Viewpoint based Mobile Robotic Exploration aiding Object Search in Indoor Environment

Karthik Desingh*
Robotics Research Center
IIIT Hyderabad, India
desinghkar@gmail.com

Akhil Nagariya
Robotics Research Center
IIIT Hyderabad, India
akhil.nagariya@gmail.com

K Madhava Krishna
Robotics Research Center
IIIT Hyderabad, India
mkrishna@iiit.ac.in

ABSTRACT

We present a probabilistic method of finding the next best viewpoint that maximizes the chances of finding an object in a known environment for an indoor mobile robot. We make use of the information that is available to a robot in the form of potential locations to search for an object. Extraction of these potential locations and their representation for exploration is explained. This work primarily focuses on placing the robot at its best location in the environment to detect, recognize an object and hence do object search. With experiments done on the exploration, object recognition individually we show the robustness of this approach for object search task. We analyse and compare our method with two other strategies for localizing the object empirically and show unequivocally that the strategy based on the probabilistic formalism in general performs better than the other two.

Keywords

Segmentation, Scene understanding, Object recognition and classification, Robotics

1. INTRODUCTION

The arrival of personal robotics as a prominent research and application domain [1] along with the advent of RGB-D cameras like Microsoft Kinect provides for renewed interest in object recognition based on point cloud datasets as applied to personal robotic context. For example a robot operating in an indoor home or office setting is often entailed to look for, identify, fetch and transport objects of interest. This paper provides a formulation and an implementation consequent of the formulation for object search in a known indoor environment.

We consider a robotic agent equipped with depth camera such as a Kinect, required to localize an object in a known workspace in as fast a time as possible. The description of the workspace consists of both its metric (occupancy grid) and semantic (presence of tables, cupboards) components. The robot's motion involves moving from one location to another in search of the object as well as panning of the camera to unambiguously localize the object. Main contributions of this work are of two folds, one is to provide an exploration strategy for the robot moving in indoor environments to localize objects and the second is to give a formalism to find the best view to localize an object at the same time reducing the effort during the search. We have not found any such formalism or exploration strategy available for the object localization tasks to the best of our knowledge.

The object recognition is accomplished through state of the art modules [7] and the objects recognized include cups, water bottles and etc. The method has been empirically verified over several runs on the modified version of Turtlebot equipped with Kinect and baseline localization modules. A comparative tabulation over various variants of the proposed search method vindicates its accuracy. The variants include search by adapted frontier based exploration [15] and viewpoint only search. These variants are explained in Section 4. Some of the assumptions that are made in this work are confined to object recognition implementation. They are: Objects used are not reflective or transparent, are not cluttered, are small in size and only 3D shape context is used in recognizing them.

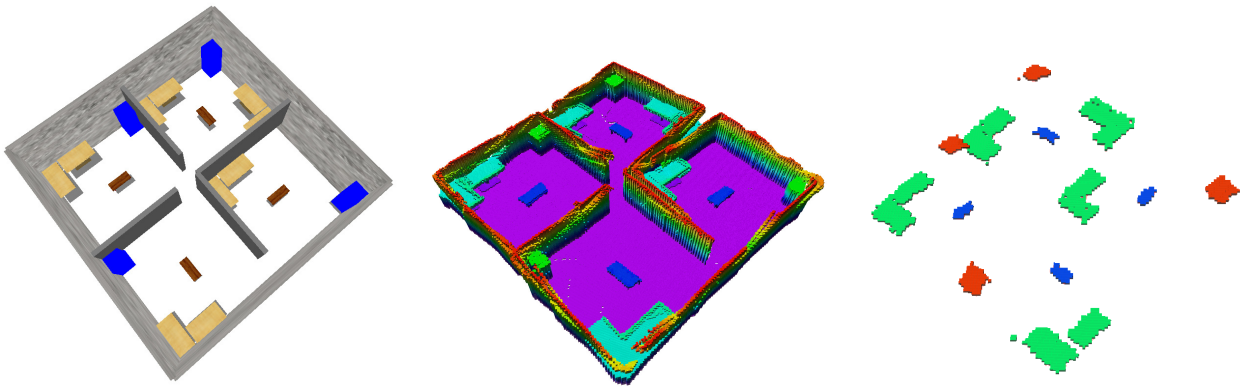
Despite these assumptions, the object recognition by robot in search, grasping and manipulating tasks prone to fail, as not every location the robot moves into will provide a good recognition setting. In this work we try to solve this problem of positioning the robot in an indoor environment so as to support the 3D object segmentation and recognition techniques. We chose object search to be the application where our exploration technique can be used efficiently.

2. LITERATURE REVIEW

A paper highlighting the theme of using robots for object localization is most vividly described in the active exploration formalism proposed in [13]. The method was not so much about object search but rather about maximizing object detection during robot traversal between two locations. Moreover the experimental verification of the method was very limited. In [10] a mechanism for object search integrating semantic understanding was proposed and demonstrated for the PR2 robot from Willow Garage. While the current method has commonalities with [10] in terms of using the semantic map to prune the search it differs prominently in the way the next best view for the object is computed inte-

*Corresponding author

ICVGIP '12, December 16-19, 2012, Mumbai, India



(a) (b) (c)

Figure 1: a) Environment b) 3D Occupancy map c) Filtered semantic locations

grating the semantic understanding. Specifically the current paper provides a formalism for deciding whether the robot should continue to search the current semantic construct (a table) for an object (a cup) or move to another semantic construct (a different table) to maximize its probability of localizing the object. This formalism is not dealt with or finds mention in [10]. While others such as [7] and [12] have focused on object recognition and semantic labelling from point cloud data that can be used for an object search formalism. Outside of these methods there exists a plethora of literature on object recognition based on modern machine learning techniques such as [11] [5] that are beyond the scope of this effort. Exploration for object search is a very new theme in the robotic community. We begin the paper with the motivation of understanding of the environment for efficient search and then explain the formalism in detail along with experimental results on exploration and object recognition followed by object search task.

3. POTENTIAL MAP GENERATION

Motivation for this work comes from the fact that every object in an indoor environment is associated with a container or a location for which it is purposefully built for. For example Coffee Mug is more likely to be on tables and a Vase is more likely to be on Flower stands. In order to make use of this object and its container association, efficient mapping techniques along with semantic understanding is vital. With the advent of low cost 3D sensor, 3D mapping of the environment is possible using octomap mapping package [14]. In this section we explain how to create the representations of these locations. 3D occupancy map can be interpreted as point cloud with each 3D point representing a voxel. By filtering this point cloud based on the knowledge of the locations like height from ground, area of the container and other metric information, locations like desks, table and other containers can be segmented. Fig. 1 shows how this process looks for an environment where we filtered three object containers based on their height and area from the 3D Occupancy map. High level understanding of the objects and its possible potential locations in the indoor environment is done by previously by Rusu [9]. In this work we create a one to many mapping of the objects and its potential semantic locations. Depending on the object in search, the potential locations for exploration decreases from the complete 3D map to a set of locations. This idea of object-location association is shown in the Fig. 2. This representation is however generated with manual intervention and used by the robot for its search task.

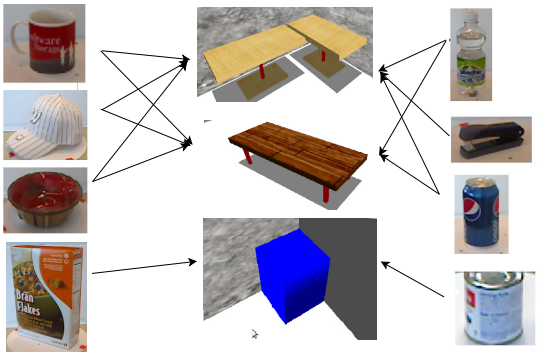


Figure 2: Objects and Potential locations association

4. SEARCH DRIVEN EXPLORATION

Once the potential locations are known based on the object in search, exploration on these potential locations is the primary task. For a mobile robot, a 2D map with obstacle information in the form of occupancy cells is sufficient for its navigation. Hence the potential locations as mentioned in Section 3 are projected on to a 2D plane along with the occupancy details. These maps are shown in Fig. 3. We call this representation of potential locations and occupancy values of the environment as Potential Location Map which is right part of Fig. 3. The grey regions are unexplored, black regions are occupied, white regions are non-occupied and green regions are potential locations. In order to explain the exploration part lets assume that the object in search is a Coffee Mug and potential locations to be explored are Tables from here. Having known the locations to be explored, the exploration algorithm should be able to cover the table area starting from a location in the map. State of the art algorithm for exploration till date in robotics field is Frontier exploration [15]. Frontier is the boundary between

the explored and unexplored region. This method looks for frontier locations in the map and decide the closest frontier to visit at every iteration untill all the frontiers are covered. We adapted this method to our known environment setting where the borders of the tables become the frontiers. Main drawback of this method for our problem is that, the robot location as a result of the next closest frontiers is not most likely to have the object in view. Hence we move on to the viewpoint based exploration. Terminology involved in the formalism and their practical implementation issues are explained in the following subsections one by one.

Figure 3: Occupancy Grid map and Potential Location map for object search

4.1 Viewpoint

Viewpoint is a location in the map (x, y, z) along with an orientation (quaternion). It can be defined differently for various sensors. For Kinect, it is a location in the map from which the view frustum is able to see the scene at an orientation. Best viewpoint is the location from where the view of the potential locations of the map is maximum. Computing the viewpoint strengths as ratio of the area of the table seen for every (x, y) in a map at some orientation to the total area of the table. This gives values as seen in the Fig. 4(c) with values represented by the color intensity as mentioned in Fig. 4(b). For example if the robot is at 0 degrees orientation then viewpoint strengths computed at all the locations in the map having robot posed at this orientation will result like 0 degrees image of Fig. 4(c). In the same way, other images show the viewpoint strengths at their respective orientations. Fusing all these strengths to find the over all best viewpoint results in the Fig. 4(b). These viewpoint strengths are independent of the robot's location at any instant. When robot is searching for an object and is exploring the potential location map, the next best viewpoint at an instant should depend on the explored regions and location of the robot at that instant of time. For example as shown in the Fig. 5, though the best viewpoint as a whole from Fig. 4(b) is at A irrespective of robot R's starting/current pose, the one that is nearer at C is more appropriate location where the robot has to look for the object at next instant. In the other way if the strength of the location B (view covering two tables) is higher than C (view covering single table) then robot has to choose B which is not as far as A. This ambiguity in choosing the next best viewpoint with respect to its location and viewpoint strengths is solved in this work with a probabilistic formalism.

4.2 Probabilistic Framework

Given a known workspace (potential location map) W and an existence of an object O somewhere in W, robot finds the

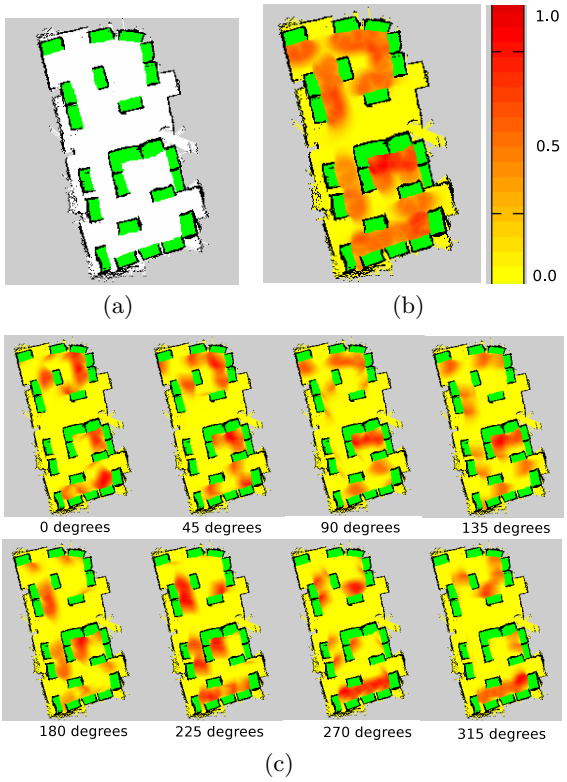


Figure 4: a) Potential location map b) Best viewpoint locations c) Viewpoint strength for 8 orientations

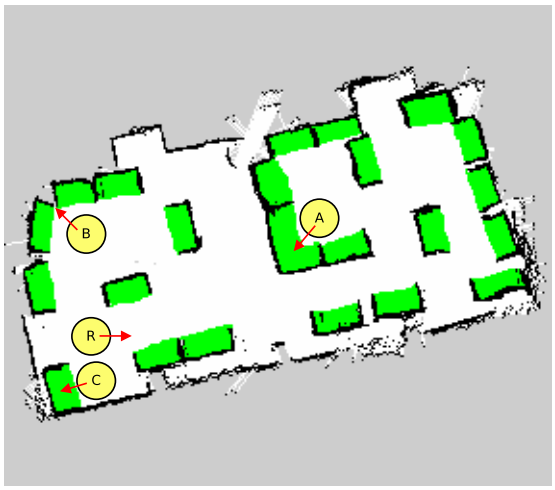


Figure 5: Viewpoints in a map based on robot location

path that reduces the time taken to find the object O. We assume O to be on top of tables T_i in W, which decides the Potential Location map discussed in earlier section.

Let S_o be a random variable whose values are sighting or non sighting of the object. $P(S_o)$ represents the probability that the object is sighted, while $\overline{P(S_o)}$ represents the probability of not sighting it.

Let $P(S_o, V_t)$ be the anticipated probability of sighting an

object in the next view V_t. Where subscript t is the time instant. In general we want to find that view V_t for which $P(S_o, V_t)$ is maximized. In other words

$$V_t = argmax_{Vt} P(S_o, V_t) = argmax_{Vt} P(S_o/V_t) P(V_t)$$

$$= argmax_{Vt} P(S_o/V_t) P(V_t/t_v) P(t_v)$$

The last part of the last term brings in tacitly the time factor $P(t_v)$, the probability of reaching the viewpoint V to obtain the view V_t. It simply says that the probability of obtaining a view V_t is conditioned on reaching the viewpoint V to obtain the view and the robot would choose to reach a viewpoint V with a probability $P(t_v)$ that varies inversely as the time taken to reach V. $P(S_o/V_t) = A(V_t)/A(tables)$ where $A(V_t)$ is the area of table seen in V_t and $A(tables)$ is the total area of all the tables. Essentially V_t is the view that maximizes the chance of seeing as much area as possible while minimizing the time to reach that as we define $P(t_v) = 1/t$ since $P(V_t/t_v) = 1$ as a view V_t is always possible from a viewpoint V.

The general formulation would want to maximize $P(S_o, V_t/V_{t-1}, ...V_1)$. Considering two views V_t, V_{t-1}, which can eventually be generalized, the view $V_t = argmax_{Vt} P(S_o, V_t/V_t, V_{t-1})$ simplifies to after some steps as $V_t = argmax_{Vt} P(S_o, V_t/V_{t-1})/t$.

$$P(S_o, V_t/V_{t-1}) = NewA/(TotalA - PrevA) \qquad (1)$$

where $NewA$ is the new area in V_t, $PrevA$ is the area seen till V_{t-1} and $TotalA$ is the total area of the tables in the environment. Computation of table area is explained in the next subsection.

4.3 Area of the table

Above framework talks about the area of table seen at every viewpoint. Here we explain on how the area of the table is estimated from the Kinect point cloud in practical scenarios. At every viewpoint before the computation of table area, the object recognition module is requested to check if the object in search is seen or not. Table area is calculated only when the recognition module responds negatively. We may encounter two cases of table views in practical scenarios which are 1)With objects on the table and 2)Without objects on the table. In either case the computation of the table is same. Extracting planes and trying to fit a convex hull to calculate area of the table may not always work. Hence we take the point cloud of the view from kinect, filter it for the table locations and then voxelize the point cloud to a leaf size. Now counting the voxels will directly give the area of the table. This is not the exact area of the table but it is the rough estimation of the area of the table which works fine if the leaf size is same as the scale of the occupancy map generated. Fig. 6 shows how the point cloud in the current view is filtered to get the plane of the table which is later voxelised to give the area of the table for both the cases.

5. OBJECT RECOGNITION

Object recognition module is invoked at every viewpoint during exploration. Various descriptors are available for the 3D recognition like PFH (Point Feature Histogram), FPFH (Fast PFH) [6] and VFH (Viewpoint Feature Histogram) [7]. We make use of VFH as it is fast compared to other

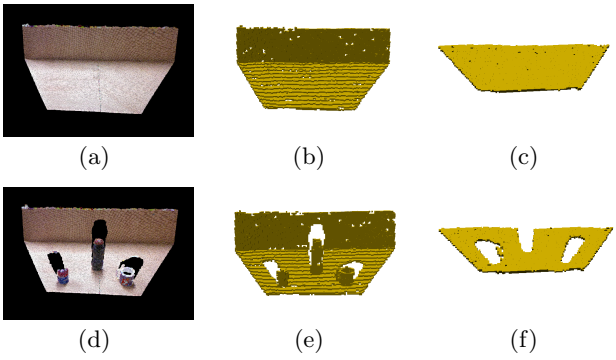


Figure 6: Table area estimation for table views with objects and without objects

histogram based descriptors and gives the signature of the whole object from a viewpoint rather than computing descriptors of keypoints in a cloud and matching them at every recognition call. Hence every sample of cloud in training is associated with a VFH signature on basis of which the training is done. The following subsections talk about the training and testing phases of the recognition module. We make use of the implementations available at PCL (PointCloud Library) [8].

5.1 Training

Supervised learning is done with well cropped point clouds of 8 objects shown in the Fig. 7. Viewpoint feature histogram (VFH) is the 3D descriptor used to distinguish the shape characteristics of these objects. For the purpose of real time recognition we use the implementation of fast approximate K-Nearest Neighbors (K-NN) from the FLANN library [4]. Size of the object in the viewed scene is very less and an extra information of that lying on top of the table is known to us a priori. Hence in the testing phase objects are segmented out of the entire cloud and then subjected to the testing with the trained models. Objects used in training were placed at different orientations and positions on table from Kinect at a distance of > 0.60 m. Depending on the geometric shape of the object, the orientations and positions were changed to capture all possible views. 3D object recognition requires maximum training samples as the active projection mechanism of the Kinect like sensors prune to be noisy for smaller objects. Also number of training clouds are different for different objects based on their geometric symmetry. For example cap is less symmetry in shape compared to water bottle and requires many training samples as shown in Table. 1. Totally 1581 number VFH signatures are clustered and kd-tree structure of the same is built for easy traversal in the testing phase.

5.2 Testing

Since viewpoints are the locations where the probability of viewing the tables is high, the viewed scene is expected to have table in it. Plane segmentation is done using SAmple Consensus model which is available in PCL [8]. Once the plane segmentation is done we filter the planes which are not horizontal and the largest supporting plane which is the tabletop (potential location) is extracted. Now the entire cloud on top of the table is segmented using the fact

Figure 7: Objects for training

Table 1: Training details

Objects	No of training clouds
Water Bottle	168
Cap	230
Coffee Mug	261
Cube	168
Glue Bottle	145
Tetrapack	131
Mug	303
Soda Can	175

that they lie on top of this plane. This entire cloud will not always be a single object. Hence the cloud is subjected to standard Euclidean clustering algorithm to partition the cloud into cluster representations of the individual objects. Fig. 8 shows the pipeline of segmentation from Kinect seen view to the individual clusters. This technique of segmenting the objects in point clouds is being used effectively in 3D perception area [2].

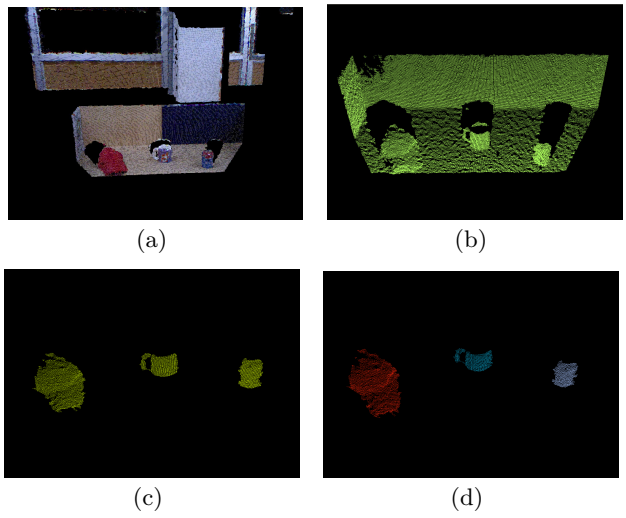


Figure 8: Segmentation process a) Entire scene cloud b) Filtered cloud c) Tabletop cloud d) Clustered objects cloud

Each cluster obtained from the above segmentation is tested against the trained models based on K-Nearest Neighbour. To decide on the K value for the K-NN classifier we have tested the object classification performance on the real time Kinect data for the 8 objects. This is done by plotting the Precision and Recall curve with varying K and the highest K that maintains high precision >0.75 for all the objects is chosen for the object search task. Training and testing samples are exclusive of each other as the training is done on the dataset collected and testing is done on the real time data with point clouds from Kinect at 30Hz.

6. EXPERIMENTATION

We have experimented on the above sections separately and integrated them to perform the object search task. Modified Turtlebot mounted with Kinect sensor as shown in Fig. 13 is the robotic platform used for all the experiments.

6.1 Exploration

In viewpoint based exploration we primarily focus on the following two aspects. a) Exploration module should compute the best viewpoint based on the proposed probabilistic formalism and give it to the robot as a goal position for its navigation. Exploration should be smooth and continuous based on the locations with minimal effort involved. b) Every viewpoint the robot is given, should have the view of table (potential location) for the recognition to work.

We have tested the performance of the proposed exploration strategy on both simulation and real environments. Fig. 9 shows the potential maps of the environments involved in the experimentation. Fig. 9(a) 9(b) 9(c) are real environmental maps and Fig. 9(d) 9(e) 9(f) are simulated maps generated using 3D simulator Gazebo [3]. Now each of these maps are tested with three different starting locations marked as A, B and C. Evaluation of the proposed probabilistic method (Viewpoint with time) is done comparing it with the adapted frontier based exploration and a method that decides the next view based on maximization of viewpoint alone without invoking the distance criterion (Viewpoint only method). Every map has three timings given by three strategies for complete exploration of the table regions (potential locations). Time taken by these strategies to explore all tables are plotted in Fig. 9 with their corresponding maps and robot locations. It can be inferred that the proposed method clearly outperforms the viewpoint only method in all environments. Occasionally at some environments like Fig. 9(e) frontier exploration is comparable with the proposed method. It is to be noted that adapted frontier based exploration looks for the unexplored table area and its framework lags to support viewpoint based approach in maximizing the view of the table regions. Paths traced by the exploration methods are plotted in the Fig. 10. Adaptive frontier based exploration moves towards the frontier regions which overlap with the tables as shown in Fig. 10(a). Viewpoint only method swings around the map from one viewpoint to other in the process as shown in Fig. 10(b). Path lengths of Fig. 10(a) and Fig. 10(c) look similar. However adapted frontier based exploration spends more time to scan the table area, whereas the proposed method has the ability to get the complete area at single shot. Because of its inability to maximize the viewpoint, the adapted frontier method is not supportive to the object search applications. From the plots on the timings and paths traversed by the robot, it can be concluded that the proposed method performs exploration quickly, also considering the view of the table for object search task.

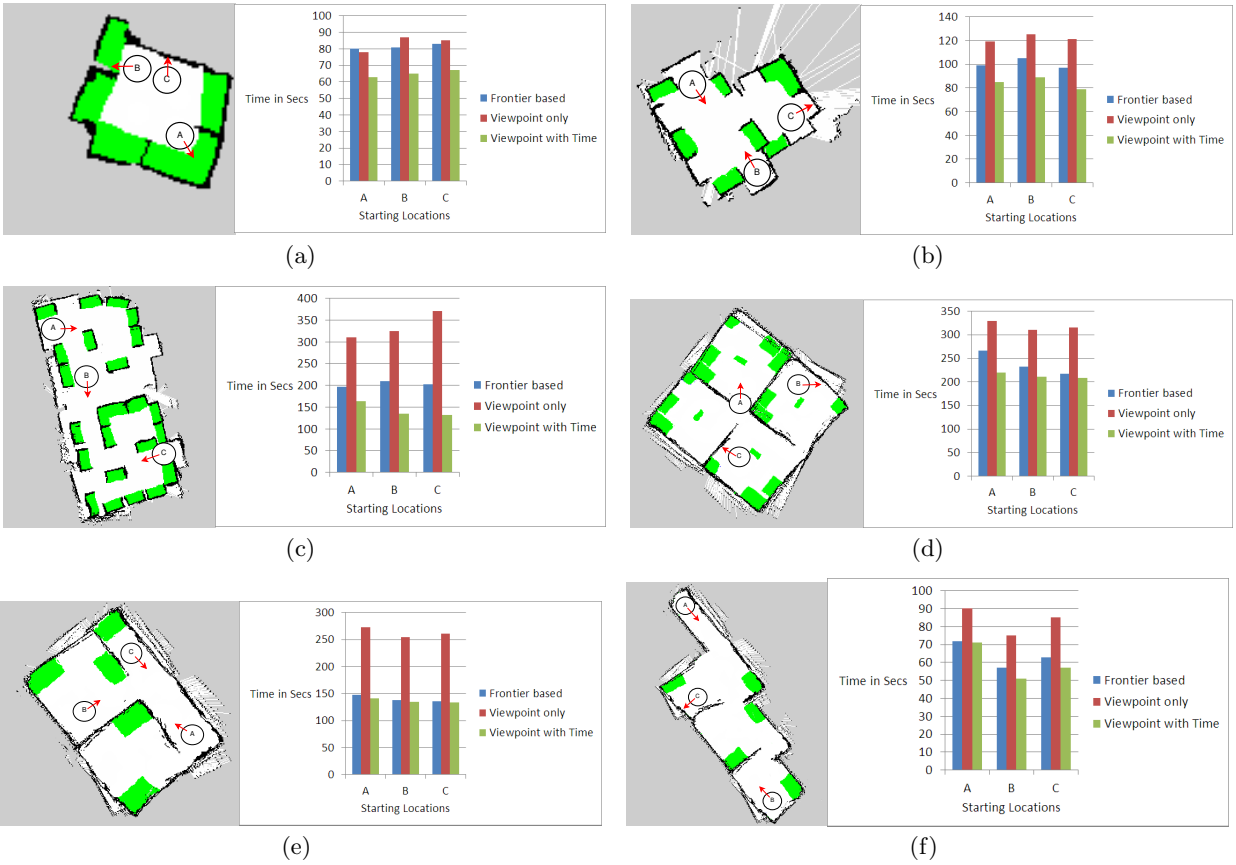


Figure 9: Graphs plotted for time taken by different methods for different maps

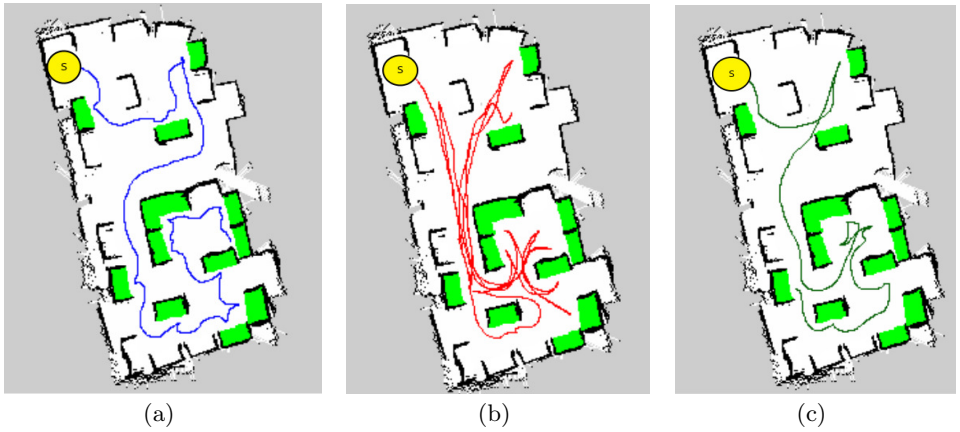


Figure 10: a) Adapted frontier based approach b) Viewpoint only c) Viewpoint with time constraint (Proposed method)

6.2 Object Recognition

3D object recognition is dependent on the performance of the segmentation of the object from the entire scene cloud. Having evaluated the exploration strategy where the table top is always available to segment out object, we can consider that the segmentation performance is favoured at all locations. In the testing phase of the object recognition we have analysed the performance of the various objects based on their precision values to choose a threshold K that can be used in the search task. K computed out of the analysis is 12. Fig. 11 has the precision and recall curve for all the 8 objects tested. Though the contribution of the paper lies in the exploration for object search, we are evaluating the performance of the object recognition with VFH descriptor for our objects mentioned above. We notice that not all the frames from the Kinect sensor gives the complete shape description of an object due to the noise in the IR pattern

reception. It is not certain to use just a single frame of the cloud for real time recognition. We add multiple frames and then sample it to remove this uncertainty.

6.3 Object Search

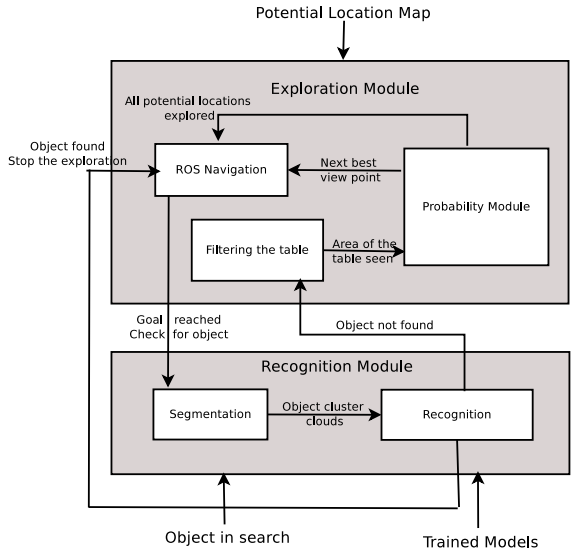


Figure 12: System Architecture

Object search task is performed by integrating the exploration and object recognition modules. Modules have to communicated with each other with messages, requests and responses. High level structure of this integration is shown in a system architecture Fig. 12. Exploration module requires the Potential Location map as an input. It gets the next best viewpoints based on the map and current location of the robot from the probability module. ROS navigation stack navigates the robot to move towards this viewpoint. Once goal is reached the navigation stack requests the recognition module to check if the object is found. Recognition module performs the required segmentation to get the objects in the scene and check if the object in search is found. If the recognition module responds positively, then the exploration stops, else the area of the table in view is computed and given to probability module to get the next viewpoint for the search. This is continued till all the potential locations are explored. Images of the robot at some viewpoints with objects in view while performing object search based exploration is shown in the Fig. 13.

7. CONCLUSION AND FUTURE WORK

We proposed a probabilistic method of exploration for object search based on viewpoint, that eventually maximizes the chances of finding the object. This is first such work where autonomous exploration driven object search is solved by a formal procedure. We demonstrated that the proposed method took least time to explore potential location where object recognition is favoured. We evaluated the exploration and recognition thoroughly considering the practical issues and presented the statistics substantiating our claims. We have integrated the modules and solved the object search problem. Future work aims at improving the object recognition and exploration based on the image cues in addition to the 3D cues to work on real cluttered environments.

8. ACKNOWLEDGMENTS

We would like to thank the Dept of Information Technology to have funded this work through the grants made available by the National Program on Perception Engineering - Phase 2

9. REFERENCES

[1] J. Bohren, R. Rusu, E. Gil Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the pr2. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5568–5575. IEEE, 2011.

[2] M. Gupta and G. S. Sukhatme. Interactive perception in clutter. In *Robotics: Science and Systems*, Jul 2012.

[3] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[4] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISSAPPâĂŹ09)*, pages 331–340, 2009.

[5] O. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1427–1434. IEEE, 2011.

[6] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[7] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155–2162. IEEE, 2010.

[8] R. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[9] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[10] M. Saito, H. Chen, K. Okada, M. Inaba, L. Kunze, and M. Beetz. Semantic object search in large-scale indoor environments.

[11] J. Shotton, A. Blake, and R. Cipolla. Efficiently combining contour and texture cues for object recognition. In *British Machine Vision Conference*, 2008.

[12] H. Swetha Koppula, A. Anand, T. Joachims, and A. Saxena. Labeling 3d scenes for personal assistant robots. 2011.

[13] J. Velez, G. Hemann, A. Huang, I. Posner, and N. Roy. Active exploration for robust object detection.

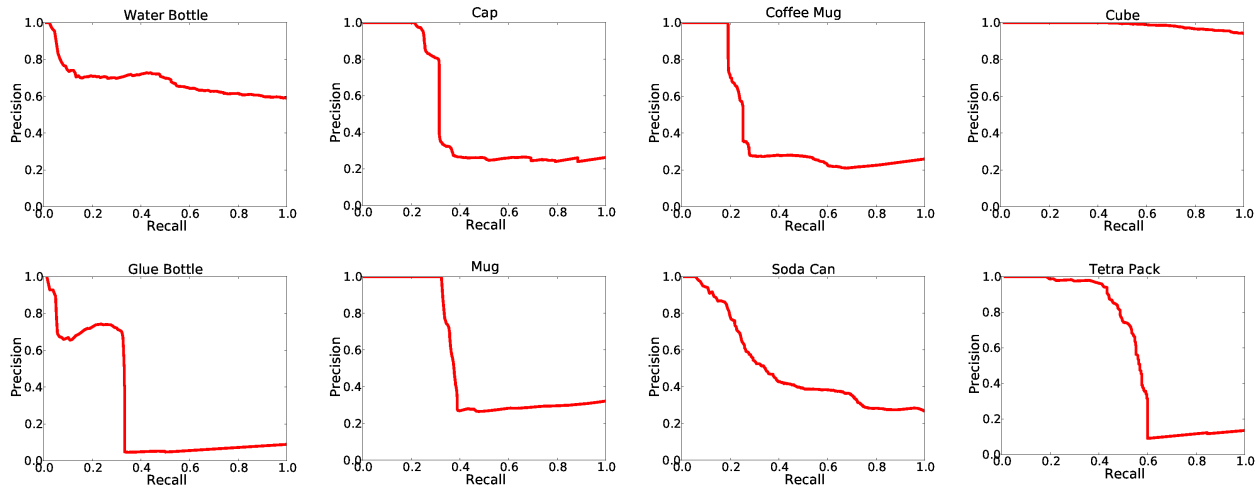


Figure 11: PR curves for 8 objects used in the experiments

Figure 13: Screenprints of the robot performing object search based exploration: Potential map (from left), Robot in the environment at a viewpoint, Scene captured at the viewpoint, Object segmented and recognized

In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[14] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[15] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.