Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 13-025

View Planning For Cloud-Based Active Object Recognition

Gabriel Oliveira and Volkan Isler

September 12, 2013

View Planning For Cloud-Based Active Object Recognition

Gabriel L. Oliveira Volkan Isler

Abstract— **One of the central problems in computer vision and robotics is to recognize objects in a scene. State-of-the-art algorithms for object recognition are extremely data intensive. Cloud technologies hold the promise to make such algorithms available to robots with limited computation capabilities. However, collecting and transferring large amounts of data with such robots remains a challenge. In this work, we investigate the possibility of enabling cloud-based object recognition by carefully planning the robot's viewpoints. While view planning techniques for object recognition exist, such techniques are too costly to be executed by robots with limited capabilities. This is unfortunate because these are the robots which would benefit most from cloud-based techniques.**

In this paper, we present evidence for the existence of universal viewpoints: a small number of viewpoints which guarantee accurate object recognition regardless of the object pose. Our experiments with real data show that such viewpoints exist for common objects. Hence, view-planning for object recognition can be performed in an open-loop fashion without the need for running costly algorithms on small robots.

I. INTRODUCTION

Cloud technologies hold the promise to relieve robots from data and computation limitations. With access to the cloud, a simple robot with limited capabilities can have access to state-of the algorithms for countless tasks such as language understanding, scene and object recognition and so on. However, before this technology makes a real impact, issues such as latency must be addressed.

The focus of this work is object recognition which plays a central role in many robotics perception tasks. We envision a simple robot connected to an object recognition server on the cloud. The robot is equipped with an RGB-D camera such as Microsoft Kinect. As we will see shortly, the viewpoint where the object is observed has a significant impact on object recognition performance. On the other hand, transferring a large number of views can easily overwhelm both the robot and the network. Hence, we focus on the natural question: Can we perform accurate object recognition with only a few viewpoints?

A common approach for viewpoint generation is the Next Best View (NBV) approach [9], [11], [4]. These approaches take a partially generated object model and decide the next viewpoint for observing it. Next best view approaches require reasoning about occlusions and often require model reconstruction. Hence they are computationally intensive, and therefore not appropriate for simple robots with limited computation capabilities.

In this work, we present an alternative approach. We seek a *Universal Viewpoint Set (UVS)* which we define as a set of viewpoints that guarantee high-quality object recognition regardless of the object and its pose. A UVS, if it exists, would allow for open-loop viewpoint planning and minimize the effort spent for planning views. Of course, to be useful a UVS has to be small. Hence the main question is whether a small cardinality UVS exists for object recognition.

Our main contribution is to show that Universal Viewpoint Sets exist for common objects. We take an empirical approach where we start with a database of common objects. After studying the effect of viewpoint selection on object recognition, we present a comparison of various view configurations. We present a configuration which provides high recognition performance based that the server can recognize it in the first place. This in turn yields a practical view planning strategy which can be executed by any robot capable of localization.

The paper is organized as follows: After discussing related work in the next section, in Section III we present our methodology to recognize objects based on shape information. Section IV describes the system including details about the dataset, the robot setup and a discussion on performance related issues. Single view results are described in Section V. Section VI introduces the proposed view planning approach. Section VII presents a summary of our results and reports future research directions.

II. RELATED WORK

Our work is related to view planning and the emerging field of cloud robotics. Several cloud-robotics approaches have been proposed in recent years. Some of these approaches focus on information sharing and learning adaptation [15], [7], [3], [10] while others use the computational power of the cloud for manipulation or recognition [6], [2].

The work of Waibel et al. [15] outlines potential gains of using the cloud for robotics. Their project aims to provide a centralized knowledge database for sharing data between robots. They also provide one the earliest demonstration of real applicability of this technology for robot tasks such as manipulation and navigation.

Ben Kehoe et al. [6], present one of the first cloud-based object recognition and grasping architectures. This work merges both of the main advantages of cloud systems: computation and storage. The system uses Google Goggles object recognition engine (computation) and Google cloud storage to store $3D$ object models for manipulation. Three dimensional data is employed at the manipulation module whereas object-recognition is image-based. One of the differences between our work and this paper is that we employ a three dimensional object recognition server.

Cloud robotics for assistive manipulation has been studied in [2]. This work implements cloud-based manipulation techniques to make it possible for a quadriplegic person

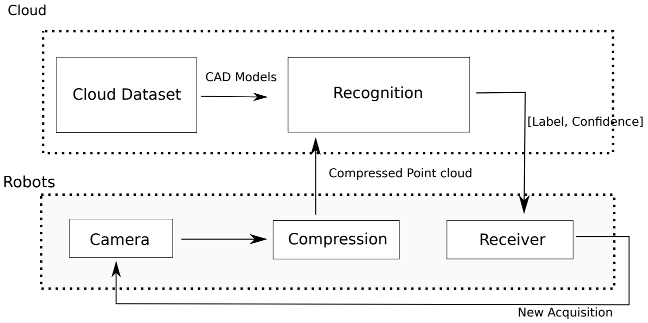

Fig. 1. System Overview. The Proposed system is divided into Robot and Cloud components. The Robot is responsible for acquiring data from the camera, possibly compressing it and sending it to the cloud. The cloud is in charge of object recognition.

manipulate objects and perform daily tasks such as shaving or scratching.

While some studies focus on the computational aspect of cloud services, others demonstrate the use of cloud to store, manipulate and share vast amounts of data. Lai et al. [7] take advantage of online datasets to reduce the need for manually labeled training data. Ben Kehoe et al. [6] use the cloud to store grasp information about objects which is hard to maintain locally due to size. Additionally, some works also demonstrate the information sharing ability between networked robots to accomplish tasks in different periods of time and space [3], [10].

There is also substantial work on view planning. The seminal paper by Maver and Bajcsy [9] proposes a view planning approach grounded in exploration of occluded areas for next view selection. We refer the reader to the surveys [1], [13], [14] for an overview of related work. Our work differs from this line of work in philosophy: Rather than a closed-loop approach in which significant memory and processing is required to compute partial models, we seek open-loop view planning techniques.

III. Object Recognition Methodology

In this work, we target systems with two components: a robot which acts as the client and the cloud-server. The robot obtains one or more views of an object and sends it to the server. It is possible that some local computation such as data compression is done on the robot to improve system performance. Figure 1 presents an overview of the system. In this section, we present an overview of the object recognition methodology employed in this work.

The object recognition pipeline is composed of an off-line and online stages. The off-line stage or training step is responsible for generating partial views of the CAD models.

A. Training

The cloud recognition training module generates n partial views of each CAD model in the training dataset, using the Visualization Toolkit (Vtk) [12]. The training process that creates these partial views, here defined as training viewpoint vector V, consists of rendering and sampling the z-buffer from views around the objects. The number of views is chosen empirically, since these values can range from a couple dozens to hundreds, based on the application and memory availability. We follow the work of [16], [17] and we fix the number of views to 80 to provide enough information for a wide range of objects.

After generating these partial views, geometric descriptors are computed. A geometric signature is a descriptor that incorporates shape metrics, like relationships between points, lines and planes, to describe an input set.

B. Extracted Features

The class of descriptors used in the system is $3D$ shape-only global descriptors, specifically the Ensemble of shape functions (ESF) descriptor [17]. Shape descriptors rely only on $3D$ dimensional information that provides invariance to light conditions and robustness to slight viewpoint changes. The ESF descriptor requires segmenting the scene, in order to find the probable object area, because it creates a signature of all the provided points. The computed descriptor merges three shape functions, $D2$, $A3$ and $D3$. The first metric ($D2$) is the distance between a randomly chosen pair of points. The second metric encodes the angle between two lines ($A3$). Each line connects two of the three randomly created points on the partial model. The last shape function computes the triangle area of three randomly selected points ($D3$). It can be considered an extension of $D2$, since it extends the previous shape function in one dimension. The final part of the descriptor defined by R is the ratio of the line distances.

We performed tests to identify characteristics of these descriptors. In Figure 2, we present the histograms of the three features for the airplane model and the mug model from two arbitrary viewpoints. As the figure shows, the distribution $D2$ features are very similar. $A3$ and $D3$ are more discriminative than $D2$.

Another observation was that some classes are inherently hard to discriminate between. Figure 3 shows two histograms of Toilet paper and Mug models. Both histograms are almost identical. This is true for all viewpoint of these two objects. Therefore, without using top views, recognition is likely to fail even if we provide a very large number of viewpoints.

After training, the online recognition module can be executed.

Once the robot obtains a point cloud, it sends it to the object recognition module as a query viewpoint q. This is described next.

C. Recognition

The input to the recognition module is a query viewpoint (q) and $V(V_1, ..., V_k)$ training viewpoints, where each element of V is defined by a set of descriptors $\{d_1, d_2, ..., d_n\}$.

Let $d(q, V_t)$ be an L_1 distance function

$$d(q, V_t) = L_1(C_q, C_{V_t}) + L_1(Std_q, Std_{V_t}) \qquad (1)$$

where C_q and C_{V_t} are the centroids of the sets q and V_t and

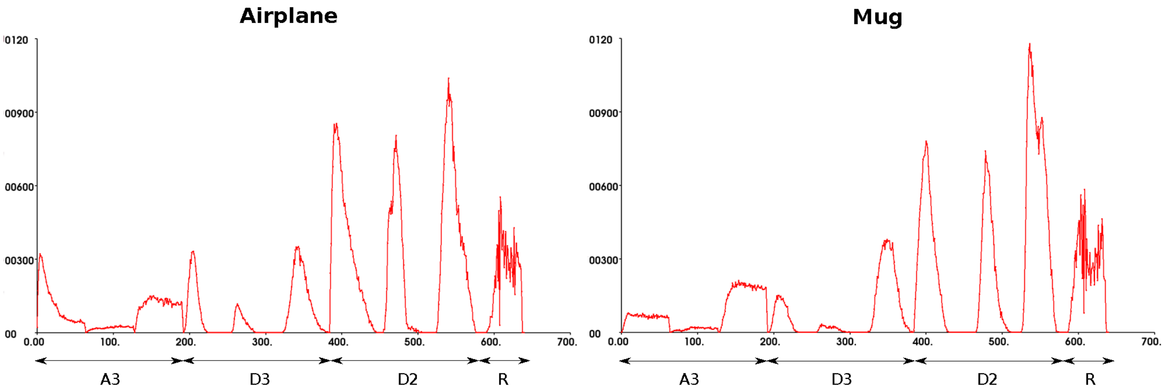

Fig. 2. Computed histograms of Airplane and Mug classes. The two histograms have a great similarity in the $D2$ metric.

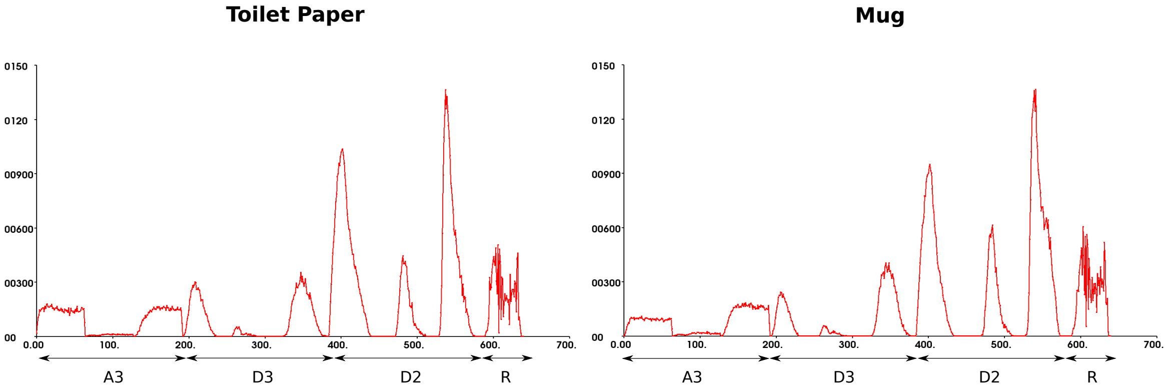

Fig. 3. Computed histograms of mug and airplane classes. Both objects have very similar signatures.

$$Std_q(i) = \sqrt[2]{\frac{1}{|q|}\sum_{j=1}^{|q|}(q_j(i) - C_q(i))^2}, i = 1, ..., n$$

and likewise for V_t; where n is the size of the descriptor. The L_1 distance is $L_1(q, V_t) = \sum_{i=1}^{n} |q(i) - V_t(i)|$.

The goal of this metric is to find the training viewpoint with the highest similarity when compared with the query viewpoint q, returning the smallest L_1 distance. When the query contains multiple viewpoints, we choose the best match by:

$$m_p = \min_{\alpha} d(q, V_\alpha) \qquad (2)$$

where $q \epsilon X$, X is a set of n_v views, and m_p is set of confidence values for each class which quantify the likelihood of the model being in that class. Each of the confidence sets are part of the prediction set $M = \{m_1, m_2, ..., m_{n_v}\}$.

After obtaining the prediction set M and given R the recognition threshold, we identify an object O if,

$$L = \arg\max(M)$$
$$S = \max(M)$$

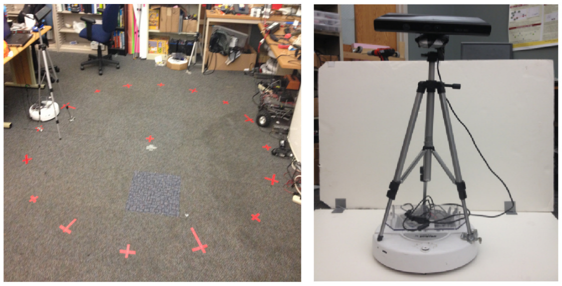

Fig. 4. Experimental Setup. The left image is the area used for the experiments, while the right image is the employed platform.

where S is the highest confidence class and L is the corresponding label among the views. If $S > R$, then return valid recognition and present the level of confidence associated to the label.

IV. System Performance

Before we explore the role of view planning, in this section we investigate system parameters that effect performance.

The Experimental setup consists of a roomba robot with a tripod and a Kinect camera, see Figure 4.

The hardware used in the experiments were an Intel Core 2 Duo 2.83GHz (using only one core) at the cloud side and an Intel Core $i5$ 2.4GHz (using only one core) for the robot. The dataset on the server contains $3D$ CAD models, where each class has a number of instances ranging from 4 (toilet paper) to 84 (mug). Since we generate 80 partial views per instance, the final number of views surpass thirty eight thousand partial point clouds.

A. *Speed and range recognition tests*

Initial test consist of estimating the recognition effective range and latency between robot and cloud recognition module. We analyzed the recognition effective range. The experimental setup consists of a camera in a tripod with a default system parameter settings. Experiments show that the best range was from 1.00 to 2.00 meters. In our tests the camera was elevated 1.1 meters from the ground and 1.3 meters from the object in a 45 degrees inclination (1.7 meters from the object).

Second we evaluate the communication issues between robot and the remote recognition module. The original size of the point cloud with color is 4500Kb, so to have a close to real-time communication between robot and recognition, compression is needed to transfer data through Internet. We perform a spatial decomposition based on octree data structures [5]. In addition to compression, we apply a pass-through filter in order to remove any points that are not inside the range from 1.0 to 3.5 meters. Based on the minimum and maximum default ranges of the camera, respectively 0.8 and 4.0 meters, we remove any data closer to 1.0 and farther than 3.5 meters. Therefore, the system operational range is from 1.0 to 2.0 meters.

Table I presents results on the performance of communication between the robot and the cloud by comparing filtered/non-filtered point clouds with compression. Compressed point clouds could be sent to the cloud recognition module in a rate of 10 frames per second, with a compression rate of $(1:42)$. However, the server cannot handle all the transferred data. To reduce the problem we employ a pass-through filter which reduces the point cloud size to 87Kb $(1:51)$ and balances the transmission frame rate with the system process capabilities.

TABLE I
COMMUNICATION RESULTS. FRAME RATE OF THE SYSTEM WITH A COMPRESSED CLOUD IS IN AVERAGE 10 FRAMES PER SECOND, WITH A SIZE OF 105KB. THE RESULTS ONLY ACCOUNT FOR THE COMMUNICATION THROUGHPUT OF THE SYSTEM, WITHOUT ANY COUPLED RECOGNITION.

Method	Mean	Size
Compressed without filter	10.35	105.26 Kb
Compressed with filter	6.55	87 Kb

After investigating communication issues, we focus our attention to the computational complexity of the recognition system. Two possible bottlenecks can be pointed out: feature descriptors and point cloud segmentation. Feature descriptors are not a big issue in the recognition method, since we use the ESF descriptor [17] which is a light weight feature.

For the segmentation we employ a plane detection approach based on RANSAC. Table II exhibits processing times for tests that consist of segment the point cloud with one, two and three objects in scene. We can notice that for one object an average of 3 point clouds can be processed per second.

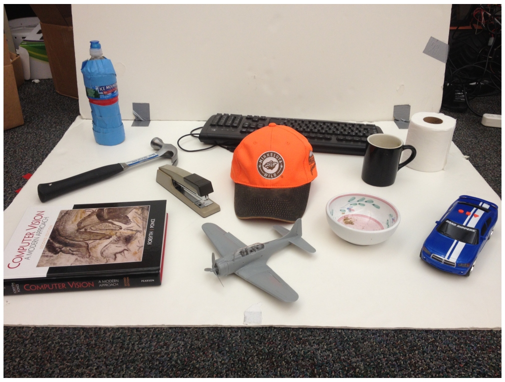

Fig. 5. The dataset consist of 11 classes. We selected these object based that they are common objects in a human-like environment.

TABLE II
SEGMENTATION COMPUTATIONAL TIMES. OUR SYSTEM SUPPORTS RECOGNITION OF MULTIPLE OBJECT IN SCENE. HOWEVER, DOING IT OUR PERFORMANCE DROPS SUBSTANTIALLY.

Segmentation # of objects	Frame-rate	Min (ms)	Max (ms)
1 object	3.4	270	310
2 objects	2.6	355	400
3 objects	1.9	500	530

These results confirm that near real-time performance can be achieved by performing segmentation and compression on the client side. These operations do not require significant computation power and can be executed on most client robots. If this is not possible, the performance of the system can decrease drastically regardless of view planning and object recognition performance.

V. Single View Results

We now investigate the role of view planning on object recognition. First we describe the experiment set up. The experiments were done using 16 planar views, with a 22.5 degrees between each view, ranging from 0 to 360 degrees. The dataset used is a subset of the 3dnet database [16]. The original dataset contains 60 classes of objects. We used 11 categories that can be grouped as usual household objects, that are stapler, mug, hammer, keyboard, cap, book, bowl, toilet paper, car, airplane and bottle, see Figure 5.

First we ask the question: is view planning necessary at all? To answer this question, we executed queries from 4 selected views in orthogonal configuration at 0, 90, 180 and 270 degrees. We first focus on what can be achieved with a single query and present results on multi-view point queries at Section VI.

1) 0 Degree Results: We have tested our system with a 0 degree orientation. Figure 6 presents the results. Classes like Stapler, Cap, Bowl, Toilet and Bottle could not be correctly recognized from this viewpoint. Stapler is confused with Car

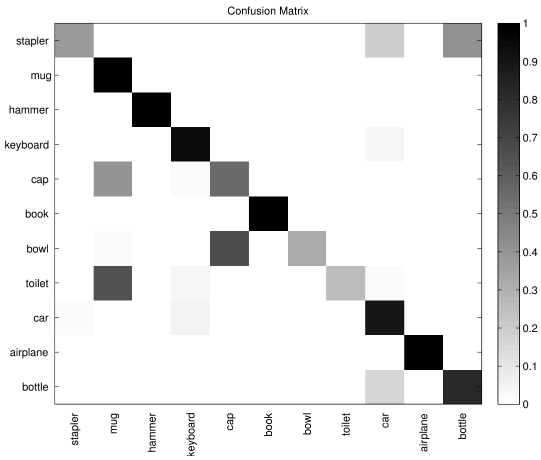

Fig. 6. Confusion Matrix 0 degree orientation.

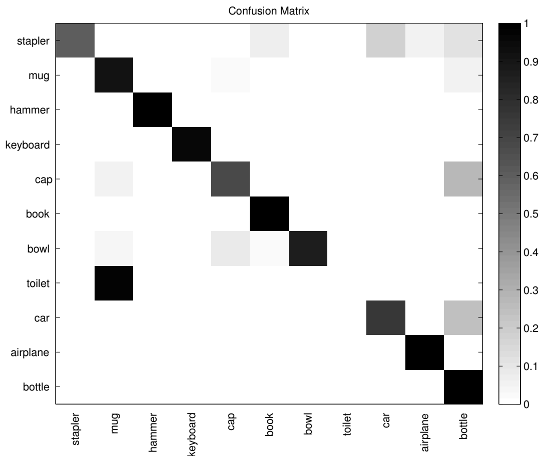

Fig. 7. Confusion Matrix 90 degree orientation.

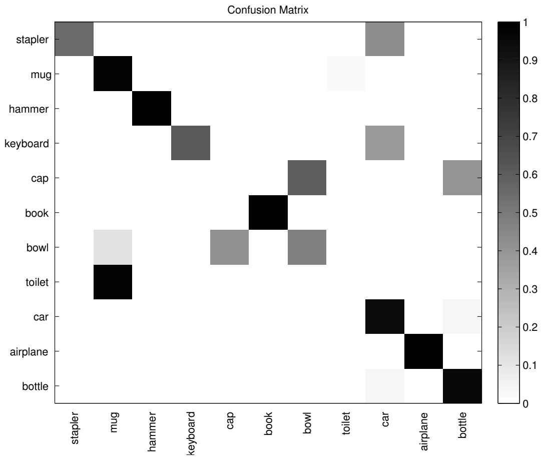

Fig. 8. Confusion Matrix 180 degree orientation.

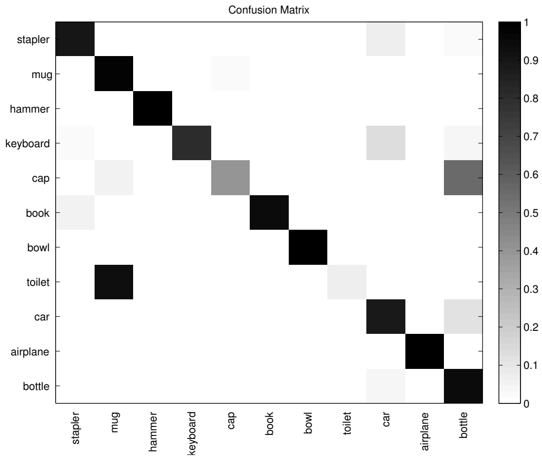

Fig. 9. Confusion Matrix 270 degree orientation.

and Bottle. Cap with mug and Bowl with Cap. Toilet Paper has been erroneously recognized as a Mug. Sometimes, the Bottle is recognized as Car. Here, wrong recognition is defined as any classification accuracy below a certain threshold, in our case below 90%.

2) 90 Degree Results: Figure 7 presents the results for the 90 degree viewpoint. Similar to the previous case, classes Stapler, Cap, Car, Bowl and Toilet could not be correctly recognized. Stapler is confused with Car, Cap with Bottle and Toilet Paper with Mug. In a small percentage Bowl is wrongly recognized as Cap and Car is confused with Bottle.

3) 180 Degree Results: Figure 8 shows the results for 180 degrees orientation. Stapler and Toilet Paper continue to be wrongly recognized with Car and Mug, respectively. Another confusion occur with Cap, that is classified as Bowl or Bottle. We can also noticed that Keyboard is confused with Car and Bowl with Cap and Mug.

4) 270 Degree Results: The last orientation tested was 270 degrees. Figure 9 shows that similarly to 0, 90 and 180, Toilet Paper and Cap could not been recognized. Toilet paper is again confused with Mug, while Cap is recognized as Bottle.

After identifying which classes are more susceptible to accuracy drop due to changes in viewpoint, we continue with the analysis of the most representative views to each tested classes.

A. Distribution of Views

We now focus on classes where the performance of recognition was highly dependent on the viewpoint. These classes are: i) Stapler ii) Cap iii) Bowl iv) Keyboard. Next, we present the recognition performance as a function of the viewpoint location for these objects for all 16 viewpoints. We also report the expected performance of a viewpoint chosen uniformly at random.

1) Stapler: Stapler was one of the objects that presented a clear region of high recognition. From 225 to 337 degrees most of the obtained views had accuracies above 80%, see Figure 10. The average accuracy for a randomly selected initial viewpoint in an orthogonal configuration is 91.7%.

2) Cap: Cap is a challenging object to recognize because certain views present no shape information that could be discriminated, see Figure 11. We observed that a view without the brim of the cap is easily confused with the mug class, it happens at the region around 180 degrees orientation. Figure 11 shows one property of this class that is absence of a

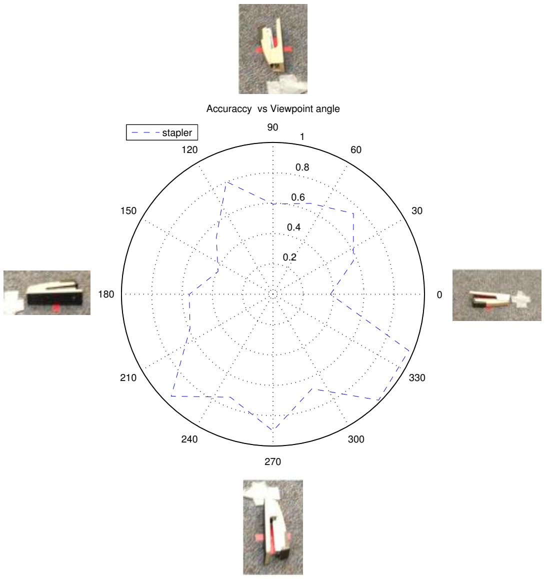

Fig. 10. Viewpoint vs Accuracy Stapler Class.

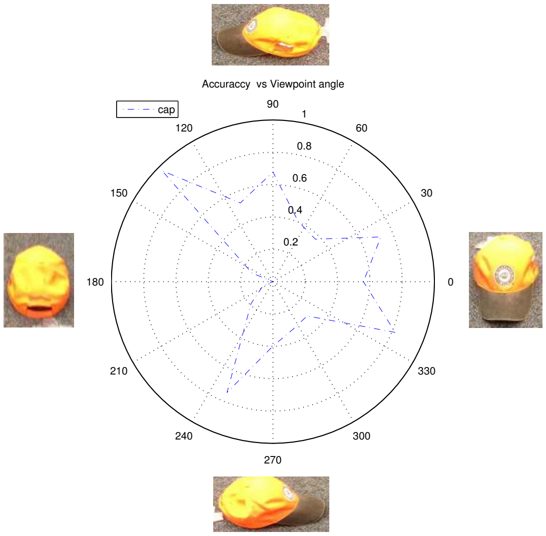

Fig. 11. Viewpoint vs Accuracy Cap Class.

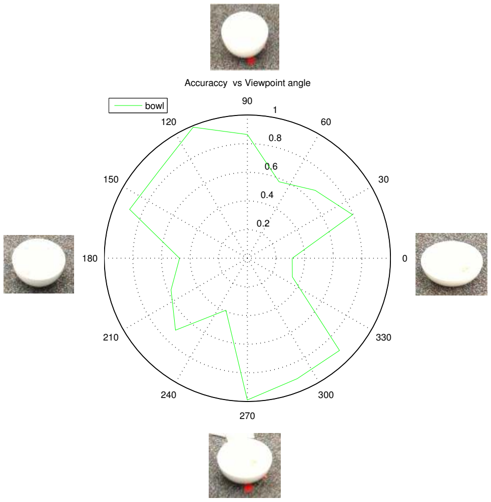

Fig. 12. Viewpoint vs Accuracy Bowl Class.

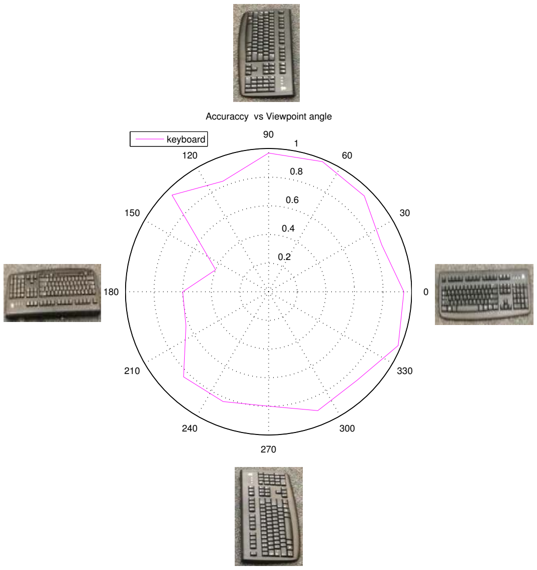

Fig. 13. Viewpoint vs Accuracy Keyboard Class.

broad region of high accuracy, orientation 135, 247.5, 337 show high observable results, however their neighbors exhibit drastic decay in term of classification. Having as prerogative the lack of a clear recognition region, we obtained an average accuracy for a random initial viewpoint of 79.8%.

3) Bowl: Bowl is one the classes with the most clear regions of recognition. Figure 12 displays two regions, one from 90 to 157 degrees and a second one from 270 to 315 degrees. The first region covers 67 degrees of the object while the second 45, in a total of 112 degrees. For one random initial position of the orthogonal views configuration, the chance of hit these regions is very high, with an average accuracies of 92.65%.

4) Keyboard: The keyboard class is one of the highest accuracies in the performed tests, that can be visualized at Figure 13. Regardless of this high rate for most of the views, the class presented a significant performance drop for the region in the keyboard back, from 135 to 225 degrees. In our test the drop reach to at least 20% when compared to the region from 0 to 90 degrees, that is respectively the range of orientations with the best results.

This motivates the use of multiple view queries. As we will see in the next section, expected accuracy can be increased to 95% with a multi-view approach.

VI. Multiple View Approach

Equipped with the evidence that regions with high discriminative power exist, we now search for the Universal Viewpoint Sets (UVSs). The goal is to choose a set of viewpoint guaranteed to include at least one of the highly distinguishable views. We investigated several viewpoint configurations with varying cardinality and angle arrangements.

We have tested 8 different configurations, four with 3 views, three with 4 views and one with 6 views. The metric we used was:

$$\underset{\theta}{avg} \max_{c=1,\ldots,n} Quality(\theta) \tag{3}$$

where c is the cardinality of the view planning set and $\max Quality(\theta)$ measures the accuracy of one specific initial viewpoint $\theta = \{0, 22.5, ..., 360\}$ maximizing it's value over the wished configuration. Therefore, Eq. 3 provides the average accuracy of the best viewpoint of the set, given all possible initial positions.

One characteristic of open loop approaches that must be analyzed is the robustness to the initial viewpoint, described as the 0 orientation. It is a key element of the system, since given different initial points the method must be able to hit the UVS of the object. We highlight the tests based on random selection of the initial views for the stapler class, see Table III. For this class, we observe that with less than 4 views, we cannot observe the object in a way that could capture one of the best views. This suggests a lower bound on the cardinality of the UVS. More views, like 6, can produce some gain over the 4 orthogonal view approaches, however the marginal gain is small. For example the average response to the stapler class for 4 orthogonal views is 91.7 and for 6 views is 94.9 percent. Since size 6 is a 50% bigger sample than 4, that gain are not expressive enough to be picked. Another reason is that for the 4 views case, we obtain an average accuracy above 90%, which exceeds our threshold for correct recognition.

TABLE III

AVERAGE ACCURACY OVER A RANDOMLY SELECTED INITIAL VIEW, USING EQ. 3. FOUR ORTHOGONAL VIEWS ARE JUST BELOW THE 6-VIEW CONFIGURATION. HOWEVER, THE OBTAINED ACCURACY WITH 4 VIEWS ALREADY SATISFIES THE THRESHOLD FOR CORRECT RECOGNITION.

# of views	Configuration	Avg Accuracy (%)
3	0, 45, -45	85.88
3	0, 90, -90	87
3	0, 112, -112	87.3
3	0, 135, -135	87.6
4	0, 45, -45, 180	90.5
4	0, 67.5, -67.5, 180	90.8
4	0, 90, -90, 180	**91.7**
6	0, 67.5, 135, 202.5, 270, 337.5	**94.9**

With these experiments we can state our main result: Four orthogonal planar views are sufficient to recognize common household objects.

As a final step, we show how to fuse each of the 4 orthogonal views into one prediction. The so-called aggregative method is based on the selection of the highest confidence response among the views, see Figure 14. The matrix shows that only the toilet paper and cap classes were not correctly recognized. This constitutes a clear progress over single view approaches. Additionally, it can be noticed that no particular viewpoint can classify the whole dataset correctly. Since we have an almost uniform division of views, with 3 classes best recognized with 0 degree, 3 classes with 90 degrees and 2 classes for 180 and 270 degrees.

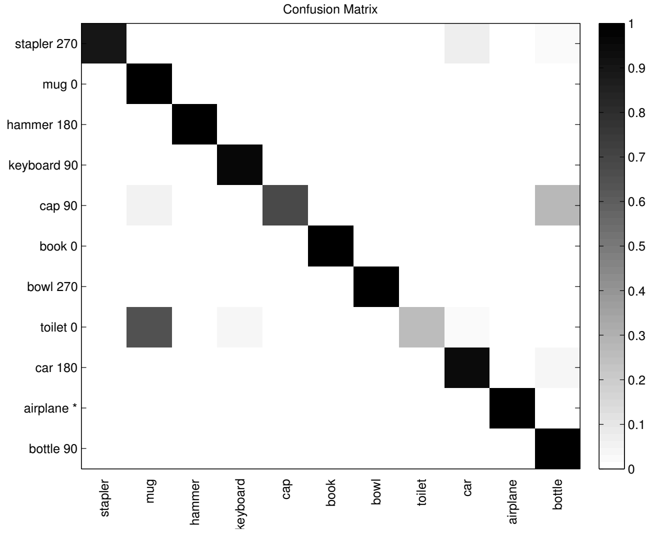

Fig. 14. Aggregated confusion matrix, produced by the merging of the 4 views at 0, 90, 180 and 270 degrees. Airplane class is marked with $*$ because for 90, 180 and 270 all outputs were the same. Another characteristic of this matrix is the vertical column that indicate the orientation with the maximum response.

VII. CONCLUSION

This paper initiates our investigation of the role of view planning for cloud-based object recognition. View planning is a key aspect of cloud-based approaches because latency has significant impact on these applications, and selecting views can greatly reduce the amount of communication between robot and cloud. We study the impact of the view-point and performance of various configurations. We propose an open-loop view planning technique, in which the robot simply obtains four orthogonal views on the plane. In contrast to well-known view planning techniques, our method does not build any partial model of the object to compute the next view. This significantly reduces the computational load on the client side without decreasing recognition performance.

Our results are based on experiments performed over a subset of the 3dnet dataset, with 11 classes. They provide evidence that Universal View Set (UVS) exists and hence an open loop view planning method is feasible.

The implemented prototype of cloud-based object recognition are promising. However more work is needed to scale up the system. Our future works consist of evaluating the system with a larger dataset on a more powerful server and investigating the existence of UVSs using analytical methods.

In this work, we focused on planar viewpoints since most robots which would benefit from cloud based methods are simple mobile robots such as Romo [8]. In our future work, we would like to extend the method by incorporating views in $3D$ space which would be valuable for robots with manipulators and indoor flying robots.

VIII. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under the grant number 1317788.

REFERENCES

[1] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *IJRR*, 30(11):1343–1377, 2011.

[2] Tiffany Chen, Matei Ciocarlie, Steve Cousins, Phillip M. Grice, Kelsey Hawkins, Kaijen Hsiao, Charlie Kemp, Chih-Hung King, Daniel Lazewatsky, Adam Eric Leeper, Hai Nguyen, Andreas Paepcke, Caroline Pantofaru, William Smart, and Leila Takayama. Robots for humanity: A case study in assistive mobile manipulation. *IEEE Robotics & Automation Magazine, Special issue on Assistive Robotics*, 20, 2013.

[3] D. Hunziker, M. Gajamohan, and R. D'Andrea. Rapyuta: The robotearth cloud engine. In *ICRA*, 2013.

[4] Zhaoyin Jia, Yao-Jen Chang, and Tsuhan Chen. Active view selection for object and pose recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 641–648, 2009.

[5] J. Kammerl, N. Blodow, R.B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach. Real-time compression of point cloud streams. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 778–785, 2012.

[6] Ben Kehoe, Akihiro Matsukawa, Sal Candido, James Kuffner, and Ken Goldberg. Cloud-based robot grasping with the google object recognition engine. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2012.

[7] K. Lai and D. Fox. 3d laser scan classification using web data and domain adaptation. In *Robotics: Science and Systems (RSS)*, 2009.

[8] Romotive LLC. Romo robot. *http://romotive.com*, Accessed 2013-09-15.

[9] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(5):417–433, 1993.

[10] Gerard McKee. What is networked robotics? In *Informatics in Control Automation and Robotics*, pages 35–45, 2008.

[11] Farzin Mokhtarian and Sadegh Abbasi. Automatic selection of optimal views in multi-view object recognition. In *BMVC*, pages 1–10, 2000.

[12] Will Schroeder, Kenneth M. Martin, and William E. Lorensen. *The visualization toolkit (4th ed.): An object-oriented approach to 3D graphics*. Litware, Inc., 1998.

[13] William R. Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1):64–96, 2003.

[14] Konstantinos A Tarabanis, Peter K Allen, and Roger Y Tsai. A survey of sensor planning in computer vision. *Robotics and Automation, IEEE Transactions on*, 11(1):86–104, 1995.

[15] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, and R. van de Molengraft. Roboearth. *Robotics Automation Magazine, IEEE*, 18(2):69–82, 2011.

[16] W. Wohlkinger, A. Aldoma, R.B. Rusu, and M. Vincze. 3dnet: Large-scale object class recognition from cad models. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5384–5391, 2012.

[17] W. Wohlkinger and M. Vincze. Ensemble of shape functions for 3d object classification. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 2987–2992, 2011.