Georgia Institute of Technology, Technical Report, GT-GOLEM-2012-001

Detecting Partially Occluded Objects via Segmentation and Validation

Martin Levihn   Matthew Dutton   Alexander J. B. Trevor   Mike Stilman

Abstract— **This paper presents a novel algorithm: Verfied Partial Object Detector (VPOD) for accurate detection of partially occluded objects such as furniture in 3D point clouds. VPOD is implemented and validated on real sensor data obtained by our robot. It extends Viewpoint Feature Histograms (VFH) which classify unoccluded objects to also classifying partially occluded objects such as furniture that might be seen in typical office environments. To achieve this result, VPOD employs two strategies. First, object models are segmented and the object database is extended to include partial models. Second, once a matching partial object is detected, the full object model is aligned back into the scene and verified for consistency with the point cloud data. Overall, our approach increases the number of objects found and substantially reduces false positives due to the verification process.**

I. Introduction

The ability to reliably detect and classify objects is crucial to the success of robots in realisitic, human environments. Knowledge about the existence and position of objects within the robot's vicinity has many practical applications in robotics. For instance, semantic mapping, the creation of maps that include meaning, could be enhanced by detecting meaningful objects, such as furniture. Knowledge about furniture helps to enable robots to understand commands such as "bring me the mug from the table". In addition, the classification of rooms can benefit from knowledge about furniture. For example, a room with a table and many chairs is likely to be the dinning room.

Moreover, this work is strongly motivated by recent work in the field of Navigation Among Movable Obstacles (NAMO) [11]. In NAMO, the robot attempts to reach a fixed goal position in a reconfigurable environment. The planner presented in [11] is capable of dealing with partial world knowledge and incrementally adding new information to the world model once it is perceived. However, in order to transfer the planner to a physical system, the robot must be able to detect movable objects based on real sensory information, such as 3D point clouds obtained by a laser range finder.

All of the examples provided above must operate in human environments. In these environments, objects are often partially occluded by other objects, walls, and people. Any algorithm trying to reliably detect objects in human environments, such as furniture, must therefore be able to deal with occlusions and arbitrary orientations.

Previous work has shown success in classifying mostly unoccluded objects but does not perform well in human environments with frequent, large occlusions. This paper presents

The authors are affiliated with the Center for Robotics and Intelligent Machines (RIM) at the Georgia Institute of Technology, Atlanta, Georgia 30332, USA. Emails: levihn@gatech.edu, matt.dutton@gatech.edu, atrevor@cc.gatech.edu, mstilman@cc.gatech.edu

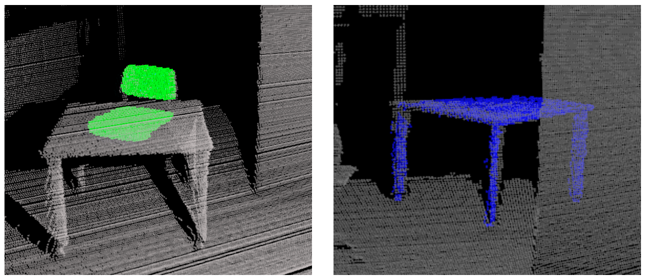

(a) Example of a detected chair. The detected chair model is colored in green.
(b) Example of a detected table. The detected table model is colored in blue.

Fig. 1. Example of correctly classified furniture. The chair is correctly detected despite the fact that only the top part of the chair is actually visible in the scan. Similarly, the table is detected despite the fact that it is partially occluded.

the Verfied Partial Object Detector (VPOD) algorithm that is capable of working in these environments, even when more than 50% of the object is occluded. VPOD segments a point cloud of a scene into clusters by point distances and classifies each resulting cluster. This classification is based on a two step approach. The first step builds upon Viewpoint Feature Histograms (VFH) [9], a descriptor for 3D point cloud data that encodes geometry and viewpoint. The VFH of the query object is computed and compared to VFHs in a database composed of both full object models and auto-generated partial object models.

A second step verifies each candidate match against the actual full point cloud, to eliminate full positives introduced by partial models. To verify these matches, the algorithm maps the full model associated with each candidate into the actual full point cloud of the scene. Points of transformed models that are occluded by objects in the full point cloud are eliminated from the model. The remaining points are checked for matching points in the full point cloud. The proportion of matching, unoccluded points yields the final classification score. This steps ensures that matches for clusters in the point cloud are consistent with our expectation of the model in the context of the world and the current viewpoint.

This paper addresses this topic in the domain of recognizing furniture and is organized as follows. After discussing related work in Section II, we will briefly describe the basic problem of full-model VFHs in Section III. A detailed analysis of our approach will then be provided in Section IV and experimental results be demonstrated in V. Section V will also discuss the limitations of our approach. The paper will be concluded with a future work in Section VI.

II. Related Work

The detection of objects in 3D laser data has been studied intensively in various research fields. As such, we are only addressing the work most relevant to ours.

In [8] Rusu et al. present a system for the acquisition of hybrid Semantic 3D Object Maps for indoor household environments based on 3D point cloud data. The authors use a two step approach for detecting kitchen furniture based on the detection of horizontal and vertical planes as well as knobs.

In [3] the authors describe a mapping system acquiring 3D object models for indoor environments. The authors present a system for segmenting and geometrically reconstructing cabinets, tables, drawers and shelves based on multiple scans of the objects. Tables, the most relevant part to our work, are detected by finding horizontal surfaces within a given height range. This does not allow for distinguishing between, for example, tables and shelves.

Holz et al. are using 3D Time-of-Flight cameras in [4] for semantic scene analysis. The authors are using an MSAC-based approach and surface information in a point's local neighborhood to detect table tops. This is done by assuming that a table top point has a surface normal nearly parallel to the z-axis and that the local surface is smooth. MSAC is used to fit planar surface models into the table point set. This approach yields the same limitations as [3].

Johnson et al. presents in [5] a shape-based object recognition system based on matching Spin Images. Spin Images however require a high resolution image, and as such are difficult to use in 3D point clouds obtained by a laser.

Marton et al. incorporate in [6] 3D laser scans and 2D vision data for object classification. They use the Radius-based Surface Descriptor (RSD) on 3D data, extract the region of interest into a camera image, and compute a 2D SURF vector for each patch. The final classification is performed through a SVM. However, it remains unclear how this work can be extended to work with furniture, which usually lacks texture. In addition, partial occlusions would be difficult to handle by this approach.

Steder et al. [10] demonstrated the detection of chairs and other objects in 3D point clouds using point features from range images. The authors use euclidean distance in a vector space spanned by Harris feature vectors to find candidates. GOODSAC is used to find a model transformation and false positives are rejected based on a score function using scaled range images. Steder also presented the normal aligned radial features (NARF) in [1]. Interest points are detected on stable surface areas with significant local changes. A descriptor is then computed by overlaying a star pattern on the range image generated by looking at the interest point along the estimated normal for this point. The authors also describe the potential of matching the feature descriptors as an object recognition approach. However, we experimented with NARF and found that the descriptor is only of limited use in object recognition due to a lack of local texture in range images and the loss of valuable orientation information during normal alignment. E.g. a horizontal surface becomes indistinguishable from a vertical one.

Viewpoint Feature Histograms as presented by Rusu et al. [9] are encoding geometry as well as viewpoint information into a descriptor. The authors demonstrate the effectiveness of the descriptor on a dataset consisting of more than 60 unoccluded kitchenware objects. The first part of VPOD builds upon this work to support partial occlusion and is presented in the following section.

Mozos et al. [7] demonstrate a method of categorizing partially occluded objects from object parts learned from segmented 3D models, which are first segmented based on the object's structure. The database of segmented parts is used to suggest categorizations from a scene. The candidates are combined through Hough voting and verified through model fitting. As this approach segments based on the object's structure, the segmented parts do not correspond exactly with occlusions caused in real scenes, which is a property of the occluding object and the scene, not the occluded object. It remains unclear whether [7] is sensitive to partial occlusion of the segmented parts. Our approach segments objects based on common occlusions and relies instead on real 3D scans of objects, which can be generated by the robot, and not prebuilt models.

III. Viewpoint Feature Histrogram

Viewpoint Feature Histograms are histograms describing the geometrical relationship between all points in the object. Rusu et al. [9] show that VFHs can discriminate according to the structure of the entire object, if the entire object is visible.

However the VFH is sensitive to partial occlusions, as in Fig. 1(a) where only the top part of the chair is visible in the scan. As the top part of the chair is roughly just a flat surface, the points have an entirely different geometrical relationship to each other than all the points for an unoccluded chair. This results in different VFHs for the full and partial chair, as shown in Fig. 2. The components of the VFH are described in [9].

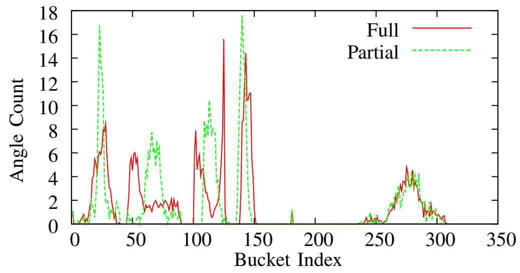


Fig. 2. Example VFHs for a partial model and its associated full model

The consequence is that partially occluded objects are not reliably detectable with VFH or any other method that works with the properties of the full object alone.

IV. Algorithm

A. Approach

The key concept here is that fragments of a partially occluded object can be treated and classified as objects themselves.

VPOD extends the VFH database to include partial models auto-generated by occluding portions of full models. The partial model generation process is designed to generate typical occlusions occurring in the world. For human environments, we assume that the objects are usually occluded from one side (e.g partially behind a wall) or the bottom (e.g. a chair underneath a table). Our algorithm therefore generates partial models out of every full model by successively removing points from each side of the object and from the bottom independently. This is done for a step size s and continued until a threshold t is reached for the remaining object size on the side currently affected by the removal. Fig. 3 shows an example. Other types of common occlusions can easily be added by generating additional partial models. The resulting partial models are included in the VFH database *with* the full models.

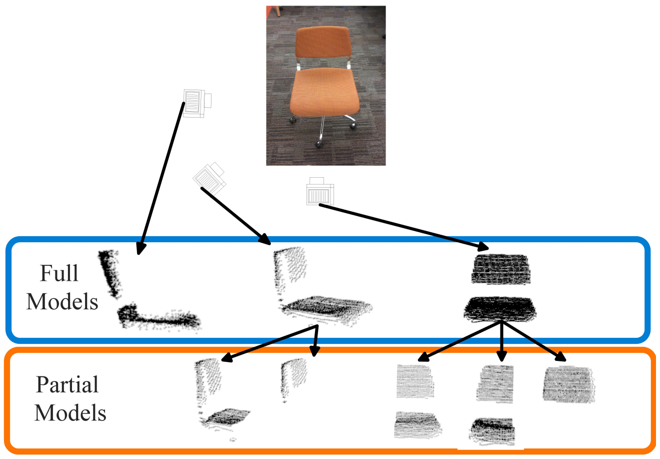


Fig. 3. Hierarchy of full models and auto generated partial models.

Including partial models in the VFH database increases the likelihood that a partially occluded object will be matched. For example, it is now possible to match the back of a chair against models in the database that represent just the back of a chair. But this also increases the likelihood of false positives, as the partial models are less distintive (e.g. the back of a chair is mostly flat). Extending the database to artificially created objects, especially pieces of objects, allows for matches of arbitrary objects. For example, the back of the chair could now be matched against a simple piece of board. To compensate for this, the algorithm includes a verification step that verifies the candidate classifications against the actual scene in which the point cloud was captured (called the "world" from here on).

The intuition behind this verification step is that we can test that the full model assoicated with the matched partial model could produce the observed cluster, given the occlusions in the world. For example, if we matched the back of a chair against an object in the world, then the full chair should be consistent with the world. Which parts of a model should be occluded can be determined by a simple line-of-sight test against the scan of the world.

We have implemented this intuition:

1) the model, from which the matched partial model was obtained, is mapped into the world (section IV-C)
2) the parts of the full model that should be occluded based on information provided by the world are removed from the model (section IV-D)
3) the remaining model points are checked for matches in the world (section IV-E)
4) based on this score, the model is rejected or verified (section IV-E)

Fig 4 shows the detailed workflow and each step is discussed in detail in the following.

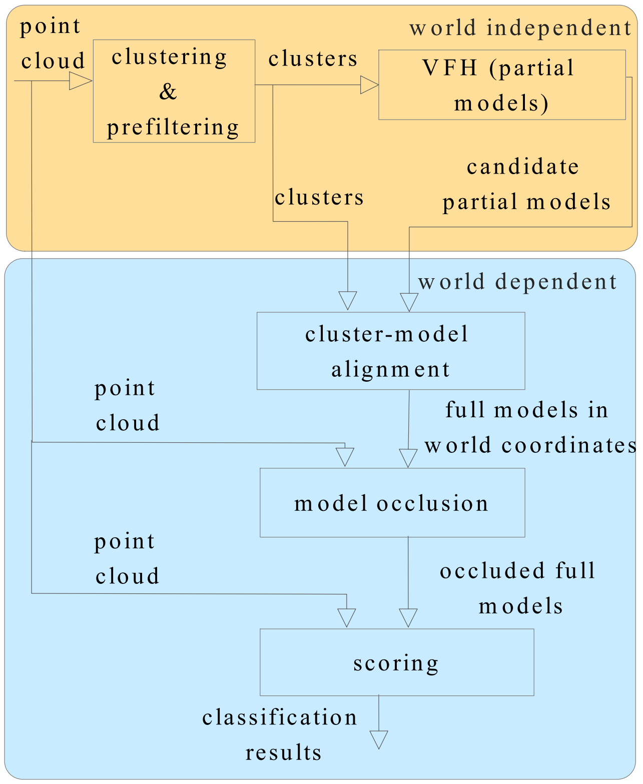


Fig. 4. Workflow. Steps independent of the actual world are colored in orange while world dependent steps are colored in blue.

B. Clustering

Point clouds collected by the robot are segmented into query clusters. First, we filter the point cloud to remove outlying points that are not near any surface or object. This type of noise is especially common on occlusion boundaries with laser range finders. The remaining points are then downsampled to reduce computation time. A grid size of 1 cm was used for the downsampling in this work. Because the sensor's height relative to the ground is known, we can easily remove the ground plane by discarding any points below a given z-coordinate. With the ground plane removed, we can then cluster the remaining points. In this work, we require a minimum of 100 points per cluster, and allow 10cm distance between points within a cluster. Objects such as chairs, tables, desks, and the walls of the room are returned as clusters. Many of these do not correspond to furniture objects, and they are filtered by bounding box size to discard objects that are much too large, much too small, or not near the ground.

C. Alignment

For each candidate partial model M_p, VPOD computes the transformation matrix T that transforms the centroid of the partial model M_p to the centroid of the cluster C it was matched against and apply T_{center} to M_p. This yields the transformed partial model $M'_p = T_{center}M_p$. However, models in a database usually have limited angular resolution, e.g we might have a model of a chair rotated at $10°$ and at $20°$ in the database. The transformation might not align M_p and C optimally. In order to compensate for this, as well as centroid computation errors, Iterative Closet Point (ICP) [2] is performed on M'_p and C. The algorithm assumes that the objects are mostly upright and disqualifies candidate models if the rotation around the ground plane is beyond a threshold. The resulting transformation matrix T_{ICP} is saved. T_{center} and T_{ICP} are now both applied to the full model M_f out of which M_p was generated yielding $M'_f = T_{ICP}T_{center}M_f$. Consequently M'_f represents the full model M_f mapped into world coordinates at the candidate location.

D. Occlusion

Each point in the mapped full model M'_f is checked for occlusion. This is done through a technique similar to ray casting. We adapted ray casting to the property of a laser that the lateral error increases with radial distance. As such, for each point p in M'_f, we check if any point in W lies within a cone originating at the viewpoint origin and facing p. To check this easily, all the points in M'_f and the world W are transformed to radial coordinates around the viewpoint first. The slope of the cone is tuned to match the known angular error of the laser. In order to compensate for noise in the radial distance, we require the occluding point to be a minimum distance in front of the occluded point. In addition, because a model should not occlude itself, we remove C from W prior to the occlusion test. Points that have been determined to be occluded in this way are omitted from M'_f. The resulting model $M^*_f = occlude(M'_f, W)$ is then scored.

E. Scoring

The scoring of M^*_f is done by checking if each point p in M^*_f can be matched against a point in W. A point p in M^*_f is declared to have a match if W has a point within a small sphere around p. The final score is the ratio of matched points to points in M^*_f.

However, different scoring techniques are possible. An additional weighting factor determined by the number of points in M^*_f could be added to capture the lack of evidence that only a small number of points provides. Additionally the scoring could be two-way. This is, ensure that not just M^*_f coincides with W and C but that C also coincides with M^*_f. This would guarantee that most points in C have to actually be accounted for. Details are discussed in V-B.

V. Experiments and Analysis

To verify our approach, we implemented the algorithm and tested it on scans obtained through a Hokuyo laser range finder on a Pan Tilt Unit. Fig. 5 shows an example experimental scene.

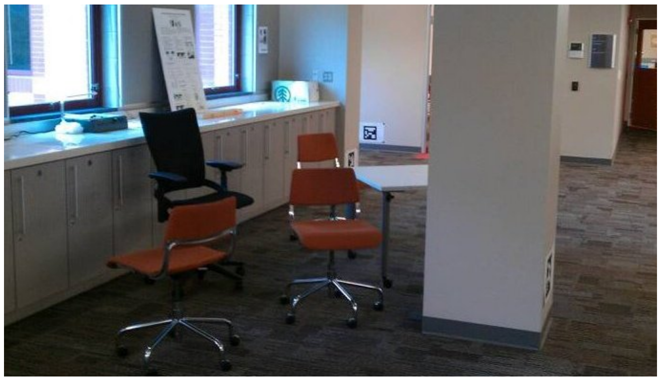
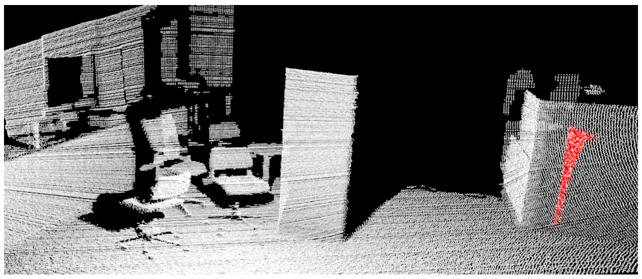


Fig. 5. Experimental setup.


(a) Sample point cloud. Red: cluster classified by VFH as the top of a chair.

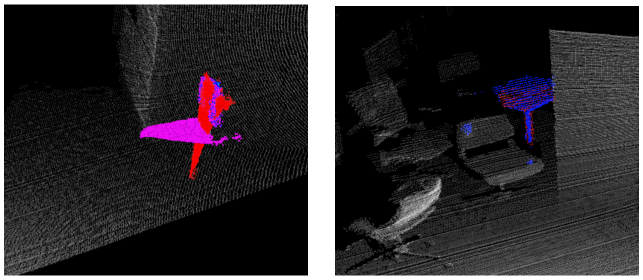
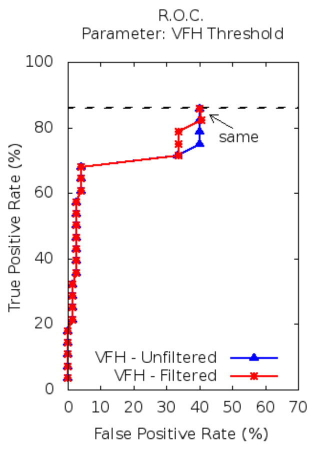

(b) False positive detection. (c) True positive detection.

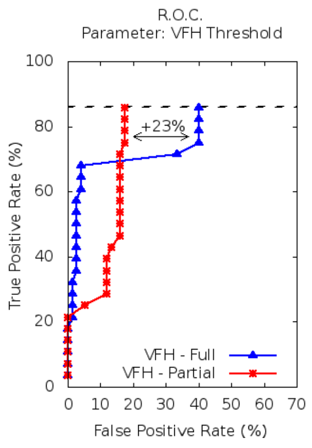
Fig. 6. Example of false positive and true positive detection.

We first created models of a chair and table for 7 distances and 16 rotations from scans with the laser range finder. The distances were chosen such that we have a constant vertical viewpoint change on the obstacle of 9°. For our sensor height of approximately 1.5m, this resulted in distances of 0.86m, 1.09m, 1.35m, 1.66m, 2.06m, 2.59m and 3.37m. For each of the distances, scans of the object with a 22.5° rotational step size were obtained. This resulted in a total of 224 scans of full models. Out of those models, an additional 1068 partial models were created, as described above, this was done by consecutively removing points from bottom to top, left to right, and right to left from the model. We have used a 10cm step size and proceeded until the remaining model had a size between 20-30cm on the axis currently affected by the point removal. Fig. 3 shows an example.

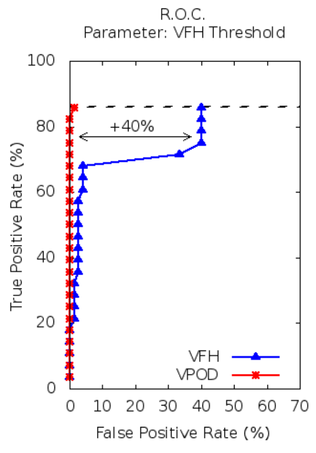
We took 30 test scans of scenes in an office environment with random configurations of up to four chairs of different types and two tables per scan. These scenes had non-furniture items, walls, unoccluded furniture, and partially occluded



(a) Filtering improvement



(b) Partial models improvement



(c) VPOD improvement

Fig. 7. Relative operating characteristic, showing improvement over VFH. Dashed line marks the best possible rate, due to clustering errors.

furniture. For example, some scenes had chairs behind and pushed under a table. An example can be seen in Fig. 6(a). As shown in the results section, these are difficult cases for VFH using full models alone.

When comparing the cluster VFH to the VFH database, a chi-squared distance metric was used. Each cluster from each test scan was compared against all partial and full models for a full evaluation.

A. *Results*

Our algorithm was able to accurately classify every possible cluster in the test scenes, which included previously unseen furniture types (e.g. Fig 5), with a 3.3% of the false positives introduced by VFH with full models alone. VFH with full models did not even classify every partially occluded cluster correctly, while providing many false positives.

Fig. 7(a), Fig. 7(b), and Fig. 7(c) show the behavior of the classification error rates of different algorithms as the VFH threshold is relaxed. The True Positive Rate (TPR) is the ratio of actual furniture correctly identified; The False Positive Rate (FPR) is the ratio of non-furniture incorrectly identified. Due to clustering errors explained below, not all objects in a test scene could be classified by VFH, VPOD, or any other algorithm.

Fig. 7(a) shows the behaviour of VFH with full models on the dataset and the same algorithm with pre-filtering of clusters, as described in section V-C. Even if the VFH threshold is relaxed greatly, allowing *many* false positives, VFH still isn't able to correctly classify many clusters due to partial occlusions. Despite having a more than 40% false positive rate, plain VFH was not able to achieve the maximal possible classification rate on our dataset. Even at a low threshold, the original algorithm still produced a 5% false positive rate while only classifying 70%. Prefiltering provides only a modest reduction in the false positives produced by VFH. Pre-filtering alone does not account for all of the benefits of VPOD.

Fig. 7(b) shows the behaviour of VFH with and without partial models. At very low VFH thresholds, the partial models have no effect. As the threshold increases, the partial models are allowed to incorrectly match clusters, causing an increase in the false positive rate relative to full models alone. But VFH with full models can only match the unoccluded clusters well and must allow many false positives in order to match the partially occluded clusters. However, VFH with partial models is able to classify all test clusters with a 23 point reduction in the FPR. This is because VFH with partial models can match partially occluded clusters using a tighter VFH threshold.

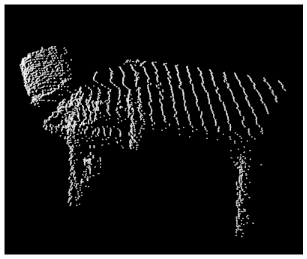
Fig. 7(c) shows the behaviour of VPOD compared to VFH with full models. Fig. 7(c) shows that VPOD, through the use of partial models as in Fig. roc-partial, properly classified *all* test cases. Unlike VFH with partial models, VPOD classified all test cases with only a 1.3% false positive rate, due to the VPOD verification step. This allows VPOD to classify the partially occluded objects without introducing many false positives.

B. *Examples*

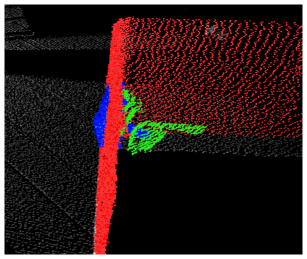
We disabled the pre-filtering and ICP restriction on y-axis rotation to obtain the following examples.

Fig 6(b) demonstrates an example of the false positive detection. VFH classification with partial models classified the cluster marked red in Fig. 6(a) as being the top part of a chair. The algorithm then mapped the full chair into world and scored it. Since the sitting surface is not occluded but also not visible in the scan this classification was rejected. In contrast, the table behind the chairs in Fig. 6(a) was matched by VFH with partial models and verified by the algorithm. Note that VFH with full models alone would not be able to classify this table due to the occlusion.

Fig. 8(b) demonstrates a typical example where our algorithm fails if no pre-filtering or two-way matching is performed. If the cluster is unreasonably large, almost any furniture piece can be fitted in and have its points being accounted for. In Fig. 8(b) a chair (green) is being fitted into a big wall (fitted chair shown in blue). The algorithm is effectively saying that the chair is lodged in the wall. This demonstrates the necessity of pre-filtering or two-way matching in combination with our algorithm.



(a) Example of a chair and table being clustered together.



(b) Example of a small model fitted into a big cluster.

Fig. 8. Examples of where our algorithm fails.

C. Clustering

The clustering distance is currently set to a fixed value. This can yield results similar to Fig. 8(a) where multiple furniture pieces have been clustered together, which hinders classification. In future work we plan to have the clustering distance be data driven. The basic assumption is that points on the same object have a smaller relative distance to each other than points between objects. We therefore plan to evaluate the distances of points within a cluster and re-cluster a cluster based on a new, smaller distance. Clusters that are not classified can be progressively partitioned and reclassified until either a match is found or the cluster size is unreasonably small.

Further, many objects such as chairs and tables can appear as multiple distinct parts due to self-occlusions. For example, when observing most standard office chairs, we typically observe the chair's seat, back, and wheeled base, but not the central supporting column due to the downward viewing angle. Similar results can occur as well with other types of furniture. To address this problem, we plan to project the clusters down to the ground plane, find the convex hull for each, and merge clusters that have overlap. This allows us to consider such objects as one unit, despite being separated by a significant vertical distance. Again, this step can be verified by checking if the classification results have improved in comparison to the single clusters.

D. Runtime

We are performing the occlusion and matching simultaneously in VPOD. As such the occlusion and matching together takes an average of 3.3 seconds for a world scan with about 32,000 points and an average cluster size of 19,000 points. If runtime is a concern the occluding and matching do not need to be run against the full point cloud. Rather, it can easily be determined which parts of the world are affecting the current occlusion and matching operation and the operation be performed against a subset of the world. The additional runtime can be justified in domains that call for the greater classification accuracy provided.

VI. Conclusion

In this paper we presented the VPOD algorithm for detecting partially occluded objects by matching clusters against segments of models and verifying our expectations against the world. VPOD extends the scope of VFH classification to the core idea of using partial models. To reject false positives, VPOD verifies expectations about the predicted classification with the world. We verified the effectiveness of our approach on real data, showing improvement on cases not handled by VFH with full models alone.

We are currently working on improving the clustering algorithm as described above. In addition, we are investigating techniques of enhancing the clustering through feedback from the algorithm. It is possible to eliminate points from a cluster which correspond with points in a matching model and rerun the remaining cluster. For example the cluster visualized in Fig. 8(a) was actually classified as a table by our algorithm, the table could then be removed from the cluster and the remaining cluster classified as a chair.

Further, we are currently integrating our algorithm into semantic mapping and NAMO on our robot.

ROS was used for most parts of this work and the code developed for this project will be made open source.

References

[1] Kurt Konolige Wolfram Burgard Bastian Steder, Radu Rusu. Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries. *IEEE International Conference on Robotics and Automation*, 2011.

[2] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[3] Nico Blodow, Zoltan Marton, Alina Soos, and Michael Beetz. Towards 3D object maps for autonomous household robots. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3191–3198, October 2007.

[4] Dirk Holz, Ruwen Schnabel, David Droeschel, Jörg Stückler, and Sven Behnke. Towards Semantic Scene Analysis with Time-of-Flight Cameras. Number June, 2010.

[5] A.E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 2002.

[6] Zoltan-Csaba Marton, Dejan Pangercic, Radu Bogdan Rusu, Andreas Holzbach, and Michael Beetz. Hierarchical object geometric categorization and appearance classification for mobile manipulation. In *Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010. Accepted for publication.

[7] O.M. Mozos, Z.-C. Marton, and M. Beetz. Furniture models learned from the www. *Robotics Automation Magazine, IEEE*, 18(2):22 –32, june 2011.

[8] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Andreas Holzbach, and Michael Beetz. Model-based and learned semantic object labeling in 3d point cloud maps of kitchen environments. In *The 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 10/2009 2009.

[9] RB Rusu, Gary Bradski, Romain Thibaux, John Hsu, and W Garage. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. *willowgarage.com*, 2010.

[10] Bastian Steder, G Grisetti, M Van Loock, and W. Robust on-line model-based object detection from range images. *Intelligent Robots and*, pages 4739–4744, October 2009.

[11] Hai-ning Wu, Martin Levihn, and Mike Stilman. Navigation Among Movable Obstacles in Unknown Environments. *IEEE International Conference on Intelligent Robots and Systems*, 2010.