Incremental Multi-view Object Detection from a Moving Camera

Takashi Konno
AIST, Japan

Ayako Amma
Toyota Motor Corporation, Japan

Asako Kanezaki
AIST, Japan

ABSTRACT

Object detection in a single image is a challenging problem due to clutters, occlusions, and a large variety of viewing locations. This task can benefit from integrating multi-frame information captured by a moving camera. In this paper, we propose a method to increment object detection scores extracted from multiple frames captured from different viewpoints. For each frame, we run an efficient end-to-end object detector that outputs object bounding boxes, each of which is associated with the scores of categories and poses. The scores of detected objects are then stored in grid locations in 3D space. After observing multiple frames, the object scores stored in each grid location are integrated based on the best object pose hypothesis. This strategy requires the consistency of object categories and poses among multiple frames, and thus it significantly suppresses miss detections. The performance of the proposed method is evaluated on our newly created multi-class object dataset captured in robot simulation and real environments, as well as on a public benchmark dataset.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; *Scene understanding*; *Vision for robotics.*

KEYWORDS

object detection, 3d object recognition, neural networks

ACM Reference Format:
Takashi Konno, Ayako Amma, and Asako Kanezaki. 2020. Incremental Multi-view Object Detection from a Moving Camera . In *Singapore '20: ACM Multimedia Asia, December 16–18, 2020, Singapore.*

1 INTRODUCTION

Real-time object detection is fundamental to realizing autonomous cars and automobile robots. Recent remarkable advances in deep neural networks enabled fast and highly accurate multi-class object detection in images. Although there are also large advances in 3D sensing and processing techniques, using 2D images for object detection is still promising due to the efficiency and the use of large-scale data. Object detection in 2D images, however, suffers from the lack of spatial consistency over multiple frames. Object appearance in a single image significantly changes due to illumination change, occlusion, and viewpoint variance. The object detection results therefore tend to vary from frame to frame. Assuming that most objects in a scene are static, it is unlikely that an object in a certain place suddenly disappears or changes its category or pose. Therefore, by taking the spatial consistency into account, the reliability of the detection results can be further improved.

In this paper, we propose a method to integrate object detection results from multiple frames captured by a moving camera. We assume that the camera position and the distances between the camera and objects can be estimated by *e.g.* a depth sensor, visual SLAM, or robot odometry. Then we associate the object detection results from multiple frames according to their estimated 3D locations. The key idea of our method is to adaptively integrate the multi-view information of objects based on geometry consistency. Suppose, for example, that an object is observed from two different viewpoints, between which the angle is 90° in the azimuth plane. If the object is classified as the front view of a car in an image and is classified as the side view of a car in the other image, the object is most likely a car. On the other hand, if the object is classified as "car" in an image and "bicycle" in the other image, it can be said that the detection results are unreliable. If the object is classified as the front view of a car in an image and is classified as the back view of a car in the other image, the likelihood of the object being a car should be lower than the first described case, because the estimated pose of the object is inconsistent. In order to accomplish such a prediction based on geometry consistency, we train *viewpoint-aware* object detectors in 2D images. We also present a new method to utilize the viewpoint-aware object detection results to select the best candidate among multiple object pose hypotheses.

2 RELATED WORK

2D object detection using 3D information: In the scenario of object detection in autonomous driving systems, many works are based on the combination of object detection in 2D images and LiDAR or stereo-camera 3D data. In order to extract region proposals, the 3D data are converted to voxels [7, 45], depth images [8], or bird's-eye view images [9, 24, 38]. The 3D information is then integrated with 2D networks for object detection. Well-known object detectors such as Fast R-CNN [20], Faster R-CNN [36], and YOLO [35] (or similar alternative networks) are often used in the pipeline. Regarding the 2D object detection part, most works take a single RGB image as input to networks, while the main motivation of our work is to use multi-frame information in order to improve the performance of 2D object detection.

3D object detection and semantic mapping: Semantic mapping is one of the most fundamental problems in computer vision. Defacto standard approaches such as sliding shapes [39] use 3D voxel data to train 3D object detectors. In the context of semantic mapping, object recognition results are often utilized to improve the performance of structure from motion (SfM) [3, 4, 19], bundle

Singapore '20, December 16–18, 2020, Singapore

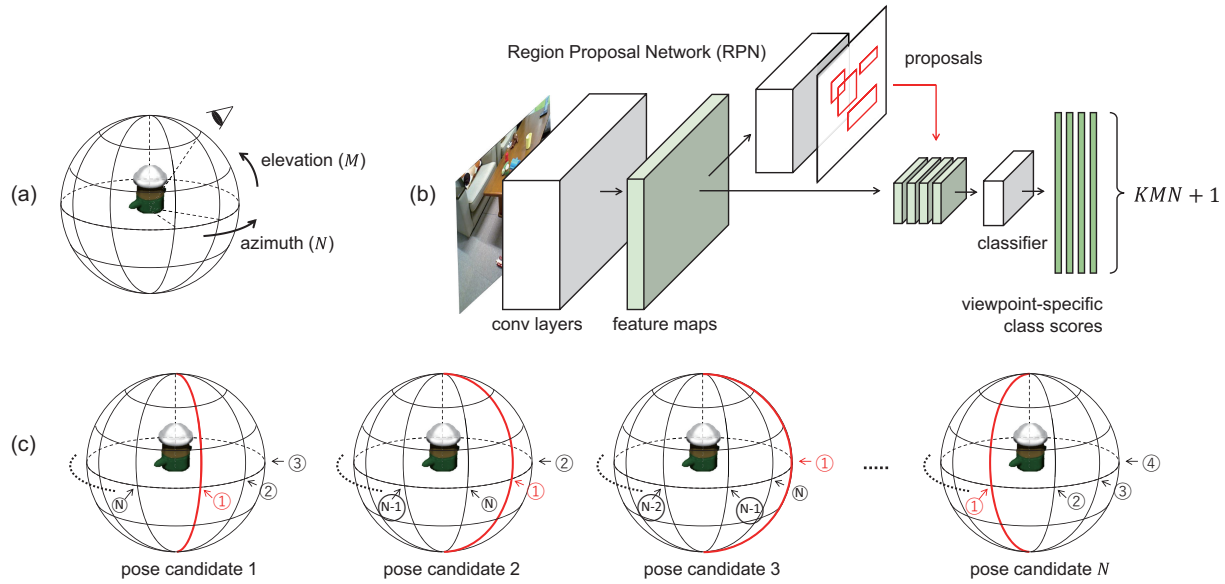


Figure 1: Illustration of (a) viewpoint setup, (b) CNN architecture, and (c) object pose candidates.

adjustment [16, 17], and simultaneous localization and mapping (SLAM) [18, 31, 37, 42]. The main concept of these works is to consider semantic consistency as a constraint of geometry estimation. Likewise, semantic estimation can benefit from geometry consistency. Pillai *et al.* [34] proposed "SLAM-aware" object recognition, where frame-by-frame object detection results are aggregated and refined using the estimated camera trajectory. This method, however, does not take the object pose consistency across multiple frames into account. Bao *et al.* [5] proposed object co-detection, which utilizes multiple images to simultaneously detect an object and find the correspondences among the images. Our work is highly inspired by those works and use multiple images as well as the viewpoint transformation to detect objects by estimating their poses.

Viewpoint estimation in images: In general, object appearance changes significantly when the viewpoint changes. Viewpoint estimation from a single image is widely tackled by training real images [2, 15, 28, 46, 49], synthetic images [21, 23, 41, 43], and the combination of them [13]. These methods are classified in two categories: regression-based methods that predict continuous pose values [15, 21, 46] and classification-based methods that predict discrete viewpoints [13, 23, 28, 41]. The former approach is able to estimate precise poses of objects, whereas the latter approach is more reliable and stable. Some works address the viewpoint estimation problem in conjunction with the object detection task [13, 21, 41, 43, 46]. In particular, Su *et al.* [41], Xiang *et al.* [46], and Divon *et al.* [13] proposed end-to-end CNN models that jointly output the category likelihood and viewpoint information of objects in images. Our work also trains a single CNN model that jointly estimates object category and viewpoint.

Multi-view object classification: Object classification is known to be significantly improved by using multi-view images compared to the case with single image input. Thomas *et al.* [44] extracted multi-view correspondences and transfer votes across views for generic object classification. MVCNN [40] is the first work that used multi-view images as the input of a CNN to classify 3D objects. RotationNet [22], which showed the current state-of-the-art results on a 3D object classification benchmark dataset "ModelNet" [1], also used multi-view images to jointly estimate object categories and poses. Multi-view images are used not only for object classification but also for semantic segmentation [10] and object detection [25, 30]. Kumar *et al.* [25] used hough forest based object detectors to integrate the detection process across multiple frames of a short video sequence. More recently, Li *et al.* [30] proposed a new framework that uses multi-view images of multiple classes to train a CNN that infers the 6-DoF pose of an object. Note that the scope of their work is the pose estimation of multi-class objects, but it does not include the object classification task. In contrast, our work tackles multi-class object detection and pose estimation using multi-view images. Bertasius *et al.* [6] proposed object detection for multi-frames using temporal relationships. Whereas their scope is object detection in videos from fixed cameras, our method aims object detection using multi-views captured by a moving camera. Please refer to [48] for a comprehensive review of object detection in video images captured by a moving camera.

3 METHOD

In this section, we describe the proposed method that trains a CNN and performs multi-view object detection. The core idea of the proposed method is to estimate the viewpoints as well as the categories of detected objects and to integrate the object detection scores by predicting their best poses. Our CNN takes a single image as input and outputs multiple bounding boxes of objects. When a camera moves and captures two or more image frames, the detected bounding boxes are incrementally integrated with the bounding boxes detected in previous frames. The proposed system requires the 3D locations of a camera and observed objects predicted by a SLAM

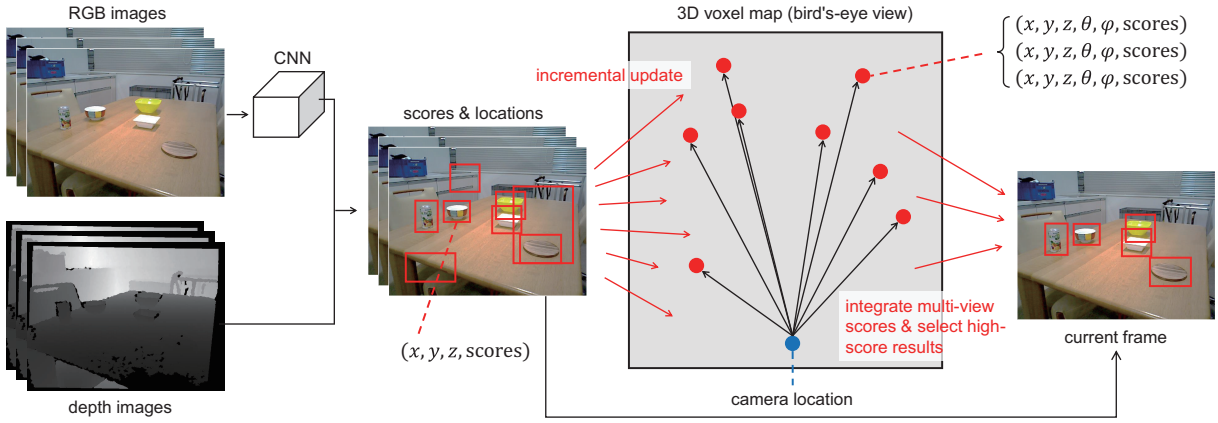


Figure 2: Inference process. Using a depth frame and estimated camera location, each detected box is associated to a 3D location, which is discretized into 3D grid. Object scores in the same voxel grid are integrated in the manner described in the main text.

technique. In this purpose, we use an RGB-D camera[1] equipped to a mobile robot. The viewpoint setup, our CNN architecture, and the training and inference processes are described below.

3.1 Viewpoint Setup

Figure 1 (a) shows the viewpoint setup in this work. We assume the upright orientations of objects are fixed. When an image of an object is input, we estimate the azimuth and elevation levels of the discrete viewpoint. The candidate viewpoints are equally distributed in azimuth and elevation, respectively. Letting M and N be the numbers of viewpoints in elevation and azimuth, the total number of viewpoints we consider is MN. Figure 1 (c) shows the illustration of the candidates of an object pose. We consider the case where objects are rotated around the gravity vector, and thus the total number of discrete object poses is N.

3.2 CNN Architecture

We train an end-to-end CNN that takes an image as input and outputs the bounding boxes of objects with the likelihood of categories and viewpoints. Each element of the output vector corresponding to a bounding box represents the likelihood of an object category observed from a viewpoint. We simply modify the number of the final output layer of a certain existing object detector. In this work, we used Faster R-CNN [36] for the base architecture. It is worth noting that other 2D object detectors such as YOLO [35] can be used as an alternative. Figure 1 (b) shows our CNN based on the Faster R-CNN architecture. Letting K denote the number of the target object categories, the dimension of a final output vector is $KMN + 1$, which includes one element that represents the likelihood of background. We train the CNN with cross-entropy loss, where the elements of an output vector are exclusive. Specifically, the CNN outputs $p^k_{i,j}$ as the likelihood for the k-th object at i-th azimuth level and j-th elevation level and p^{bg} as the likelihood for background, where $\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K} p^k_{i,j} + p^{\text{bg}} = 1$. In a preliminary study, we also tried the geometric loss proposed in [41]. However, we found no meaningful difference in the accuracy, and therefore we decided to use the above mentioned cross-entropy loss.

[1]This could be replaced by mono SLAM because actual depth values are unnecessary in the proposed method.

3.3 Training

It is a non-trivial task to annotate viewpoints of a large number of training images with multi-class objects. In a similar manner to previous works [13, 21, 23, 41, 43], we synthesized training images with the reconstructed 3D models of target objects. First, we rendered each object from all the MN viewpoints using Blender[2] software. Then we synthesized up to 5 randomly picked object images on random background images by using an image synthesis method [14]. This method smooths the boundaries between objects and background in the blending step, which has the effect of giving patch-level reality to the synthetic data. This processing can therefore reduce the influence of artifacts generated by a naive synthesis procedure. In this work, we adopted the Gaussian Blurring, the Poisson Blending [33], and Motion Blurring as smoothing processings. The scale of an object synthesized in an image was randomly selected from 0.1 to 1.0. In each training image, two objects are allowed to overlap with up to 0.75 intersection of union (IoU). The total number of the synthesized training images is approximately 18,000.

For training a CNN based on Faster R-CNN, we used the ResNet101 based model pretrained on the PASCAL VOC 2007 detection task. We fine-tuned it for 100 epochs using momentum SGD with a learning rate of 0.001 and a momentum of 0.9. The learning rate was reduced by a factor of 10 after 8 epochs for optimization.

3.4 Inference

The inference process of the proposed method is illustrated in Fig. 2. Suppose that a camera continuously captures RGB-D images while moving. We apply our CNN frame by frame to detect objects in color images. Here, the detection score threshold is set to a low value in order to reduce false negative results. Using the corresponding depth image, each detected bounding box is associated to the 3D location of its center point relative to the camera. Next, we project

[2]https://www.blender.org

the 3D locations to the world coordinate using the estimated camera location, which are then discretized into 3D grid. Each box is also associated with azimuth and elevation rotations (θ and ϕ), which are calculated using the 3D locations of boxes and the camera. The continuous azimuth and elevation rotation values are then discretized into a ($1 \leq a \leq N$) and e ($1 \leq e \leq M$). Assuming that each voxel grid contains a single object, we integrate multiple bounding box scores associated to the same voxel grid. As already mentioned, each frame is fed into our CNN to extract the per-frame object score $p^k_{a,e}$ which denotes the likelihood that the object in a certain voxel grid observed from the a-th azimuth level and the e-th elevation level belongs to the k-th object category. We set $p^k_{a,e}$ to 0 when there is no object bounding box observed from the a-th azimuth and the e-th elevation viewpoint. The problem here in the inference phase is to estimate the *offset* of the azimuth level from that in the training process (see Fig. 1 (c)), which is denoted by d ($0 \leq d \leq N-1$). The pose of an object observed in the inference phase can be defined as

$$f(a,d) = \begin{cases} a+d & (a+d \leq N) \\ a+d-N & (\text{otherwise}). \end{cases} \tag{1}$$

Finally, we compute the integrated likelihood h^k that the object belongs to the k-th class as

$$h^k = \max_{0 \leq d \leq N-1} \sum_{e=1}^{M} \sum_{a=1}^{N} p^k_{f(a,d),e} \tag{2}$$

which is used to determine whether the corresponding bounding box in the current frame belongs to the k-th class or not using a certain threshold. In this way, the optimal category and pose of the object in each 3D location are jointly estimated. Note that the integrated likelihood h^k becomes high when multi-view scores achieve a consensus on a certain d value (*i.e.*, pose).

4 ROBOT ENVIRONMENTS DATASET

In this section, we describe our newly created robot environments dataset[3]. Our dataset contains multi-view images of multiple objects for training and RGB-D image sequences in several scenes for testing, which are associated with (estimated) camera positions. The scenes were captured in simulation environments as well as real environments. Unlike an egocentric video dataset containing hundreds of objects such as EPIC-Kitchens [11, 12], our dataset contains a small variety of objects which are however densely annotated with pose information. It also has the advantage of pairing simulation data and real data for respective objects.

Our dataset consists of 17 target objects, whose names are shown in the first column of Table 1. We tested two sets of viewpoint parameters: $\{M, N\} = \{3, 16\}$ and $\{M, N\} = \{3, 8\}$. In both cases, the elevation is divided into 10, 30, and 50 degrees from the horizontal. For 48 viewpoint, the azimuth is divided into 16 divisions in 22.5 degree increments. For 24 viewpoint, the azimuth is divided into 8 divisions in 45 degree increments. The background images used to synthesize the training dataset were captured by moving a real/simulated robot. In these experiments, we used the Toyota Human Support Robot (HSR) [47] as a hardware platform.

For the test data, a total of 300 RGB-D images were created in the simulation environment, and a total of 2,113 RGB-D images were created in the real environment, where the camera positions estimated by odometry of HSR for all the frames are included. As a simulation environment, we used four worlds simulating a house environment created on Gazebo[4] and a 3D model of HSR. As a real environment, we prepared three real environments similar to the simulation environment. Note that "spoon_iron" does not appear in real environment dataset because it was lost in that period.

5 RESULTS

In this section, we evaluate the proposed method on our dataset (Sec. 5.1) and RGB-D object dataset [27] (Sec. 5.2). A comparative evaluation of single-view detection (baseline) and multi-view detection (proposed) was performed. As a single-view evaluation, in addition to our base CNN model ("w/ viewpoint"), we also trained a model that classifies only object categories and does not classify viewpoints ("w/o viewpoint"). As a multi-view evaluation, in addition to the proposed method, we evaluated a method that simply sums the scores of multi-frame detections ("Naive"), which was also used in [34]. For the RGB-D object dataset, we also comparad the proposed method with several previous works [26, 29, 34] which used multi-view images for object detection. We used the Average Precision (AP) at IoU of 0.5 to evaluate the performance of object detection. We apply non-maximum suppression (NMS) on detected bounding boxes per category, where we set the IoU threshold to 0.7.

5.1 Results on our robot environments dataset

We describe the experimental results on our new multi-class object datasets.

Columns 2-9 in Table 1 show the AP in simulation environments whereas columns 10-17 show the AP in real environments. In the single-view setting, "w/o view" and "w/ view" performed comparably. The naive method in the multi-view setting is outperformed by the single-view performance. In this dataset, since the robot approaches objects from a distance, the detection accuracy becomes higher in later frames. In the naive method, since the detection scores are simply summed up, we consider that the false detection results in early frames affected the subsequent frames. The proposed method outperformed both the single-view object detection and the naive method in most classes. The possible reason for the performance decrease in some classes is that they are symmetric, and thus the proposed approach based on pose estimation was not fully effective. Figure 3 shows the qualitative results in our robot simulation and real environments dataset. The proposed multi-view based object detection tends to have fewer false positive detections than the single-view based method.

5.2 Results on RGB-D Object Dataset [27]

In this section, we describe the experimental results on RGB-D object dataset [27], which is a well-known public benchmark dataset of multi-view object images. We compare the detection results of the proposed methods with existing methods [26, 29, 34] using multi-view images. The RGB-D object dataset contains the images

[3]The dataset is publicly available on https://www.ak.c.titech.ac.jp/projects/IMOD/

[4]http://gazebosim.org

Table 1: Comparison of Average Precision (AP) with the single-view based approach (baseline) and the multi-view based approach (proposed) for our robot simulation and real environments dataset.

	Robot Simulation Environments Dataset								Real Environments Dataset							
	$M = 3,\ N = 16$				$M = 3,\ N = 8$				$M = 3,\ N = 16$				$M = 3,\ N = 8$			
	Single-View		Multi-View		Single-View		Multi-View		Single-View		Multi-View		Single-View		Multi-View	
Object Classes	w/o view	w/ view	Naive	Ours	w/o view	w/ view	Naive	Ours	w/o view	w/ view	Naive	Ours	w/o view	w/ view	Naive	Ours
ball_pink	0.907	**0.909**	**0.909**	**0.909**	**0.909**	**0.909**	**0.909**	**0.909**	0.631	**0.711**	0.459	0.592	**0.718**	0.499	0.526	0.548
building_block_house	**0.909**	0.723	0.906	0.778	**0.906**	0.862	**0.906**	0.903	**0.907**	0.900	0.696	0.843	0.816	0.815	0.759	**0.926**
cellphone	0.872	0.740	**0.996**	0.870	0.996	0.894	0.961	**1.000**	0.522	0.393	0.414	**0.640**	0.540	0.417	0.545	**0.652**
gorilla_doll	**0.909**	0.894	0.906	**0.909**	**0.961**	0.877	0.727	0.903	0.499	0.487	**0.744**	0.716	0.373	0.485	0.622	**0.729**
kidney_beans_doll	0.495	0.724	0.859	**0.958**	0.424	0.621	0.602	**0.909**	0.534	0.515	0.456	**0.699**	0.442	0.409	0.555	**0.627**
lego_green	**0.887**	0.622	0.412	0.761	**0.818**	0.812	0.633	0.808	0.628	0.536	0.541	**0.726**	0.527	0.528	0.544	**0.585**
lego_red	0.808	0.755	**0.909**	0.878	0.818	0.804	**0.904**	0.876	0.813	0.795	0.664	**0.863**	0.815	0.696	0.651	**0.817**
mouse_doll	0.456	0.689	0.434	**0.805**	0.425	0.651	0.264	**0.821**	0.369	0.451	0.374	**0.579**	0.346	0.480	0.476	**0.596**
plastic_cup_yellow	0.727	0.725	0.727	**0.751**	0.687	0.722	0.630	**0.781**	**0.907**	0.892	0.651	0.778	**0.907**	0.782	0.544	0.676
rail_bridge	0.818	0.818	0.818	**0.831**	0.871	0.889	**0.909**	**0.909**	0.816	0.800	0.834	**0.852**	0.726	0.708	0.741	**0.789**
rice_bowl	0.883	**0.896**	0.764	0.882	0.810	**0.892**	0.674	0.850	0.909	0.893	0.835	**0.991**	0.908	0.906	0.754	**0.923**
spoon_iron	0.622	0.671	0.625	**0.741**	0.687	0.623	**0.701**	0.681	-	-	-	-	-	-	-	-
square_bowl	**0.907**	0.863	0.816	0.900	**0.904**	0.857	0.693	**0.904**	**0.908**	0.889	0.845	0.889	**0.907**	0.895	0.736	0.836
square_dish	**0.902**	0.849	0.771	0.830	0.868	**0.904**	0.859	0.863	0.538	0.444	0.652	**0.787**	0.263	0.196	**0.566**	0.477
steel_juice	0.720	0.695	0.455	**0.723**	0.818	0.811	0.273	**0.834**	0.387	0.383	0.459	**0.630**	0.368	0.302	0.518	**0.642**
teacup	0.906	0.801	0.894	**0.909**	**0.909**	0.886	**0.909**	**0.909**	**0.904**	0.756	0.452	0.695	**0.906**	0.691	0.587	0.632
wood_coaster	0.273	0.329	0.358	**0.541**	0.482	0.501	0.273	**0.545**	0.635	0.486	0.754	**0.871**	0.545	0.530	0.755	**0.830**
mean	0.765	0.747	0.739	**0.822**	0.782	0.795	0.696	**0.847**	0.682	0.646	0.614	**0.759**	0.632	0.584	0.617	**0.705**

Table 2: Comparison of Precision/Recall (AP) with the single-view based and multi-view based approaches on RGB-D Object Dataset [27].

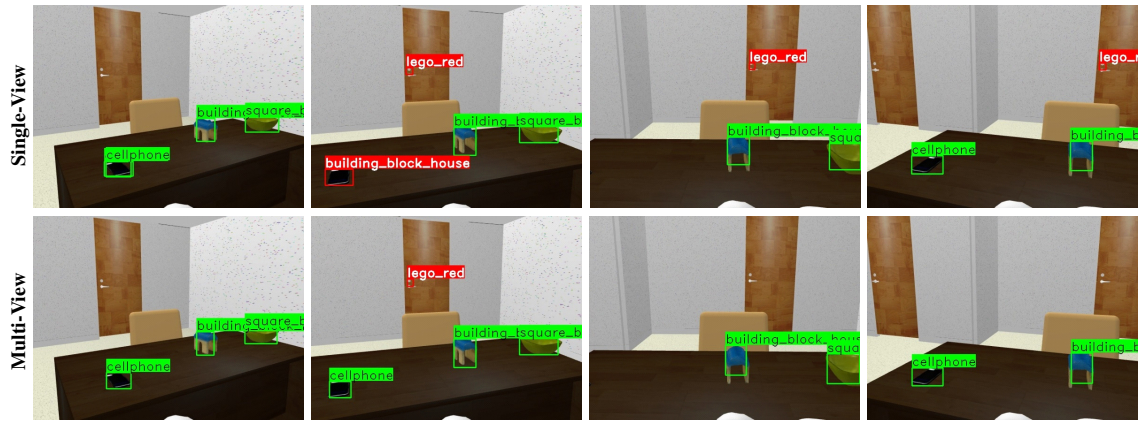
	Single-View				Multi-View				
Object	w/o viewpoint	w/ viewpoint	DetOnly [29]	SLAM-aware [34]	Naive	Ours	Det3DMRF [29]	HMP2D+3D [26]	SLAM-aware [34]
bowl	96.3/65.8 (0.631)	92.5/61.0 (0.606)	46.9/**90.7** (-)	88.6/71.6 (-)	94.6/66.0 (0.613)	**97.0**/68.1 (**0.709**)	91.5/85.1 (-)	**97.0**/89.1 (-)	88.7/70.2 (-)
cap	84.4/69.1 (0.626)	88.0/61.9 (0.599)	54.1/90.5 (-)	85.2/62.0 (-)	82.2/69.2 (0.564)	90.1/69.2 (**0.689**)	90.5/91.4 (-)	82.7/**99.0** (-)	**99.4**/72.0 (-)
cereal_box	94.3/74.4 (0.724)	93.3/71.3 (0.709)	76.1/90.7 (-)	83.8/75.4 (-)	92.4/76.0 (0.711)	94.9/76.1 (**0.802**)	93.6/94.9 (-)	**96.2**/**99.3** (-)	95.6/84.3 (-)
coffee_mug	93.0/79.0 (0.721)	83.2/80.6 (0.719)	42.7/74.1 (-)	70.8/50.8 (-)	91.1/79.4 (0.775)	**93.9**/79.4 (**0.795**)	90.0/75.1 (-)	81.0/**92.6** (-)	80.1/64.1 (-)
soda_can	96.8/65.4 (0.631)	96.0/66.5 (0.627)	51.6/87.4 (-)	78.3/42.0 (-)	93.4/66.7 (0.612)	**97.8**/72.3 (**0.723**)	81.5/87.4 (-)	97.7/**98.0** (-)	89.1/75.6 (-)
background	90.8/96.5 (-)	90.4/95.2 (-)	98.8/93.9 (-)	95.0/90.0 (-)	89.6/96.8 (-)	91.6/97.1 (-)	**99.0**/**99.1** (-)	95.8/95.0 (-)	96.6/96.8 (-)
mean	92.6/75.0 (0.667)	90.6/72.8 (0.652)	61.7/87.9 (-)	81.5/59.4 (-)	90.55/75.68 (0.655)	**94.2**/77.0 (**0.744**)	91.0/88.8 (-)	90.9/**95.6** (-)	89.8/72.0 (-)

of 300 objects in 51 categories taken from multiple viewpoints. In order to maintain a fair comparison, we used five objects shown in the first column of Table 2 for evaluation as in [26, 29, 34]. We divided the azimuth into 12 divisions in 30 degree increments and the elevation into 3 divisions in 30, 45, and 60 degrees with the horizon for a total of 36 viewpoints. The RGB-D object dataset also contains the RGB-D scenes dataset, which consists of eight video sequences (1,437 frames) of a house environments. As in [34], we estimated the 3D position of the camera in each frame using ORB-SLAM2 [32].

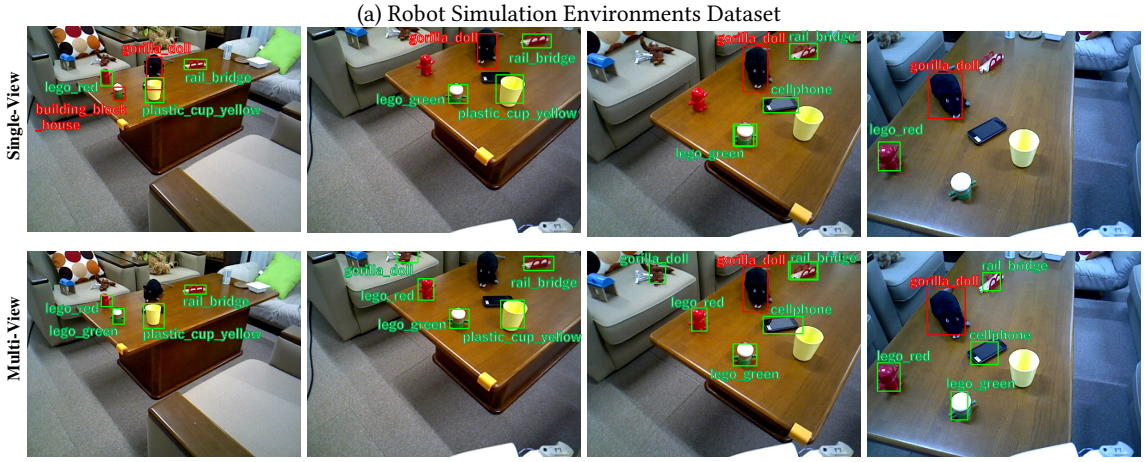
Table 2 shows the precision/recall values and the AP for the object detection task. For the AP, interestingly, even though the single-view detection model that estimates both category and viewpoint was inferior to the one w/o viewpoint, the results were overthrown when multiple frames were used. The naive method does not effectively utilize multi-view information for the same reason as described in Sec. 5.1. Point-wise precision/recall values are reported for [29] and [26], whereas those for 2D object detection per frame are reported for [34] and the proposed method. The score threshold for each class was tuned by increasing it with the step size of 0.01. The precision/recall of the background class was computed per pixel by regarding all the pixels in detected bounding boxes as foreground and all the other pixels as background. The proposed method outperformed the single-view methods and SLAM-aware [34] using multi-view images in most classes. The proposed method has higher precision but slightly lower recall than Det3DMRF [29] and HMP2D+3D [26], both of which require 3D point clouds (*i.e.*, more information than 2D images). Figure 4 shows the qualitative results. Similarly to the results on our dataset described in Sec. 5.1, the proposed multi-view based object detection has fewer false positive detections as well as more true positive detections than the single-view based method.

6 CONCLUSION

In this paper, we presented a new framework that integrates multi-frame information to improve multi-class object detection performance. Using the depth and the camera location information, the bounding boxes detected in 2D images are associated with the 3D locations. Our system incrementally refines the object detection results by integrating the scores extracted from multi-view images in each 3D location. The key idea of the proposed method is to predict the pose of an object when integrating the object detection scores. Owing to the *geometry constraint* underlying in multiple frames, our system can appropriately increase the scores of plausible results as well as suppress miss detections. Experimental results both in robot simulation environments and real environments demonstrated



(a) Robot Simulation Environments Dataset



(b) Real Environments Dataset

Figure 3: Qualitative results on our dataset. True positives are shown in green boxes and false positives are shown in red boxes. The estimated class name is displayed above each bounding box. Best viewed in color.

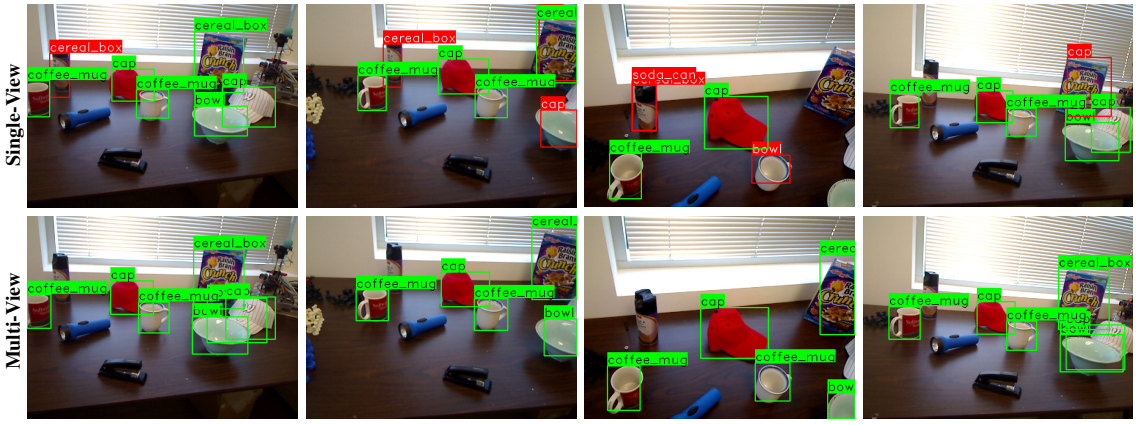


Figure 4: Qualitative results for RGB-D Object Dataset [27]. True positives are shown in green boxes and false positives are shown in red boxes. The estimated class name is displayed above each bounding box. Best viewed in color.

that the proposed method outperformed frame-by-frame object detection. Also, we showed that the proposed method achieved higher precision than several previous works on multi-view object detection on a public benchmark dataset.

ACKNOWLEDGMENT

The authors would like to thank Kazuto Murase for his support in the experiments using HSR and also for his valuable discussion.

REFERENCES

[1] [n.d.]. The Princeton ModelNet. http://modelnet.cs.princeton.edu/.

[2] Amr Bakry and Ahmed Elgammal. 2014. Untangling object-view manifold for multiview recognition and pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[3] Sid Yingze Bao, Mohit Bagra, Yu-Wei Chao, and Silvio Savarese. 2012. Semantic Structure from Motion with Points, Regions, and Objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] Sid Yingze Bao and Silvio Savarese. 2011. Semantic Structure from Motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Sid Yingze Bao, Yu Xiang, and Silvio Savarese. 2012. Object Co-detection. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[6] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. 2018. Object Detection in Video with Spatiotemporal Sampling Networks. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 2016. Monocular 3d object detection for autonomous driving. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 2015. 3d object proposals for accurate object class detection. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. 2017. Multi-view 3d object detection network for autonomous driving. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Angela Dai and Matthias Nießner. 2018. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2020. Rescaling Egocentric Vision. *CoRR* abs/2006.13256 (2020).

[12] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[13] Gilad Divon and Ayellet Tal. 2018. Viewpoint Estimation—Insights & Model. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[14] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[15] Mohamed Elhoseiny, Tarek El-Gaaly, Amr Bakry, and Ahmed Elgammal. 2016. A Comparative Analysis and Study of Multiview CNN Models for Joint Object Categorization and Pose Estimation. In *Proceedings of International Conference on Machine Learning (ICML)*.

[16] N. Fioraio and L. Di Stefano. 2013. Joint Detection, Tracking and Mapping by Semantic Bundle Adjustment. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[17] D. P. Frost, O. Kähler, and D. W. Murray. 2016. Object-aware bundle adjustment for correcting monocular scale drift. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

[18] Dorian Gálvez-López, Marta Salas, Juan D. Tardás, and J.M.M. Montiel. 2016. Real-time monocular object SLAM. *Robotics and Autonomous Systems* 75 (2016), 435 – 449.

[19] P. Gay, V. Bansal, C. Rubino, and A. D. Bue. 2017. Probabilistic Structure from Motion with Objects (PSfMO). In *Proceedings of International Conference on Computer Vision (ICCV)*.

[20] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[21] Omid Hosseini Jafari, Siva Karthik Mustikovela, Karl Pertsch, Eric Brachmann, and Carsten Rother. 2018. iPose: instance-aware 6D pose estimation of partly occluded objects. In *Proceedings of Asian Conference on Computer Vision (ACCV)*.

[22] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. 2018. RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. 2017. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[24] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. 2018. Joint 3d proposal generation and object detection from view aggregation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[25] Shyam Sunder Kumar, Min Sun, and Silvio Savarese. 2012. Mobile object detection through client-server based vote transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] Kevin Lai, Liefeng Bo, and Dieter Fox. 2014. Unsupervised feature learning for 3D scene labeling. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/ICRA.2014.6907298

[27] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. 2011. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

[28] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. 2011. A Scalable Tree-based Approach for Joint Object and Pose Recognition. In *Proceedings of AAAI Conference on Artificial Intelligence*.

[29] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. 2012. Detection-based Object Labeling in 3D Scenes. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 1330–1337. https://doi.org/10.1109/ICRA.2012.6225316

[30] Chi Li, Jin Bai, and Gregory D. Hager. 2018. A Unified Framework for Multi-view Multi-class Object Pose Estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[31] John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. 2018. Fusion++: Volumetric Object-Level SLAM. In *Proceedings of International Conference on 3D Vision (3DV)*.

[32] Raúl Mur-Artal and Juan D. Tardós. 2017. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics* 33, 5 (2017), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

[33] Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Transactions on graphics (TOG)* 22, 3 (2003), 313–318.

[34] Sudeep Pillai and John Leonard. 2015. Monocular SLAM Supported Object Recognition. In *Proceedings of Robotics: Science and Systems (RSS)*.

[35] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

[37] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. 2013. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[38] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. 2018. Complex-YOLO: an Euler-region-proposal for real-time 3D object detection on point clouds. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[39] Shuran Song and Jianxiong Xiao. 2016. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[40] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[41] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. 2015. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[42] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. 2017. Meaningful maps with object-oriented semantic mapping. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[43] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. 2018. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of European Conference on Computer Vision (ECCV)*.

[44] Alexander Thomas, Vittorio Ferrar, Bastian Leibe, Tinne Tuytelaars, Bernt Schiel, and Luc Van Gool. 2006. Towards multi-view object class detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[45] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. 2015. Data-driven 3d voxel patterns for object category recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[46] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. 2018. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proceedings of Robotics: Science and Systems (RSS)*.

[47] Takashi Yamamoto, Koji Terada, Akiyoshi Ochiai, Fuminori Saito, Yoshiaki Asahara, and Kazuto Murase. 2019. Development of Human Support Robot as the research platform of a domestic mobile manipulator. *ROBOMECH Journal* 6, 1 (2019). https://doi.org/10.1186/s40648-019-0132-3

[48] Mehran Yazdi and Thierry Bouwmans. 2018. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review* 28 (2018), 157–177.

[49] Haopeng Zhang, Tarek El-Gaaly, Ahmed M Elgammal, and Zhiguo Jiang. 2013. Joint Object and Pose Recognition Using Homeomorphic Manifold Analysis. In *Proceedings of AAAI Conference on Artificial Intelligence*.