Semantic Labelling of 3D Point Clouds using Spatial Object Constraints

Malgorzata Goldhoorn[1] and Ronny Hartanto[2]

[1]University of Bremen, Robotics Research Group, Bremen, Germany

[2]DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 5, 28359 Bremen, Germany

Keywords: Point Clouds, Probabilistic Methods, Scene Understanding, Object Recognition, Semantic Labelling, 3D Scene Interpretation, Feature Extraction.

Abstract: The capability of dealing with knowledge from the real human environment is required for autonomous systems to perform complex tasks. The robot must be able to extract the objects from the sensors' data and give them a meaningful semantic description. In this paper a novel method for semantic labelling is presented. The method is based on the idea of connecting spatial information about the objects to their spatial relations to other entities. In this approach, probabilistic methods are used to deal with incomplete knowledge, caused by noisy sensors and occlusions. The process is divided into two stages. First, the spatial attributes of the objects are extracted and used for the object pre-classification. Second, the spatial constraints are taken into account for the semantic labelling process. Finally, we show that the use of spatial object constraints improves the recognition results.

1 INTRODUCTION

For some time, there has been increasing interest to develop autonomous systems, which can support the human in everyday tasks. The robot should help people by, for example, preparing breakfast or cleaning the room. This is, of course, a futuristic scenario, because the areas of robotics and AI are very challenging and there are many problems that must be solved, until this becomes a reality. One of the so far unsolved problems is understanding the real human environment. To do a task planning, the robot must know the meaning of the objects in a given task and at the same time deal with missing information, resulting through occlusion and partly caused by noisy sensors. In this paper we describe a method which may be used for such a purpose. The main contribution is the presentation of a new idea which combines the spatial information about the object with constraints between objects using probabilistic methods.

This paper presents our approach for semantic labelling of 3D point clouds, in which the transition from the spatial into the semantic domain is done. The remainder of this paper is organized as follows. Section 2 gives an overview of the state of the art in this field. Section 3 introduces the method for point cloud segmentation. Section 4 describes the first step of the approach, namely the probabilistic object pre-classification. Section 5 presents the idea of the object constraints. Section 6 gives a quantitative survey of our algorithm. Finally, conclusions and opportunities for future work are given.

2 RELATED WORK

For some time, semantic perception became one of the most investigated research areas in robotics. This is not least because of the increasing amount of low cost 3D sensors like the Microsoft Kinect, but also the fact that semantic perception is a capability which autonomous systems need to be equipped with to perform complex tasks (Galindo et al., 2008), (Pangercic et al., 2010). One of the recent works in this field was presented in (Anand et al., 2012). The authors proposed a method for semantic labelling and search in indoor scenes using a geometrical context. In their approach, merged point clouds taken with a Kinect sensor are used. They try to extract geometric primitives from the data. To obtain a better view of the scene, an active object recognition is used. Günther et al. (Günther et al., 2011) present another related work about semantic object recognition from 3D laser data. In this work, a CAD-based method for object classification was proposed. For this, the geometrical basic primitives of the objects are extracted and com-

Goldhoorn M. and Hartanto R..
Semantic Labelling of 3D Point Clouds using Spatial Object Constraints.
DOI: 10.5220/0004874205130518
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (WARV-2014), pages 513-518
ISBN: 978-989-758-002-4

pared if they fit a given, known CAD model. Nüchter et al. (Nüchter and Hertzberg, 2008) introduce a 6D SLAM approach with semantic object recognition. In this approach, the objects (like walls, doors, and ceilings) are recognised from composed point clouds. To classify other objects like robots or humans, trained classifiers are used. Another approach for semantic labelling was proposed in (Rusu et al., 2009). The authors of this work use model-based object recognition and try to recognise household objects in a kitchen environment, like furnitures and stoves. To infer about these objects, the furniture features like knobs and buttons are extracted beforehand. In (Aydemir et al., 2010), semantic knowledge is used to search for specific objects. The authors try to find a potential place in which the object could be found using a reasoning module. The authors in (Galindo et al., 2008) and (Galindo et al., 2005) describe the use of semantic maps for task planning and spatial reasoning. They use marker identification to perform semantic interpretation of entities and to bridge the gap between the semantic and spatial domains.

3 APPEARANCE BASED OBJECTS PRE-IDENTIFICATION

In human living environments and especially in the domestic one, many regularities with regards to the objects' occurrences can be found. For example, some objects like furniture have defined heights and are larger than some other objects. Other objects like flat screen or keyboard have approximately the same width but they have different depth. In general, the objects could be distinguished from each other based on their different spatial features. We make use of those premises to do the object pre-classification step. The next very important point is that most of the objects in the domestic environment are approximately planar surfaces. Therefore we try to segment these planes and extract the spatial features from them. We call this step "pre-classification", because in this stage of our algorithm the objects are classified only by their spatial features, without taking into account their spatial relations to each other. As a result of this step we obtain a probability distribution about object classes given the measured values.

3.1 Point Cloud Pre-processing

The object recognition approach starts with the segmentation of planes from the raw 3D point cloud data. The data is taken with a tilting LIDAR (Light Detection and Ranging) laser system. For the segmentation, an optimised region growing algorithm is used. This algorithm based on the approach in (Vaskevicius et al., 2007) and was already mentioned in (Eich et al., 2010). We extended this algorithm to deal with an unorganized point clouds, like in the case of data from our tilting system. In such point cloud data, the points are not available in memory as a grid and their nearest neighbours cannot be accessed directly. Because of that, the complexity of the algorithm increases by the nearest neighbours search. Therefore, we made some optimisation steps, which make the algorithm much faster then the original one (Vaskevicius et al., 2007). We do not describe the algorithm in detail, because it was already mentioned in our other work (Eich et al., 2010). The algorithm segments the input 3D point cloud into planes, which can be used for the future processing step. The region growing needs as an input different starting parameters, whose values determine the result of the segmentation. These parameters are: the maximum distance between two points, the maximum distance between a point and plane, the maximum mean squared error of the plane, the minimum number of points, which a plane needs to have, and the maximum number of nearest neighbours of a single point. The algorithm ends when all points have been examined and results in a set of extracted planes. Fig. 1 shows the result of the segmentation after applying the listed parameters.

3.2 Extraction of Spatial Object Attributes

The extraction of spatial features of the objects starts once the planes are segmented. In this step the spatial features of each plane, like the size of a plane $A \in \mathbb{R}$, the rectangularity of the plane $R \in [0,1]$, its length and width $E \in \mathbb{R}^2$, its orientation $O \in \mathbb{H}$, and its center of mass $P \in \mathbb{R}^3$ are extracted and stored in a so called feature vector $\Phi = (A, E, R, O, P)$. For better identification of these features, the found regions are first projected into 2D space. This is done by applying the reverse rotation for pitch and yaw to the original translated plane. In the end, the normal vector of the plane is parallel to the global z-axis. We assume that through our region growing algorithm this object approximates a planar surface. Since we already rotated and translated the surface into the xy-plane we can simply set all remaining z-values to zero. By doing this we project the approximately planar surface to a planar surface. Afterwords the calculations of their hulls take place. For this, an alpha shape method from the computational geometry algorithm library CGAL

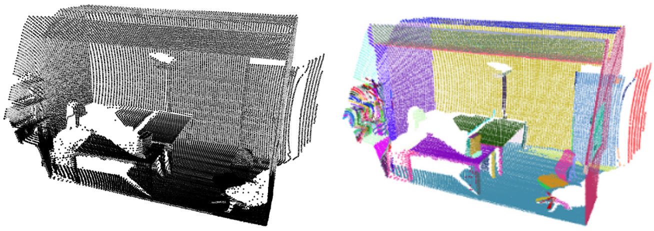

Figure 1: Result of the region growing segmentation algorithm. The left image shows the given input raw data taken in one of our offices and the right one presents the result of the segmentation, the randomly coloured planes.

is used. Having the 2D hull of a plane enables easier extraction of the geometrical features. Figure 2 shows the extracted plane together with its hull.

These five features are required to distinguish between different classes of objects. The most relevant features are orientation and position. Using these attributes it can be decided e.g. if a given object is rather a flat screen or a keyboard. On the other hand, the position can be used to distinguish between objects like walls, floors, and ceilings. The other features like size, maximum expansion and rectangularity of the plane increase the assignment of the object to a given object class. At this point, it is important to mention that the feature rectangularity is not a very critical feature for the classification and it is hard to define, and describe how rectangular an object is. Nevertheless, it improves the detection of appropriate objects. In the following, a short description of the features and their processing are given. For the extraction of the maximum *expansion* of the plane we use a method from the well-known OpenCV library. This function calculates a minimum bounding box of the plane from its hull. This bounding box has attributes like width, height, and orientation. We take the width and height of the box and treat them as the width and height of the plane. The *size* of a segmented plane is calculated using the even-odd algorithm. To do this, a set of 2D-vectors are created to form the hull by using CGAL functions. From these vectors a filled bitmap of each object is created using this algorithm. We choose a standard size of 640x480 to represent the converted bitmap objects. The scaling factor is created by using the information we already gathered from the maximum expansion which was explained before. The *rectangularity* of the plane can be computed by dividing the area of the plane by the area of its bounding box. Here, an assumption must be made that a perfectly rectangular plane has the same area as its bounding box. This is, hence, a critical point since occlusion of objects could influence the recognition. The resulting value is the percentage correspondence of area sizes, while the percentage squareness according to the definition mentioned above. The calculation could here be done straight forward since through the size computation of the created bitmap a normalization took already place. We count only the filled pixel and divide them by the total pixel count of our virtual bitmap. The *orientation* of the plan is calculated using the GNU Scientific Library. The mass centroid axes are determined by eigenvectors and their eigenvalue. We define that the eigenvector with the smallest eigenvalue corresponds to the z-axis of the plane and denotes its normal vector. The eigenvector with the intermediate eigenvalue denotes the shorter axis, namely the x-axis. The last eigenvector with the biggest eigenvalue coincides with the y-axis of the plane. By assuming that the z-vector has the smallest eigenvalue, the objects were nearly "flat" on their (local) x-/y- plane. This assumption helps us with all post-processing steps, because we can now handle objects completely invariant from their orientation. The *position* of the plane results from the center of gravity of all points of the plane.

3.3 Evaluation based on Objects Appearance

After extracting the spatial features of the plane, the evaluation step is done. To correctly evaluate the vector, a priori knowledge of the objects in the environment must be taken into account. This knowledge indicates how the objects in a typical human environment look like and what their spatial values are. If we take a table as an example then we know that it stays on the floor, is very often rectangular and its height is about 0.8 meters. Such an assumptions can be made for all objects in the human environment. Exactly this knowledge is stored in a database and serves as an input for the evaluation function, with the aid of the features are evaluated. In the database, objects like table, wall, floor, ceiling, keyboard, flat screen, and "unknown object" are stored. Each spatial value of an object contains an expected value μ and standard deviation σ. These two values enable to having objects which different appearance and helps with their robust pre-classification. Because of the difference in

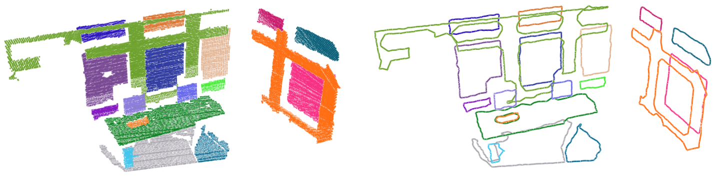

Figure 2: Result of the alpha shapes calculation together with the associated regions, segmented in the previous step.

the importance of the given attributes, a weight is applied to each of them. The features are evaluated by using the well known Bayes' theorem. First, we calculate the Gaussian distribution function, which gives the probability of the measured value x to the known object class C from the database. After some calculations and reformulations we obtain the formula 1:

$$P(C|\Phi) = \frac{\sum_{i=1}^{n} P(C|x_i) \cdot F_i}{\sum_{i=1}^{n} F_i} \quad (1)$$

This results in a probability distribution for a given object class C from the data base given the measured feature vector Φ.

4 SEMANTIC LABELLING WITH SPATIAL CONSTRAINTS

The last step in our semantic labelling algorithm is the classification of the objects based on their spatial relations. In the previous processing step, each segmented plane was assigned to each object class from the database with given probability. This objects assignment to a given class is based exclusively on the spatial attributes of the object. Because of noise and occlusion in the data, this can in many cases result in an incorrect classification of the object. In order to improve the labelling we take spatial relations between all object classes into account. For this we define a constraint network similar to (Nüchter et al., 2003), in which the objects and their spatial relations are stored. We use relations like "parallel" , "orthogonal", "above", "under", or "equal height". We treat these relations as constraints that must be satisfied for an object to belong to a given object class. In this final step of our algorithm, the valid world model is tried to be found. To do this, the resulting objects of the pre-classification are sorted by their height. This enables finding a ground plane (e.g. floor), which is necessary to build the scene right up to the ceiling, iteratively. The condition is, that the object is pre-classified as a floor with a probability of at least 40%. Further, the next objects from the sorted list are taken. During this, it is checked if the constraints related to the relations between objects are satisfied. If not, the current path will be discarded and a simple backtracking takes place. For our heuristic we use the probability of associated object classes that we calculated for each plane in our previous step (see Sec. 3.1). By using this heuristic, in the case that the pre-calculation was already correct, the right path would be taken directly without any search. If during traversing the path, an invalid model is found, it is discarded automatically and the next best path is chosen. This is done, because we assume that an error in the world-model is (mostly) related to wrong hierarchies of objects. An example could be when an object that is possibly a flat screen is not located on top of a desk. This whole search/backtracking is repeated until the ceiling is found. The result of the algorithm is a list of labelled objects together with their adjusted probability. In this way the objects get clear semantic descriptions.

5 EXPERIMENTS AND RESULTS

We have done several experiments using both raw point cloud and synthetic data to evaluate our algorithm. The raw data was taken in several offices of our lab. This data has been recorded using a Hokuyo UTM-30LX laser scanner mounted on a tilting unit. The synthetic data has been generated using Blender with some artificial noises. At the beginning, the pre-classification step of our algorithm was tested. For this we took the real and synthetic data and tried to figure out how good the extracted spatial features matches the ground truth data to evaluate the overall measurement accuracy. For this the large set of perfect generated sensor data were used. The result of the test is that besides some rounding problems (potentially caused by the floating point precision) the measured values correspond to the ground truth data. Further, we evaluate the influence of the each changed feature on the probability result. As expected, the changes in the value of the features has no large impact on the result of the pre-classification. This is because it is not relevant if a table is farther right or left, only the height of the objects has a large importance for their recognition. An example evaluation can bee

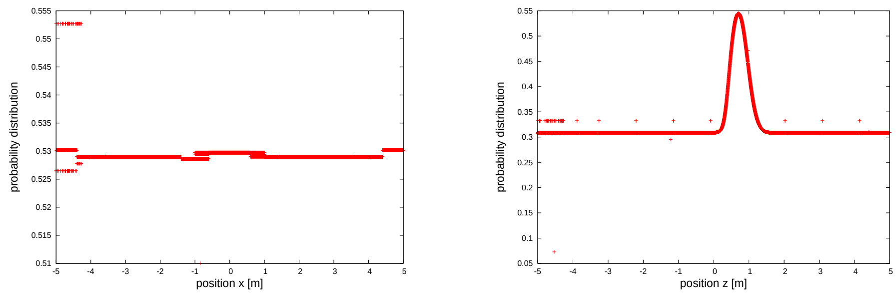

Figure 3: Influence of the position in the x- and z-axis on the result of the probability distribution. On the left graph it can be seen, that the position in the x direction has no real influence on the result of the pre-classification. In contrast, the height of the object influences strongly their classification

Table 1: Result of the semantic labelling algorithm before and after applying the spatial object constraints.

Objects	real objects	feature evaluation	final result	false positives	false negatives
Table	12	13	13	1	0
Flat screen	9	190	13	9	5
Keyboard	7	6	3	0	4

seen in the Fig. 3. Then, we evaluate the overall result of the labelling algorithm. For this, we started it with real scans taken from several office rooms of our institute to check how well the method works. In these tests, our assumptions were confirmed. Taking into account the spatial relations between objects improves the result of the labelling significantly. This is because this additional information has a large influence on the association of the objects to their respective object classes. The result of the experiments shows that after taking into account this information many false positives, as in the case of flat screen, resulting from the first step are eliminated. In this way, the recognition result was improved. Table 1 presents the result of the semantic recognition, before and after applying the spatial relations between objects. In the second column the number of office objects form the six real scenes is given. In the next one the recognized objects, after evaluation of the feature vector, are presented. It can be seen that many false positives (e.g. flat screen) have occurred. The fourth column shows the result after taking into account the spatial relations between objects. This presents that applying of the spatial constraints improve the recognition result, since the number of false recognized flat screens is reduced.

6 CONCLUSIONS AND FUTURE WORK

This paper describes our approach for semantic labelling of objects from 3D point clouds. This method combines spatial information about the objects with their relationships to each other. We showed that the application of object constraints improves the labelling process. This was shown in our experiments by reduction of false positives. Further, we presented how probabilistic methods can be used for this issue. The future work will be mainly on improvement of the recognition process. We are planning to archive this by adding probabilistic approaches to the constraint checking process. In addition, we would like to extend our algorithm to other object classes and use more spatial relations apart from those mentioned in this paper. Moreover it should handle more complex objects consist of multiple surfaces. Our goal will be to extend the approach to other application scenarios, like kitchens or robots operating in complex environments that required e.g. stair climbing (Eich et al., 2008) or deal with outdoor obstacles (Spenneberg and Kirchner, 2007), (Bartsch et al., 2010).

ACKNOWLEDGEMENTS

This work has been supported by the Graduate School SyDe, funded by the German Excellence Initiative within the University of Bremen's institutional strategy. The project IMPERA is funded by the German Space Agency (DLR, Grant number: 50RA1111) with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament.

REFERENCES

Anand, A., Koppula, H. S., Joachims, T., and Saxena, A. (2012). Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*.

Aydemir, A., Sjöö, K., and Jensfelt, P. (2010). Object search on a mobile robot using relational spatial information. *Proc. Int. Conf. on Intelligent Autonomous Systems*, pages 111–120.

Bartsch, S., Birnschein, T., Cordes, F., Kuehn, D., Kampmann, P., Hilljegerdes, J., Planthaber, S., Roemmermann, M., and Kirchner, F. (2010). Spaceclimber: Development of a six-legged climbing robot for space exploration. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–8.

Eich, M., Dabrowska, M., and Kirchner, F. (2010). Semantic labeling: Classification of 3d entities based on spatial feature descriptors. In *IEEE International Conference on Robotics and Automation, ICRA*.

Eich, M., Grimminger, F., and Kirchner, F. (2008). A versatile stair-climbing robot for search and rescue applications. In *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pages 35–40.

Galindo, C., Fernández-Madrigal, J.-A., González, J., and Saffiotti, A. (2008). Robot task planning using semantic maps. *Robot. Auton. Syst.*, 56(11):955–966.

Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernández-Madrigal, J., and González, J. (2005). Multi-hierarchical semantic maps for mobile robotics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3492–3497, Edmonton, CA. Online at http://www.aass.oru.se/∼asaffio/.

Günther, M., Wiemann, T., Albrecht, S., and Hertzberg, J. (2011). Model-based object recognition from 3d laser data. In *Proceedings of the 34th Annual German conference on Advances in artificial intelligence*, KI'11, pages 99–110, Berlin, Heidelberg. Springer-Verlag.

Nüchter, A. and Hertzberg, J. (2008). Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926.

Nüchter, A., Surmann, H., Lingemann, K., and Hertzberg, J. (2003). Semantic scene analysis of scanned 3d indoor environments. In *VMV*, pages 215–221.

Pangercic, D., Tenorth, M., Jain, D., and Beetz, M. (2010). Combining perception and knowledge processing for everyday manipulation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1065–1071.

Rusu, R., Marton, Z., Blodow, N., Holzbach, A., and Beetz, M. (2009). Model-based and learned semantic object labeling in 3d point cloud maps of kitchen environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3601–3608.

Spenneberg, D. and Kirchner, F. (2007). The bio-inspired scorpion robot: design, control & lessons learned. *Climbing & Walking Robots, Towards New Applications*, pages 197–218.

Vaskevicius, N., Birk, A., Pathak, K., and Poppinga, J. (2007). Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors. In *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*, pages 1–6. IEEE Press.