found Z-axis, we then can compute the cross product and obtain our X-axis. The point clouds is transformed according to the new axes, and the correction of alignment has been fulfilled. Thus having the real-world coordinate system, we can easily determine the floor, ceiling, and walls, as shown in Figure 7, by using a pass through filter along the Z-axis.

VI. Conclusions and Future Work

We presented a system for labeling 3D point clouds taken from human indoor environments by relying on physical features. The labeling classes are as follows: *floor*, *ceiling*, *walls*, *door frame*, *door*, *window frame*, and *window*. We also described a technique for estimating the principle axes while dealing with 3D indoor datasets. From our observations, the presented approach is relatively robust to noise and easy to compute. The method was tested on different datasets with promising results.

As research perspectives there are a number of ways to improve this work, both short-term and long-term. In short-term we intend on enlarging our classification, i.e. by adding new classes to our pipeline, e.g. heating radiators and furniture pieces. And for long-term we can improve the perception by incorporating vision into the process, thus contributing to the overall robustness. We also take into consideration using the Hough transform instead of the RANSAC algorithm. Despite its lack of randomness, by using the Hough transform we will obtain the best possible fitted shapes, every time we run the classification pipeline.

Acknowledgments

This paper was supported by the project ”Doctoral studies in engineering sciences for developing the knowledge based society - SIDOC” contract no. POSDRU/88/1.5/S/60078, and by ”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, both projects co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

References

[1] O. Wulf and B. Wagner, “Fast 3D Scanning Methods for Laser Measurement Systems,” in *International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, Romania, 2003.

[2] H. K. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg, “A 3D Laser Range Finder for Autonomous Mobile Robots,” in *International Symposium on Robotics (ISR)*, 2001.

[3] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, “Fast Registration Based on Noisy Planes with Unknown Correspondences for 3D Mapping,” *IEEE Transactions on Robotics*, vol. 26, pp. 424–441, 2010.

[4] M. Magnusson, T. Duckett, and A. J. Lilienthal, “Scan Registration for Autonomous Mining Vehicles Using 3D-NDT,” *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[5] Z. Zhang, “Iterative Point Matching for Registration of Free-Form Curves and Surfaces,” *International Journal of Computer Vision*, vol. 13, pp. 119–152, October 1994.

[6] A. Nüchter and J. Hertzberg, “Towards Semantic Maps for Mobile Robots,” *Robotics and Autonomous Systems*, vol. 56, pp. 915–926, November 2008.

[7] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, June 1981.

[8] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D Point Cloud Based Object Maps for Household Environments,” *Robotics and Autonomous Systems*, vol. 56, pp. 927–941, 2008.

[9] A. C. Murillo, J. Košecká, J. J. Guerrero, and C. Sagüés, “Visual Door Detection Integrating Appearance and Shape Cues,” *Robotics and Autonomous Systems*, vol. 56, pp. 512–521, June 2008.

[10] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, “Laser-based Perception for Door and Handle Identification,” in *International Conference on Advanced Robotics (ICAR)*, Munich, Germany, 2009.

[11] S. Tuttas and U. Stilla, “Window Detection in Sparse Point Clouds using Indoor Points,” in *ISPRS Conference: Photogrammetric Image Analysis (PIA)*, vol. XXXVIII-3/W22, Munich, Germany, 2011.

[12] A. Zhang, S. Hu, Y. Chen, H. Liu, F. Yang, and J. Liu, “Fast Continuous 360 Degree Color 3D Laser Scanner,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences: WG I/2 SAR and LiDAR Systems*, vol. 36, pp. 409–415, 2008.

[13] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments,” in *RGB-D: Advanced Reasoning with Depth Cameras, workshop in conjunction with RSS*, Zaragoza, Spain, 2010.

[14] K. Ohno, S. Tadokoro, K. Nagatani, E. Koyanagi, and T. Yoshida, “Trials of 3-D Map Construction Using the Tele-operated Tracked Vehicle Kenaf at Disaster City,” in *International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010, pp. 2864–2870.