Multi-view 6D Object Pose Estimation and Camera Motion Planning using RGBD Images

Juil Sock
Imperial College London
London, UK
ju-il.sock08@imperial.ac.uk

S.Hamidreza Kasaei*
University of Aveiro
Aveiro, Portugal
seyed.hamidreza@ua.pt

Luis Seabra Lopes
University of Aveiro
Aveiro, Portugal
LSL@ua.pt

Tae-Kyun Kim
Imperial College London
London, UK
tk.kim@imperial.ac.uk

Abstract

Recovering object pose in a crowd is a challenging task due to severe occlusions and clutters. In active scenario, whenever an observer fails to recover the poses of objects from the current view point, the observer is able to determine the next view position and captures a new scene from another view point to improve the knowledge of the environment, which may reduce the 6D pose estimation uncertainty. We propose a complete active multi-view framework to recognize 6DOF pose of multiple object instances in a crowded scene. We include several components in active vision setting to increase the accuracy: Hypothesis accumulation and verification combines single-shot based hypotheses estimated from previous views and extract the most likely set of hypotheses; an entropy-based Next-Best-View prediction generates next camera position to capture new data to increase the performance; camera motion planning plans the trajectory of the camera based on the view entropy and the cost of movement. Different approaches for each component are implemented and evaluated to show the increase in performance.

1. Introduction

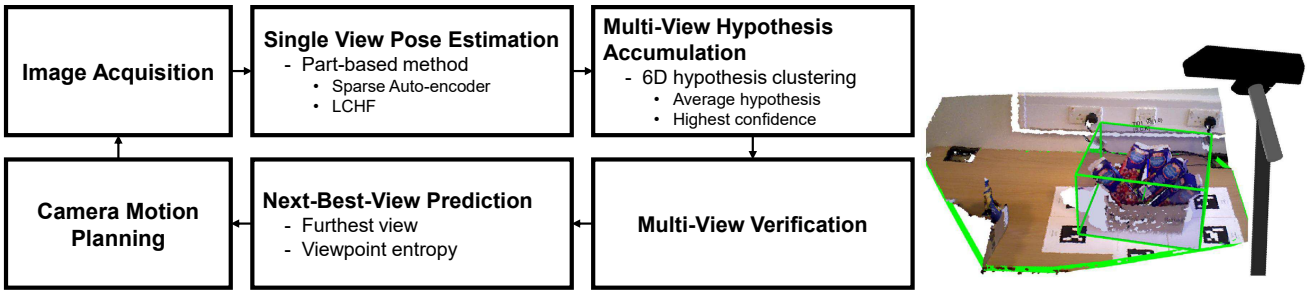
Highly accurate pose estimation of a known object is crucial for robotics applications and attracted interests in the research community recently. Many researchers participated in the public challenges such as Amazon picking challenge to solve multiple objects detection and pose estimation in a realistic scenario. This reflects that the object pose estimation is moving towards more realistic robotics environment where the platform has control over its perception, in other words, active vision.

It is a challenging task to robustly detect and estimate poses of multiple objects stacked in a pile or in a box, which are often found in a warehouse environment. The environment can be highly **crowded** and **cluttered**, deceiving the detector. **Occlusion** and **self-occlusion** can lead to only a part of an object to be visible in certain views and some captured data can be inaccurate or missing due to **sensor noise**.

Several strategies have been used to overcome the issues. Multi-view object detection and recognition[5][7][18][20] attempts to recognise and detect objects in a scene using multiple views. In particular, [7] and [5] are able to select views which would increase the classification performance for recognising a single object. Multi-view object pose estimation[6][9][10][23][25] aims to detect and estimate accurate pose of multiple objects simultaneously. However, view selection for object pose estimation is not common in this framework. Doumanoglou et al.[9] proposed an approach to predict the next view based on the class entropy, which is useful when there are different classes with highly similar appearances.

Our goal is to build a complete system for multi-view active vision pose estimation scenario to correctly estimate multiple object poses in a challenging environment. The overall system architecture is depicted in Fig. 1(a). In our setup, two state-of-the-art single object pose estimators, namely LCHF[21] and Sparse auto-encoder[9] are used to generate multiple hypotheses. Then, the point cloud information and the object pose estimation results from every view are collected and refined in *multi-view hypothesis ac-*

*The second author was funded by FCT scholarship SFRH/94183/2013.



(a) (b)

Figure 1. Overview of the vision algorithm: (a) the system iteratively accumulates more information as it captures more data from different views : (b) the system aims to detect and estimate accurate pose of multiple objects using multiple view of the scene.

cumulation module. In this stage, object hypotheses are first transformed to a world reference coordinate using camera pose and then clustered. The number of cluster and center of clusters are determined using subtractive clustering[4]. A representative hypothesis is then selected for each cluster using averaging[23] or by choosing hypothesis with the highest confidence. Next, the clustered hypotheses are verified with the registered point cloud to remove false hypothesis. The last part of the system is to predict the next best view that would increase the object detection and the recognition performance. Towards this goal, point cloud from different viewpoint together with hypotheses will be used to render unseen views for calculating view entropy. To show the proposed viewpoint entropy can be useful to determining the NBV, we built a synthetic dataset with densely sampled view points. We show that this view entropy is useful in determining the next best view in certain scenarios.

We demonstrate our system on a challenging bin-picking scenario from [9] where some objects are visible from only selective views. We show that the proposed system improves the object detection and pose estimation performance. The system is also able to select the next view based on accumulated information and generate camera trajectory taking into account the cost of movement.

Our contributions are:

1. Integration of different components to build a complete active system which detects and pose estimates multiple objects.

2. Unsupervised Next-Best-View(NBV) prediction algorithm to predict the next best camera pose for object detection and pose estimation by rendering the scene based on current object hypotheses.

3. Generating a synthetic dataset with realistic multi objects configuration using a physics engine.

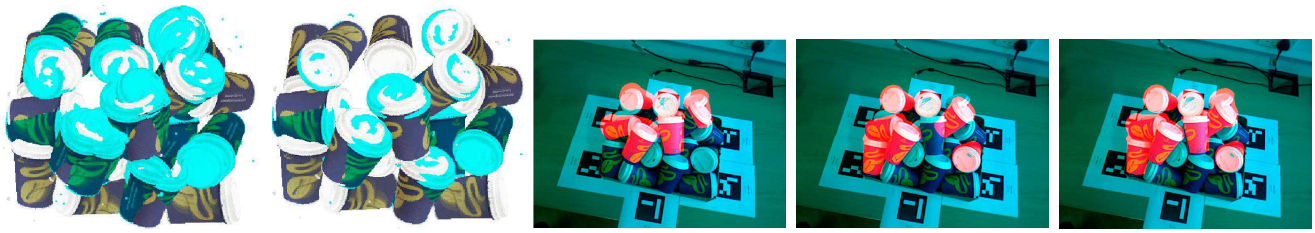
2. Related Work

In this section, we will first review the existing multi-view object recognition literature followed by the relevant literature for individual components in Fig. 1(a).

Multi-view object recognition Multi-view object recognition system aims to increase the recognition performance by combining information captured from different views. Multiple views of the scene can be used to overcome some critical issues listed in section 1, if view information is used appropriately.

Multi-view approach has shown to significantly increase object detection and recognition performance. Coates and Ng[5] executed single shot-based detector on 2D image for each view and determined the correspondences to combine the posterior probability detection. Jayaraman and Grauman[7] presented an end-to-end object recognition framework which actively selects the best view for the classification purpose. However, the objective is limited to recognizing a single object. Mustafa et al.[18] used multiple fixed Kinect and stereo camera to recognize a single object reliably using texlet. Suanto et al.[20] detected multiple objects from each view using Deformable Part Model(DPM) and used Viewpoint Feature Histogram(VFH) to eliminate false hypotheses. In general, multi-view object recognition system[7][18][15] do not need verification component in Fig. 1(a), since a single object of a predefined category is assumed to be present in the scene. Multi-view object detection system in [20][18] merge information from fixed multiple sensors. Therefore, view selection is not part of the system.

More closely related to our work is multi-view object pose estimation systems. Collet and Siddhartha[6] used multiple cameras to detect and estimate the pose of multiple objects. Hypotheses from different views are combined, and the hypothesis that has the lowest reprojection error is selected. Doumanoglou et al.[9] did not explicitly use multi-view information to increase the object pose estimation performance, but used class entropy from random forest to predict the NBV for classification. Erkent et al[10] used a probabilistic approach to integrate hypotheses generated from different views. Hypotheses were given probabilistic values by replacing each of them with Gaussian dis-



(a) (b)

Figure 2. (a) Accumulated hypotheses from 4 different frames. blue coloured hypotheses are overlaid on top of the ground truth : (left) hypotheses collected from different views are displayed. Note there may be multiple hypotheses for each ground truth object; (right) a set of hypotheses after clustering. Clustering correctly reduces multiple hypotheses to its corresponding ground truth object. (b) Discovering hypotheses: red pixels are the projection of hypotheses; (left, middle) a set of hypotheses from different view projected onto a reference image; (right) clustering result. Note that the algorithm successfully merge results from different scene, thus increasing the recall.

tribution with mean equals to the corresponding hypothesis. Viksten et al.[23] used a strategy to transform every hypothesis obtained from different views to reference frame and used mean-shift clustering to find the hypothesis for each cluster. Zeng et al.[25] presented a pick-and-place vision system which integrates multi-view information to increase the pose estimation accuracy. However, the scenes were not actively selected based on the collected information. Unlike our implementation, most of the multi-view object pose estimation in literature do not actively select the views.

Single shot based 6D object pose estimation There exist many different approaches to detect and estimate object pose from a single image, but the effective approach differs depending on the scenario. Holistic template based approach[13][14] is effective when there is less occlusion. Pixel-wise pose estimation approach in Brachmann et al.[3] assumed single object in the scene. The part-based model[21][8] handles occlusion by subdividing the images into patches and recognize each patch separately. Doumanoglou et al.[9] used sparse autoencoder to represent each patch and classified using random forest. Tejani et al.[21] used LINEMOD[13] to represent each patch and also used random forest and hough voting to generate hypotheses. [9] and [21] show strong robustness against occlusion and clutter. Therefore, they are suitable to be used in our experiment scenario.

Next-Best-View prediction Mauro et al. [17] proposed a unified framework for content-aware next best view selection based on several quality features such as density, uncertainty, 2D and 3D saliency. Using these features, they computed a view importance factor for a given scene. Unlike to this approach, we first segment a given scene into object hypotheses. Then, the next best view is predicted based on the property of those object hypotheses. In another work, Bo Li et al. [2] approached the problem of defining the representative views for a single 3D object based on visual complexity. They suggested a new method for measuring the viewpoint complexity based on entropy. Their approach revealed that it is possible to retrieve and to cluster similar viewpoints. [9] uses class entropy of samples stored in the leaf nodes of hough forest to estimate the Next-Best-View which could reduce the uncertainty of the class of detected objects.

Unlike the above approaches, some researchers have recently adopted deep learning algorithms for next best view prediction in active object detection scenario [24, 15]. For instance, Wu et al. [24] proposed a deep network namely 3D ShapeNets to represent a geometric shape as a probabilistic distribution of binary variables on a 3D voxel grid. As they pointed out, training a deep network for next best view prediction requires a large collection of 3D objects to provide accurate representations and typically involves long training times. Moreover, unlike our approach, these kinds of approaches are mainly suitable for isolated objects or single object scenarios and become brittle and unreliable in crowded scenarios.

3. Multi-View Object Pose Estimation Framework

This section presents the *hypothesis accumulation and refinement* and *unsupervised next-best-view prediction* modules in details.

3.1. Hypothesis accumulation and refinement

Hypothesis accumulation and refinement combine single-shot based hypotheses estimated from different views and extract the most likely set of hypotheses. Any single shot-based pose estimation algorithm that outputs object pose hypothesis together with a confidence value can be used in this framework.

Firstly, we accumulate hypotheses collected from different Hypotheses estimated from different views. All hypotheses from different views are transformed to the global coordinate frame using the camera pose information. Then we cluster the hypotheses with only the 3D positions of hypotheses as usually hypotheses in full 6 dimension are too sparse to be clustered successfully. However, simple euclidean clustering is not robust when the objects are tightly packed as any outlier between clusters may connect differ-

ent objects. We used subtractive clustering[4] to determine the number and the centers of clusters and clustered objects within a certain radius. Hypotheses assigned to each cluster are either averaged in quaternion space[23] or the hypothesis with the highest confidence value is chosen to represent the cluster. Fig. 2(a) and Fig. 2(b) show the effects of clustering. Fig. 2(a) shows that multiple hypotheses in each cluster merge to produce a single hypothesis. This has the effect of removing less accurate hypotheses. Fig. 2(b) shows more objects are detected, resulting in increased recall.

Unlike [13][3], which assume single object in the scene, we need verification to identify and eliminate false hypotheses to increase True Positives(TPs) and reduce False Positives(FPs). Because methods shown in [1][9] are not directly applicable in multi-view setting, we adopted the cues from the paper and applied the method on accumulated pointcloud rather than the single RGBD image.

3.2. Unsupervised Next-Best-View Prediction

In the previous step, the observer captures an accumulated point cloud of the scene and computes a list of 6D objects hypothesis. The inputs to the Next-Best-View (NBV) prediction module are: the full 3D models of the objects; the point cloud of the scene; a set of verified 6D object hypotheses; $\mathbf{P} = \{h_1, \ldots, h_n\}$; and the possible viewing pose, $\mathbf{V}$ where $\mathbf{V} = \{v_1, \ldots, v_m\}$ is a finite list of possible viewing pose representing the camera rotation and translation in 3D space.

The 6D object pose estimation is challenging for a pile of overlapping objects and it may not detect all objects from the scene. Therefore, the system should hypothesise each segment of the scene as objects. To this end, a hierarchical clustering procedure is presented to segment unstructured and highly cluttered scenes using geometric, surface normal data and color. Our segmentation pipeline is composed of two processes. The first process computes the regions of interest from the scene. It starts with extracting points which lie directly above a horizontal support plane. This is done by first finding the dominant plane in the point cloud using the RANSAC algorithm [12]. The scene is then segmented into individual clusters using a Euclidean Cluster Extraction algorithm[1][16, 19]. Finally the extracted point cloud is dispatched to the second process for further segmentation analyses.

The second process of the segmentation pipeline extracts a set of object hypotheses from the given point cloud. A region of the given point cloud is considered as an object hypothesis whenever points inside the region are continuous in both the orientation of surface normals and the depth values. The depth continuity between every point and its immediate 8 neighbors is computed. If the distance between points is lower than a threshold, then the two points belong to the same region. A color-based region growing segmentation is also applied on large hypotheses. Fig. 3 illustrates the results of the second segmentation process in three different scenes. Each object hypothesis (i.e., a cluster of points) will be treated as an object candidate namely c_i, where $i \in \{1, \ldots, K\}$. It should be noted that the number of clusters, K, is not pre-defined and it varies for different viewpoints. Next, the obtained clusters are used to compute viewpoint entropy for the given scene. There are various methods for computing the viewpoint entropy. In general, the number of visible voxel or points is used as an indicator of the area for entropy computation. This measure is not good enough for 6D object pose estimation purposes since it only considers the coverage objective. Therefore, we propose a new formulation for viewpoint entropy calculation that takes into account both the coverage (i.e. the number of visible points) and saliency (i.e. observing a large portion of an object which can potentially reduce the pose estimation uncertainty) objectives. The viewpoint entropy of a given scene is computed as follows:

$$H = -\sum_{i=1}^{K} \frac{A_i}{S} \log \frac{A_i}{S}, \tag{1}$$

where, K is the number of clusters, A_i is the area of the i^{th} cluster and S is the total area of the given scene. Before actually moving the camera, we aim to predict the NBV from the camera pose list, V. For this purpose, first, we have to predict what can be observed from each pose in V by taking a 'virtual point cloud'. Toward this goal, based on the given set of 6D objects hypothesis, the full model of objects are first added to the current scene (see Fig.4 (a) and (b)). Afterwards, for each possible camera poses, a virtual point cloud is rendered based on depth buffering and orthogonal projection methods (see Fig.4 (c) and (d)). Then, the viewpoint entropy is calculated for each rendered view as before. This procedure is illustrated in Fig.4.

In general, choosing the view with the minimum view-entropy as the next camera position has two problems. Firstly, in real system, it costs system to move the camera too far at a time. Secondly, view entropy estimation becomes less reliable if the rendering view is far from the current position, since the view entropy calculation is based on the rendered virtual point cloud. To alleviate this issue, we apply weights to the view entropy value calculated for each view candidate by Gaussian distribution.

$$\begin{aligned} H_{Vi}^{Weighted} &= W_{V_i}(V_i) H_{V_i}, \\ where \quad W_{V_i}(V_i) &= \frac{1}{\sigma\sqrt{2\pi}} e^{-(V_C - V_i)^2 / 2\sigma^2} \end{aligned} \tag{2}$$

where σ is a smoothness parameter which restrict the movement of the camera, W_{Vi} is the weight applied to view

[1] http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php

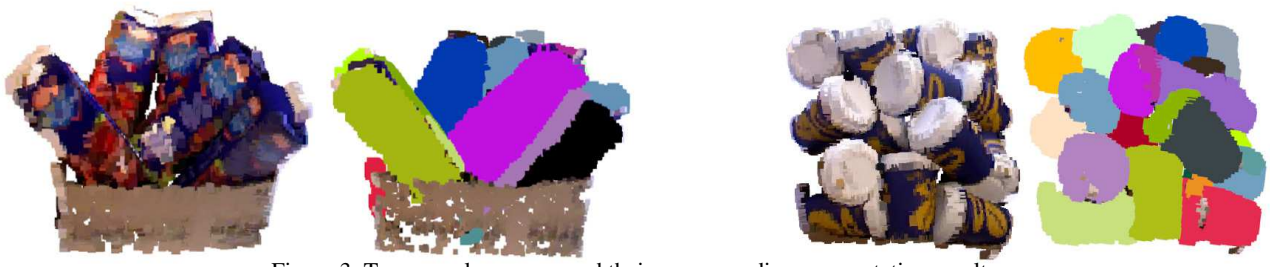


Figure 3. Two complex scenes and their corresponding segmentation results.

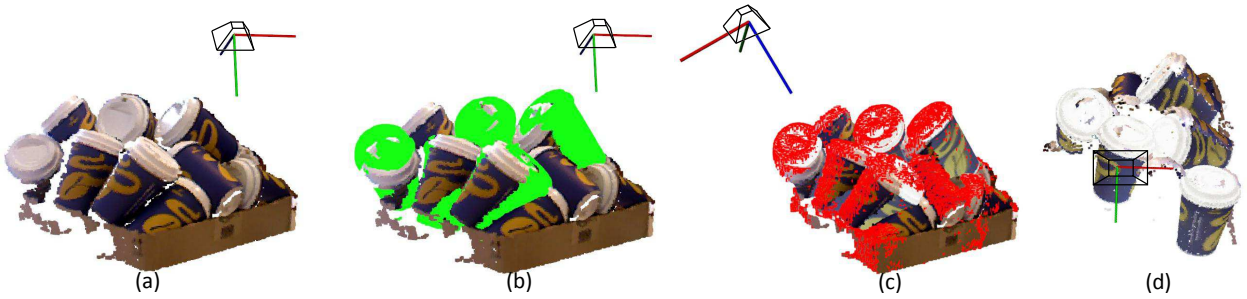


Figure 4. Rendering virtual point cloud for unsupervised next best view prediction: (a) original point cloud; (b) the full model of detected objects are added to the scene (i.e. corresponding points are highlighted by green color); (c) the visible points from the virtual camera pose are highlighted by red color; (d) the rendered virtual point cloud. The reference frame represents the camera pose of the acquired view.

entropy for V_i, V_C is the current camera pose, H_{V_i} is the view entropy of the view V_i and $H_{Vi}^{Weighted}$ is the weighted view entropy of the view V_i. Although simple $V^{Next} = \arg\min_{V_i}(W_{V_i}(V_i)H_{V_i})$ can be use to determine the next camera view position, there is a risk of the camera moving only locally. The following equation introduces perturbation in the movement to encourage the camera to move to new views.

$$p(V^{Next} = V_i) = H_{V_i}^{Weighted} / \sum_{n=1}^{m} H_{V_n}^{Weighted} \quad (3)$$

Algorithm 1 Entropy-based NBV algorithm

Input: The full 3D models of the objects.
Output: The output estimated pose **P**:

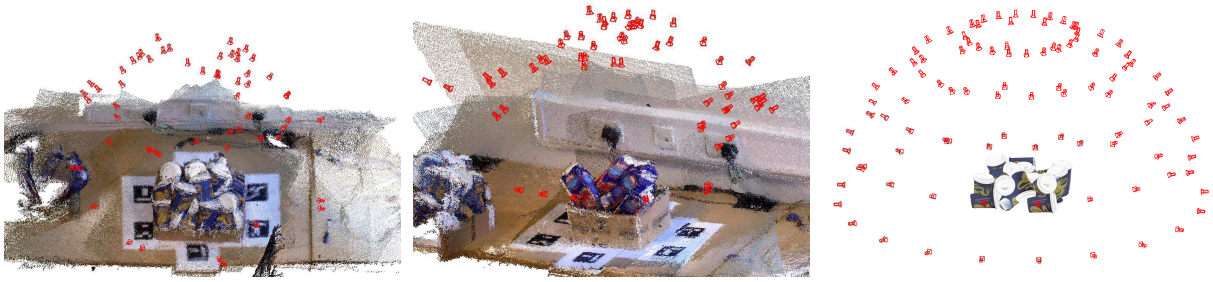
1: **Initialization** $\mathbf{P}^h = \{\emptyset\}$, $V^{Next} = v_1$
2: **while** $i \leq k$ **do**
3: Set $V_c = V^{Next}$ and obtain RGBD sensor data Z^{V_c}
4: Obtain a list of hypotheses $\mathbf{H}^{V_c}$ using [21] or [9]
5: add the hypotheses to the hypotheses pool, $\mathbf{P}^h = (\mathbf{P}^h \cup \mathbf{P}^{V_c})$
6: Use clustering[4] to obtain reduced set **P** from $\mathbf{P}^h$
7: Use **P** and Z^{V_c} to render virtual scenes and calculate viewpoint entropy(Equation 1) for **V**
8: Estimate the V^{Next} using Equation 3
9: $i = i + 1$
10: **end while**

	LCHF[21]		sparse auto-encoder[9]	
Coffee cup	Average[23]	high confidence	Average[23]	high confidence
single view	0.3365	0.3365	0.2648	0.2648
2 views	0.3710	0.4001	0.2718	0.3076
3 views	0.4332	0.4549	0.3066	0.3288
4 views	0.4601	0.5057	0.3179	0.3692
5 views	0.4819	0.5288	0.3233	0.3857
6 views	0.4826	**0.5494**	0.3399	**0.3861**
Juice box				
single view	0.3122	0.3122	0.3489	0.3489
2 views	0.1836	0.3182	0.2959	0.3385
3 views	0.1956	0.4110	0.3215	0.4306
4 views	0.1873	0.4377	0.3283	0.4757
5 views	0.1609	0.4072	0.3219	0.5386
6 views	0.1343	**0.4424**	0.2695	**0.5363**

Table 1. Detection and pose estimation performance with varying number of view accumulation. Two different pose estimator baseline([21] and [9]), and two different hypothesis selection methods(averaging[23] and high confidence) are compared. Performance is measured in AUC.

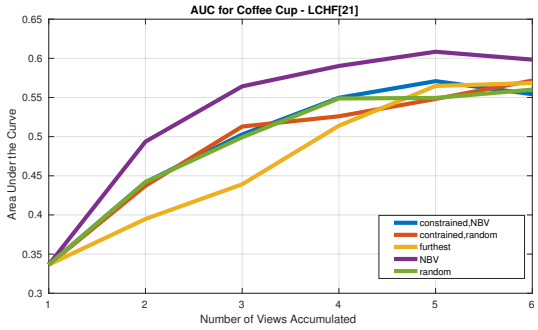
	LCHF[21]			sparse auto-encoder[9]		
Juice box	Random	Furthest	ours	Random	Furthest	ours
single view	0.3122	0.3122	0.3122	0.3489	0.3489	0.3489
2 views	0.3182	0.3685	0.3570	0.3740	0.3385	0.3623
3 views	0.4110	0.4452	0.4185	0.4135	0.4306	0.3874
4 views	0.4377	0.4650	0.4364	0.4610	0.4757	0.4918
5 views	0.4072	0.4543	0.4528	0.4942	0.5386	0.5220
6 views	0.4424	0.4569	0.4416	0.4819	0.5363	0.4944

Table 2. Performance evaluation for different strategies used to generate the next view on juice box scenario.

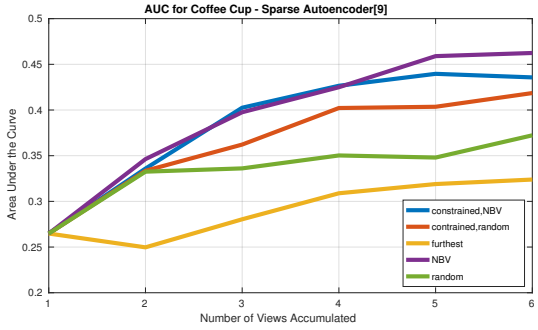


(a) (b) (c)

Figure 5. Accumulated view and camera pose of different view of bin-picking dataset[9]: (a) coffee cup scenario contains 15 coffee cups and 59 views; (b) Juice box scenario contains 5 juice boxes and 59 views. (c) Synthetic dataset contains 20 coffee cups and 100 views.



(a)



(b)

Figure 6. (a) Number of view accumulation against AUC graph for [21] using different NBV strategies. (b) Number of view accumulation against AUC graph for [9] using different NBV strategies.

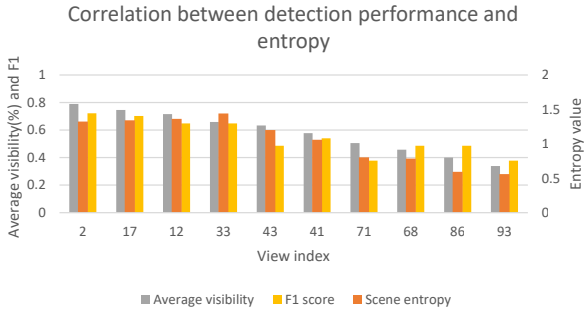


Figure 7. Graph showing the correlation between visibility, viewpoint entropy and detection performance. The graph is obtained using synthetic image. View indices are sorted in visibility descending order. The graph clearly shows the performance of detector deteriorate as less objects are visible along with the viewpoint entropy.

4. Experiments

In this section, the system is evaluated on two different crowded environments with different objects, namely coffee cup and juice box scenario. Coffee cup scenario has more objects and more occlusions whereas juice box scenario has less objects. For all experiments, ground truth camera positions provided by the dataset are used to transform point cloud and hypotheses from different views to the world coordinate. Section 4.1 contains detailed explanation on the evaluation criteria for multi-view object pose estimation system. The section 4.2 shows that performance improves when more number of views are combined and compare two different method for choosing a hypothesis for each cluster. Section 4.4 mainly compares different camera motion planning strategies and their performance.

4.1. Evaluation criteria

F1 score is a preferred method to evaluate object pose estimation result for single view methods[9][21]. However F1 score is a harmonic mean of precision and recall, values of which varies with different rank threshold(hypothesis confidence). Evaluation of the multi-view system performance is more complicated. There are two issues with the current evaluation metric: firstly, in a highly crowded scenario as shown in Fig.3, it is not reasonable to count invisible or barely visible object to calculate recall; secondly, precision, recall or F1 score change when different rank threshold value is used[11]. Therefore we evaluate our system performance using Receiver Operating Characteristic(ROC) curve and its Area Under the Curve(AUC) with visibility information of each ground truth in each view.

Doumanoglou et al.[9] calculated recall as the proportion of number of correctly estimated hypotheses to the total number of objects to be estimated, regardless of visibility in the view. We evaluated our result in a similar way

but we have taken into account the visibility criteria. If an object is not visible above a threshold, in our case 30%, the object is not included in the ground truth. Visibility score is defined as the proportion of non-occluded pixel point to all pixel point belonging to the model in the view. In multi-view scenario, if an object is visible in any of the views with score above the visibility threshold, it is added to the list of objects to be found. In this way we can fairly compare the performance of single shot-based estimator and multi-view system because this method penalises multi-view approach by having to correctly estimate more object than single shot-based.

4.2. Hypothesis accumulation and refinement

We test the system using LCHF[9] and sparse auto-encoder[21] as baselines on the bin-picking dataset[9], which is one of few datasets that contains multiple objects in a highly crowded scene. The dataset is visualize in Fig. 5. The coffee cup scenario contains 59 different views of the scene with 15 cups in a pile. Juice box scenario contains 5 juice boxes and also has 59 views.

A set of hypotheses are accumulated from different views and representative hypotheses for each clusters are obtained as described in section 4.2. From each cluster we extract one hypothesis by either averaging the hypothesis as shown in [23] or by choosing the hypothesis with the highest confidence. For every experiments in this section, each view was selected as starting camera pose and randomly selected the next views. The experiments were run 3 times and the results shown in Table 1 are the averaged values.

The metric used is the Area Under the Curve(AUC). Note that AUC score is generally low in these scenes because the estimator has to both correctly detect *and* estimate pose to be counted as true positive. Both LCHF[21] and sparse auto-encoder[9] are the current state-of-the-art single shot-based object pose estimators which were tested against this scenario.

Table 1 shows that the performance of pose estimator steady increases as more views are combined, regardless of choice of the baselines or scenario. Compared to the single view, the performance increases by 40% on average. The result is not obvious, as our evaluation method disadvantages multi-view result by having to correctly estimate more object than the single view as described in section 4.1. It is also notable that averaging hypotheses fails for juice box scenario. It can be reasoned that many false positive hypotheses with high confidence causes pose estimation to fail.

4.3. Correlation between visibility and viewpoint entropy

Experiments are designed to verify the correlation between the detection performance of single-shot based detector and the viewpoint entropy shown in section 3.2. Synthetic dataset is built for this experiment for the following reasons: More dense and even sampling of camera viewpoint can be obtained; perfect knowledge on calibration parameters, object ground truth, and camera pose are known. To obtain more realistic feasible multiple object pose configuration, 20 object models are randomly thrown into a virtual box using MuJoCo physics engine[22]. RGBD are rendered at 100 evenly sampled viewpoints around upper hemisphere(shown in Fig. 5(c)). For each object, ratio of visible pixel to the total number of pixel if the object were not occluded is calculated and these values of every objects in a scene are averaged to quantify the average visibility score for each viewpoint. Detector[9] is used to obtain the F1 score for each viewpoint and the method described in 3.2 is used to calculate viewpoint entropy. The results are shown in Fig. 7 where every 10th images are sampled. The view indices are ordered in descending average visibility score and the graph shows the F1 score and viewpoint entropy decreases along with the visibility of the viewpoint. The viewpoint entropy and F1 score are positively correlated with the correlation coefficient of 0.6644 for the dataset.

4.4. Next-Best-View Prediction

This section focuses on the effect of different next view strategy with varying number of views and camera moving distance. There are 2 strategies for the next view selection; Furthest view and view entropy based NBV. Furthest view strategy, as the name implies, selects the furthest possible view from the current camera position. It is a popular choice in the literature for comparison[9][15]. Random selection is used to provide the performance comparison when there is no next view selection strategy. The results are shown in Fig.6 and Table2. Entropy based NBV work well in coffee cup scenario regardless of the baselines used. However it is notable that Juice box scenario has no meaningful difference in performance between different strategies. It can be reasoned that coffee cup scenario is much more complex as it has more objects and many of them are occluded in different views. In contrast, objects in juice box scenario are visible in most of the views. To support this claim, the standard deviation of the view entropy of the views in juice box scenario is 0.0899, which is comparably less than coffee cup scenario which is 0.1386.

5. Conclusion

This paper presents a complete active system from hypotheses accumulation to Next-Best-View prediction and demonstrated the performance in a crowded scenario. We have shown that combining multiple views increases the detection and pose estimation performance regardless of baseline and the scenarios. We also introduce view entropy, which can be used to predict the NBV in an environment

where robot movement is costly and the scene is complex.

References

[1] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze. A Global Hypotheses Verification Method for 3D Object Recognition. In *ECCV 2012*, pages 511–524, 2012. 4

[2] S. Biasotti, I. Pratikakis, U. Castellani, T. Schreck, A. Godil, and R. Veltkamp. Sketch-based 3d model retrieval by viewpoint entropy-based adaptive view clustering. 2013. 3

[3] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D object pose estimation using 3D object coordinates. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8690 LNCS, pages 536–551, 2014. 3, 4

[4] S. L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3):267–278, 1994. 2, 4, 5

[5] A. Coates and A. Y. Ng. Multi-camera object detection for robotics. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 412–419, 2010. 1, 2

[6] A. Collet and S. S. Srinivasa. Efficient multi-view object recognition and full pose estimation. In *2010 IEEE International Conference on Robotics and Automation*, pages 2050–2055. IEEE, 5 2010. 1, 2

[7] D. Jayaraman and K. Grauman. Look-Ahead Before You Leap: End-to-End Active Recognition by Forecasting the Effect of Motion. In *ECCV 2016*, volume 9905, pages 35–35, 2016. 1, 2

[8] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. 6D Object Detection and Next-Best-View Prediction in the Crowd. In *ArXiv*, pages 3583–3592. IEEE, 6 2015. 3

[9] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3583–3592. IEEE, 6 2016. 1, 2, 3, 4, 5, 6, 7

[10] . Erkent, D. Shukla, and J. Piater. Integration of Probabilistic Pose Estimates from Multiple Views. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII*, pages 154–170. Springer International Publishing, Cham, 2016. 1, 2

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[13] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 858–865, 2011. 3, 4

[14] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2257–2264, 2010. 3

[15] E. Johns, S. Leutenegger, and A. J. Davison. Pairwise Decomposition of Image Sequences for Active Multi-view Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3813–3822. IEEE, 6 2016. 2, 3, 7

[16] S. H. Kasaei, M. Oliveira, G. H. Lim, L. S. Lopes, and A. M. Tomé. Interactive open-ended learning for 3d object recognition: An approach and experiments. *Journal of Intelligent & Robotic Systems*, 80(3-4):537–553, 2015. 4

[17] M. Mauro, H. Riemenschneider, A. Signoroni, R. Leonardi, and L. Van Gool. A unified framework for content-aware view selection and planning through view importance. In *Proceedings BMVC 2014*, pages 1–11, 2014. 3

[18] W. Mustafa, N. Pugeault, and N. Kruger. Multi-view object recognition using view-point invariant shape relations and appearance information. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4230–4237, 5 2013. 1, 2

[19] M. Oliveira, L. S. Lopes, G. H. Lim, S. H. Kasaei, A. M. Tom, and A. Chauhan. 3d object perception and perceptual learning in the race project. *Robotics and Autonomous Systems*, 75:614 – 626, 2016. 4

[20] W. Susanto, M. Rohrbach, and B. Schiele. 3D Object Detection With Multiple Kinects. *Proc. European Conference on Computer Vision ({ECCV}'2012)*, pages 1–10, 2012. 1, 2

[21] A. Tejani, D. Tang, R. Kouskouridas, and T. K. Kim. Latent-class Hough forests for 3D object detection and pose estimation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8694 LNCS, pages 462–477. Springer Verlag, 2014. 1, 3, 5, 6, 7

[22] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012. 7

[23] F. Vikstén, R. Söderberg, K. Nordberg, and C. Perwass. Increasing pose estimation performance using multi-cue integration. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2006, pages 3760–3767, 2006. 1, 2, 3, 4, 5, 7

[24] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 3

[25] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker Jr, A. Rodriguez, and J. Xiao. Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. *arXiv preprint arXiv:1609.09475*, 2016. 1, 3