Taylor & Francis
Taylor & Francis Group

Advanced Robotics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tadr20

Automatic data collection for object detection and grasp-position estimation with mobile robots and invisible markers

Suraj Prakash Pattar, Thomas Killus, Tsubasa Hirakawa, Takayoshi Yamashita, Tetsuya Sawanobori & Hironobu Fujiyoshi

To cite this article: Suraj Prakash Pattar, Thomas Killus, Tsubasa Hirakawa, Takayoshi Yamashita, Tetsuya Sawanobori & Hironobu Fujiyoshi (2022): Automatic data collection for object detection and grasp-position estimation with mobile robots and invisible markers, Advanced Robotics, DOI: 10.1080/01691864.2022.2136504

To link to this article: https://doi.org/10.1080/01691864.2022.2136504

ADVANCED ROBOTICS
https://doi.org/10.1080/01691864.2022.2136504

FULL PAPER

Automatic data collection for object detection and grasp-position estimation with mobile robots and invisible markers

Suraj Prakash Pattar [a,b], Thomas Killus[a], Tsubasa Hirakawa [b], Takayoshi Yamashita [b], Tetsuya Sawanobori[a] and Hironobu Fujiyoshi [b]

[a]Tokyo University of Agriculture and Technology Koganei Campus, Connected Robotics Inc., Tokyo, Japan; [b]Machine Perception and Robotics Group, Chubu University, Kasugai, Japan

ABSTRACT

Deep convolutional networks have dominated advances in object detection and grasp-position estimation using computer vision. The data-collection process for these networks is, however, time-consuming and expensive. We propose an automatic data-collection method for object detection and grasp-position estimation using mobile robots and invisible markers. Our method offers clear advantages over manual data annotation and synthetic data generation in terms of time consumption, cost, consistency, and similarity to real-world data. We compared data generated with our method against synthetically generated data to show how it can affect the robustness of the deep learning model when inferred under real-world conditions. We also conducted a comparison between our method and manual data-collection and synthetic-data-generation methods and demonstrated how our method could be used for data collection of asymmetric objects and key-point estimation tasks.

ARTICLE HISTORY
Received 1 May 2022
Revised 29 August 2022
Accepted 19 September 2022

KEYWORDS
Computer vision; data collection; annotation; object detection; grasp-position estimation

1. Introduction

Computer vision methods using deep convolutional networks have been influential in solving some of the major problems in robotic manipulation with object detection [1–3], grasp-position-estimation [4–6] and six-dimensional (6D) position estimation [7–9]. These supervised learning methods rely on reliable and high-quality data [6,10].

Data collection is a crucial process for building accurate models that can perform robustly in real environments. However, manual data collection is a time-consuming and expensive process.

The data-collection process for object detection involves two steps. The first is image collection and the second is annotation. In the image-collection step, one needs to ensure that there is a large variation in the position and orientation of the object in the images. Objects placed in the same position in every image can make the model overfit to the particular position and orientation and less robust when it is placed at a different location. In the annotation step, the annotator marks the bounding box of the object in the image to specify the location of the object in the image. This process can be highly laborious depending on the size and shape of the object, the number of objects in the image, and tolerance in annotation accuracy. The annotators need to have domain knowledge of the objects they are annotating in terms of how to distinguish between similar-looking objects. Similarly for grasp-position annotation, one needs to have the domain knowledge of the appropriate grasp locations for each object and knowledge of which end-effectors would be used for grasping.

Certain approaches use outsourcing to tackle this task in which people can be hired to annotate image datasets for a fee [11–14]. This gives rise to another problem, which is quality checks. When the annotation is outsourced, several annotators of varying skill levels are employed. Because the hired annotators have varying levels of skill and can annotate the same objects in different ways, these annotated datasets must be quality checked. Another issue is that confidential data cannot be outsourced to such services for annotation [15]. For accurate object detection, it is also important that the training data be representative of the real-world data the model might encounter, which makes it necessary to add similar noise in the training data as well.

CONTACT Suraj Prakash Pattar pattarsuraj@gmail.com Tokyo University of Agriculture and Technology Koganei Campus, Connected Robotics Inc., 303, Venture Port, 2 Chome-24-16, Nakacho, Koganei, Tokyo 184-0012, Japan; Machine Perception and Robotics Group, Chubu University, 1200 Matsumoto-cho, Kasugai, Aichi 487-8501, Japan

Supplemental data for this article can be accessed here. https://doi.org/10.1080/01691864.2022.2136504

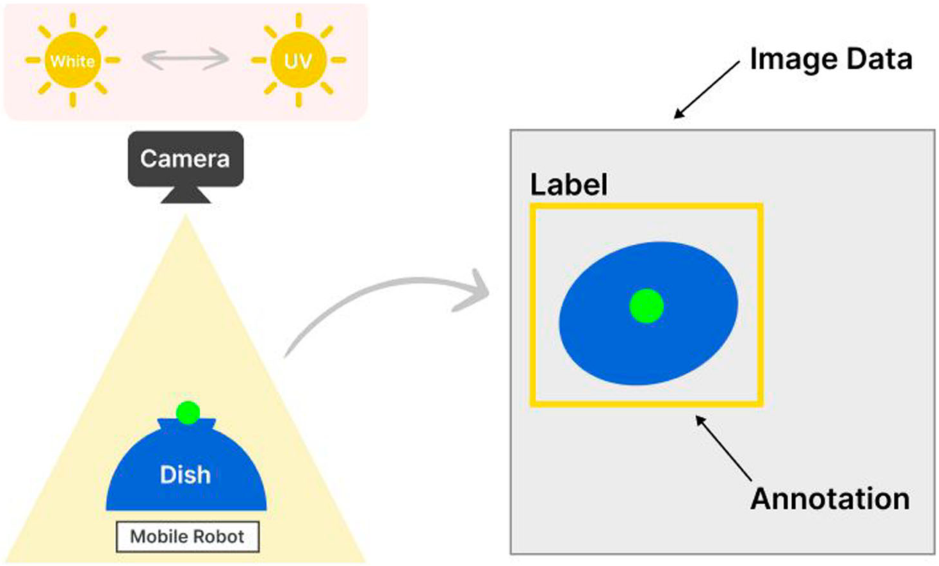

Figure 1. Automatic data collection for object detection and grasp-position estimation with mobile robots and invisible markers.

To solve these problems, we propose the automation of the data collection and annotation process using (1) mobile robots equipped with novel tilt platforms and (2) invisible markers. Mobile robots transport objects from one location to another. Invisible markers visible only under ultraviolet light are used to mark the grasp position of the objects. The concept of our data-collection method is shown in Figure 1.

Object grasping is a crucial step in robotic manipulation in various applications. It entails (1) identifying an object in a given image and (2) determining the position from *where* the robot can dependably grab the object. Objects are of various shapes and sizes in a general grasp estimation problem. Additionally, the scene is typically cluttered and is shot from an arbitrary viewpoint. A successful grasp depends on correctly identifying the object and choosing the proper position to grab it from, depending on the type of end-effector. In an industrial setting, the camera is usually fixed in a top-down view above the objects of interest. The types of objects are limited, and an appropriate end-effector is chosen, which can grab the objects reliably. The scene can also be constrained to increase the rate of successful grasps. For example, to ensure better detection, one might regulate the lighting conditions and only permit objects with specific shapes, like circles. To ensure that the end-effector can reliably grasp the objects, one can also restrict the size and weight of the objects.

We focus on data collection for the object-detection and grasp-position-estimation tasks for a commercial dishwasher system [16]. In our vision system, the camera is fixed in a top-down view above the objects. This is a conventional setup in industrial applications.

Our setup of a commercial dishwasher system is shown in Figure 2. This system has two main objectives: (1) to process numerous dishes in a short amount of time to prevent any bottlenecks in restaurant operation and (2) to pack the dishwasher rack efficiently to maximize the number of dishes washed in each cycle. Our scene is less cluttered than a regular bin-picking problem as it is a commercial human-robot collaborative dishwasher system used to automate dishwashing in restaurants. Maintaining close to real-time constraints in a commercial system is essential to avoid ambiguity and process failure. To achieve the above goals, the dishes are placed upside down by human assistants and the dishes are picked up from the center so as to pack the rack efficiently. In addition, since we use a single suction gripper in our application, the bottom center of the dish is the only position *where* the suction can work without failing.

The symmetric circular shape of the dish allows for a simpler 2D bin-packing method, which is out of the scope of this paper. The dishes need to be classified for sorting after dishwashing and to differentiate which dish needs to be washed with a brush and which one needs to be washed with water only. Although our target objects in this work have a symmetric shape, we show how this method can be used for asymmetric objects as well.

In Section 2, we describe related work and briefly compare them with ours. In Section 3, we give details of the proposed method, data-collection setup along with hardware used, and invisible markers to annotate grasp position. In Section 4, we describe the experimental setup of comparing the data-collection methods and the results. Finally, we provide our conclusion based on the tests. To the best of our knowledge, our proposed method of

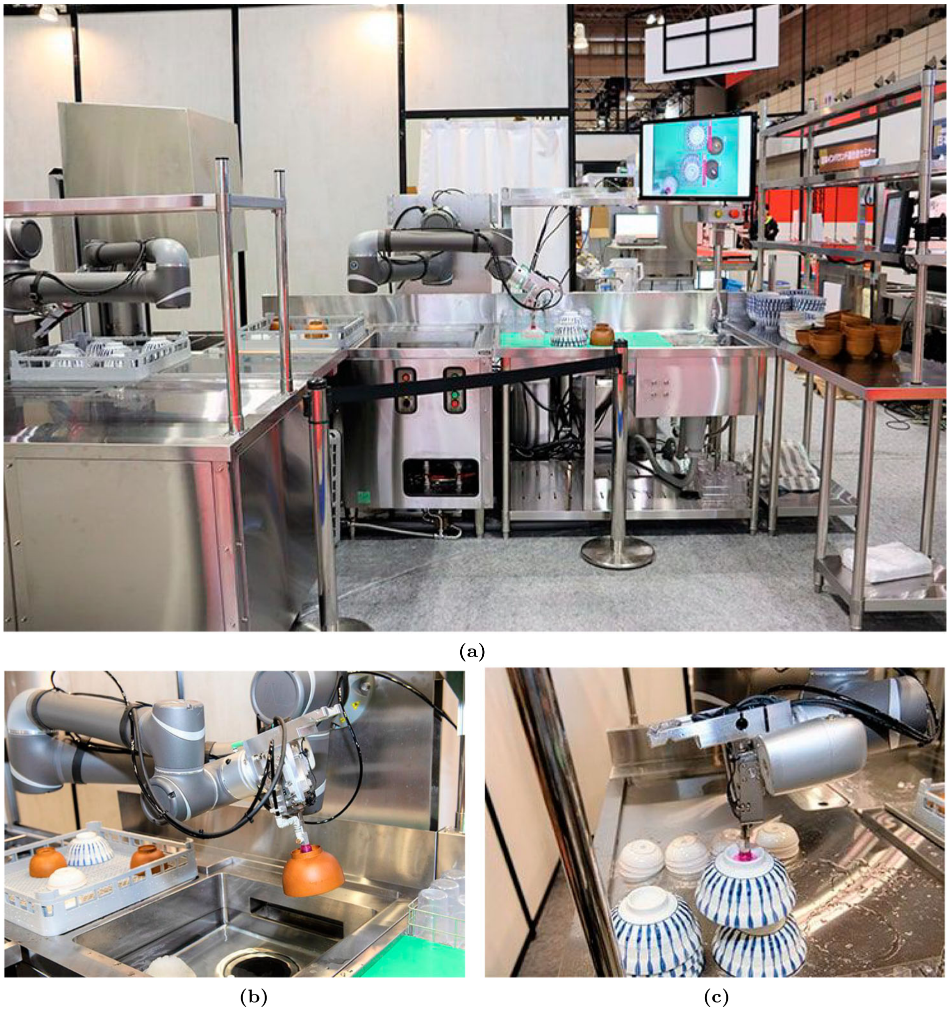

Figure 2. Commercial dishwasher system [16]. (a) Dishwasher system robot setup. (b) Pick and pre-wash dishes before placing them on the rack. (c) Sort and place the washed dishes.

using mobile robots and invisible markers has not been applied before to collect data for object detection and grasp-position estimation.

2. Related work

Manual data collection and annotation have several problems, i.e. time-consuming, expensive, requires skill, and can contain human errors. To overcome these problems, there are several methods of automatically collecting and annotating image datasets. We briefly describe some of them in the following subsections.

2.1. Data collection

2.1.1. Methods using robot arms

Certain methods use manipulator robot arms to automate the process of data collection [17–19]. The manipulator robot arms are used to change the camera position and angle to acquire images of the objects from various perspectives. The objects are stationary throughout the data collection process.

Although the robot arms help in providing diverse perspectives and a consistent dataset compared with manual methods, they are limited by the reach of the robot arms. These methods also require human supervision and are not fully autonomous as the robot arm trajectories can encounter singularities and joint limits while moving in their environment which need to be resolved.

The high cost of a manipulator robot can also be detrimental when used for data collection. Fang et al. and De Gregorio et al. used a Universal Robots UR5 robot arm with an average cost of around 30,000 USD [17,19]. Rennie et al. used a Motoman SDA10F robot, which can cost upto 100,000 USD.

2.1.2. Synthetic-data-generation methods

There are methods that use accurately textured 3D models of the objects to generate a labeled synthetic dataset

[20–23]. These methods use physics simulation environments to change the object position and orientation. The background can also be easily modified and one can use domain randomization to make the model robust against different backgrounds during inference [21].

These methods can be especially helpful for annotating the 6D pose of an object, which is much more complicated compared with 2D annotation. For 6D annotation, one needs to (1) draw a 3D bounding box around the object and (2) indicate the angle of orientation with a normal vector with respect to the camera plane.

Synthetic-data-generation methods require highly accurate 3D models of the objects for high-fidelity data. The drawback of using synthetic data is that the models trained using only synthetic data do not perform well during inference when tested on real data [24]. Poorly constructed 3D models can lead to further increasing the Sim2Real gap. Synthetically generated data have been known to increase detection accuracy when combined with a sparse real dataset [25].

2.2. Annotation

2.2.1. Bounding-box annotation

Deep learning methods and pre-trained object detectors have been used to reduce the annotation time of bounding boxes [26–28]. These methods use object detectors pre-trained on a broad variety of data and later fine-tuned with manually collected data. One major drawback with such methods is that one needs to train an object-detector model first that would be able to accurately annotate the new data.

Certain methods annotate the first image manually, then, using a well-calibrated camera, use camera poses to project the bounding boxes on the rest of the images captured in that sequence [17,19]. De Gregorio et al. used an augmented reality pen to manually outline virtual boxes around the target objects for the first image [17]. The objects in the scene need to be stationary and the camera moved using a robot arm with these methods.

2.2.2. Grasp-position annotation

Annotating the grasp position has been tackled using synthetic-data-generation methods [29,30]. However, as mentioned above, the visual domain gap can lead to low grasping accuracy when inferring on real images.

Other methods combine real-world images with computation in synthetic environments [19,31]. Fang et al. used high-quality mesh models of objects and downsampled them to obtain a large number of grasp candidates. These candidates are then filtered to fit a parallel plate gripper so there would be no collision and no empty grasp.

Although this method generates a large number of possible grasp positions for an object, it does not account for planning the placement once it has been picked up. This is essential in an industrial setting, *where* packing as many objects as possible in a small space is crucial.

Other methods use invisible markers that are visible under ultraviolet (UV) light and invisible under regular (white) light [32,33]. Takahashi et al. used invisible markers for annotating segmentation masks of deformable objects [33].

3. Automatic mobile robot data collection setup

In this section, we describe the system setup of our proposed data-collection method.

3.1. Proposed method

In a fixed camera setting, the view of the object can change considerably as its position with respect to the camera changes. For example, a spherical object, a semi-spherical object, and an inverted semi-spherical object all appear as a circle when the objects' centers align with the principal axis of the camera. However, the appearances differ greatly when they are at the edge of the camera's field of view, i.e. away from the camera center. It can be enormously expensive to manually capture images of objects for every possible position under the camera's field of view.

Our method enables data collection and annotation for both object-detection and grasp-position-estimation tasks by using mobile robots and invisible markers. There have not been any methods that have explored the use of mobile robots for data collection.

Using small mobile robots enables us to automate moving the objects and capturing the object's image at every possible location inside the camera's field of view. Our approach allows greater control over the objects, which in turn allows greater control over the scene. This enables us to randomize the scene and fill in any data gaps in a specific configuration.

Invisible markers enable us to automate the annotation process. They are visible only under UV light and invisible under white light. This enables us to extract the invisible marker's position in UV-light-lit images and annotate the white-light-lit images accordingly. The invisible markers are necessary since we do not want the markers to affect the appearance of the dish. Using visible markers can change the appearance of the dishes, and we would need to use the markers during inference as well, which would not be practical in a commercial dishwasher system.

By adding noise during data collection with random lighting and a fog generator, we are able to generate data close to what robots might encounter in the real world. Since our application is a commercial dishwasher system, we set up the environmental noise as seen at the dishwashing station in a real restaurant.

3.2. Data-collection setup

Our data-collection setup is shown in Figure 3.

3.2.1. Camera and linear guide setup

We use a ZED-M 3D camera, but our method requires only RGB images. The camera has a high capture rate of 30 frames per second. This enables us to capture multiple images of an object under motion while seemingly at the same location.

The camera is placed on a linear guide, which enables us to move the camera vertically to change the height between the platform and camera. Moving the camera on the z-axis continuously during data collection enables us to simulate real conditions under which the camera height to the platform can vary.

3.2.2. Mobile robots

Mobile robots are used to carry objects over a platform under the camera's field of view. We surveyed multiple mobile robots and selected E-puck2 and Khepera IV robots for their capabilities of accurate locomotion and control with Bluetooth or Wifi.

Khepera IV is able to bear a payload of up to 2 kg, which is sufficient to carry stacks of dishes. Although E-puck2 does not have a specific payload capacity, it is capable of carrying objects of weights up to 150 g.

The size of the robot was also an important factor since we needed the robots to be hidden in the top-down view of the camera. Figure 4 shows the mobile robots we used along with the dishes. The robots were programmed to cover every position on the platform. We used guard rails to prevent the mobile robots from falling off the platform.

3.2.3. Invisible markers and lighting setup

We used the same invisible ink mentioned in a previous study [33] to mark the grasp position of the object in our application.

Our lighting environment consisted of USB-powered RGB light-emitting diodes (LEDs) for the white-light environment, and UV LEDs for the UV-light environment. The data-collection platform was covered with a tarp to block any external light. Since we used inexpensive UV LEDs, there was a visible violet hue to the UV-light-lit images. However, only the invisible-ink-marked point stood out in the images.

We controlled the lighting using a programmable USB hub. This enabled us to switch between white light and UV light at high speeds programmatically for every frame captured with the camera.

Table 1. Three-axis platform specifications.

Elements	Technical information
Microprocessor	Raspberry Pi ZeroW
Language	Python
Communication	802.11 b/g/n wireless LAN
Sensors	MPU9250
Motors	RDS3115MG x3
Tilt angle limit	-15° to $+15^\circ$
Power supply	5V

3.2.4. Three-axis platform

Using only mobile robots would restrict the object's orientation to the XY plane. To overcome this, we designed and built a three-axis platform that can be placed over the mobile robots, as shown in Figure 5. The platform enables us to tilt the objects placed on it in any direction with a maximum tilt of up to 15°.

One caveat of our three-axis platform is that due to its size, it can be placed on top of the Khepera IV robot only. Thus, it restricted our ability to collect data using it for objects which are smaller than this robot. Table 1 provides the technical specifications of our three-axis platform.

3.2.5. Environmental noises

Flashlights were used to add random light noise to simulate reflections and harsh lighting conditions of a real-world environment. We used a fog generator to add smoke noise to the images. This is representative of a real-world dishwasher system, *where* the steam from the dishwasher can partially occlude the objects under the camera.

The data-collection setup was modular. The components could be replaced or new components added to vary the environmental conditions. Adding the above-mentioned noise enabled us to train an object-detection model with realistic data exceptionally robust against real-world conditions. Figure 6 shows a sample of our data with environmental noise.

3.3. Data-collection and annotation workflow

Figure 7 shows the workflow of our data collection and annotation process.

3.3.1. Pre-setup

We first painted the object's grasp position using an invisible marker. We marked the bottom center of the dish with this marker. The dish was then placed on a mobile

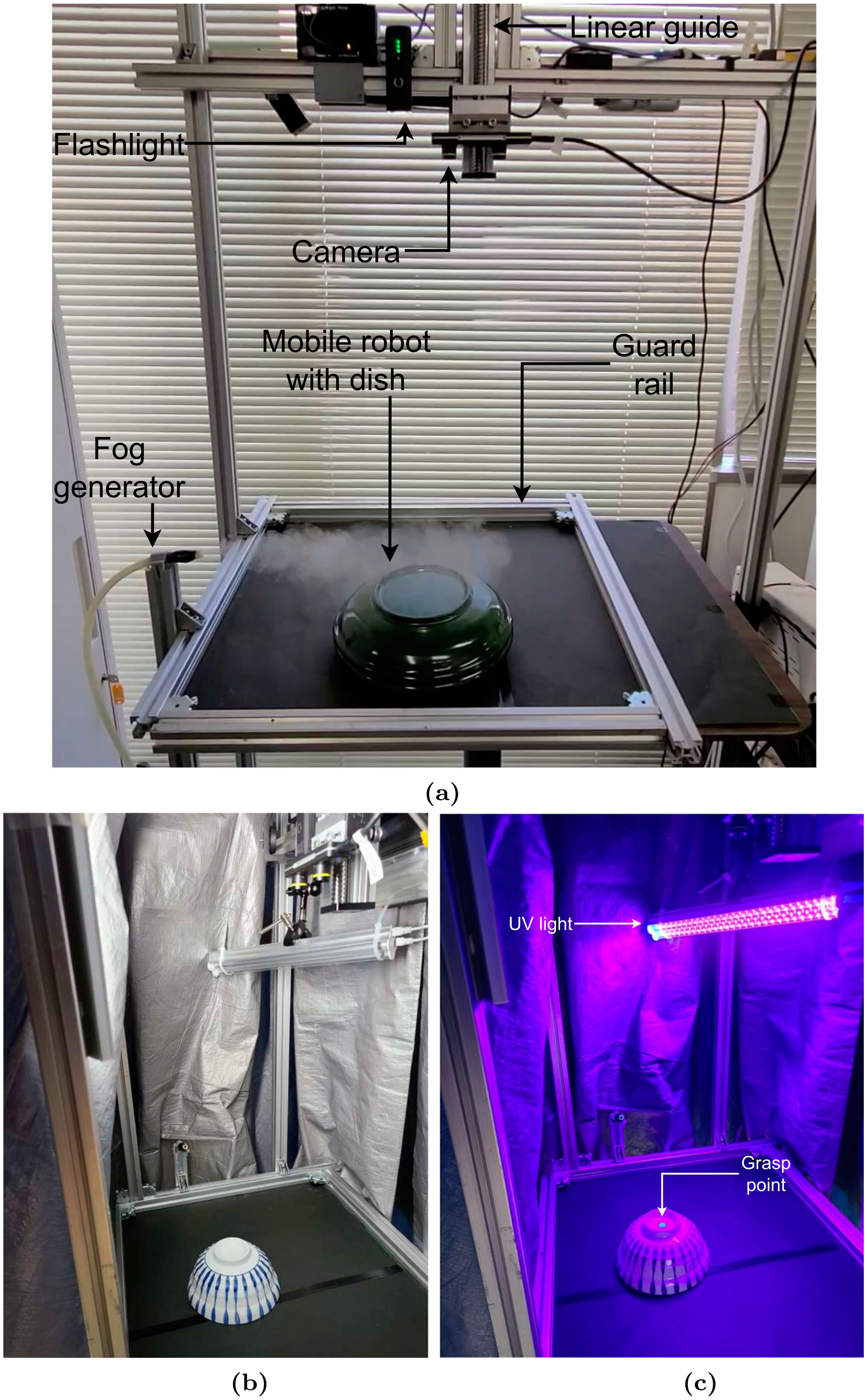

Figure 3. Automatic data collection setup: mobile robot setup with lighting environment for invisible markers. (a) Data collection setup. (b) White light environment. (c) UV light environment.

robot depending on the size of the dish. We move the robot across the platform inside the camera's field of view.

3.3.2. Image capture

At every location, we captured two images of the object, one under white light and the other under UV light. This

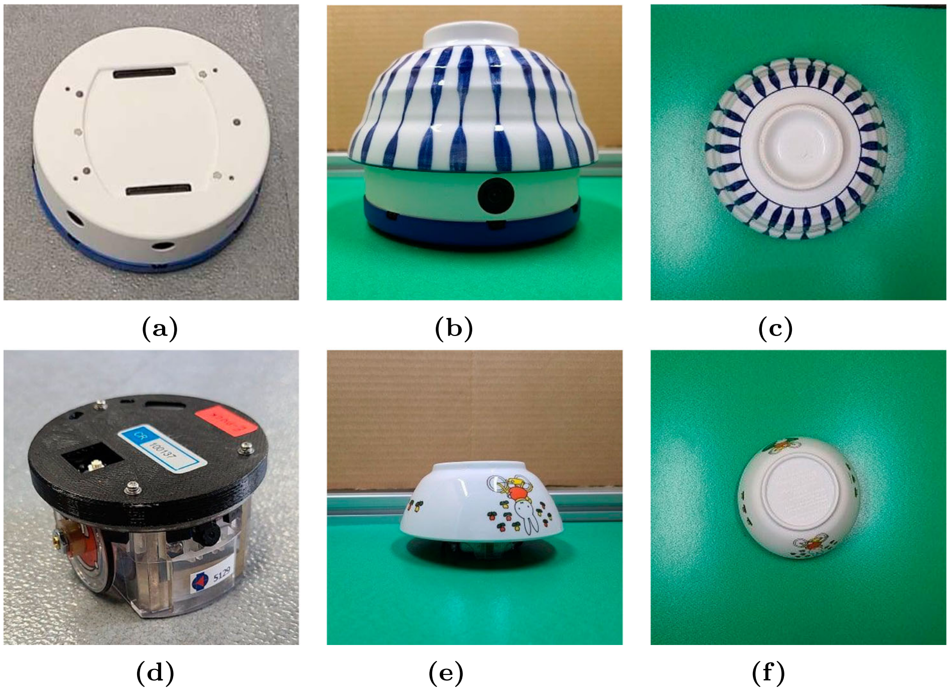

Figure 4. Mobile robots with dishes. (a) Khepera robot. (b) Khepera with dish side-view. (c) Khepera with dish top-view. (d) E-puck2 robot. (e) E-puck2 with dish side-view. (f) E-puck2 with dish top-view.

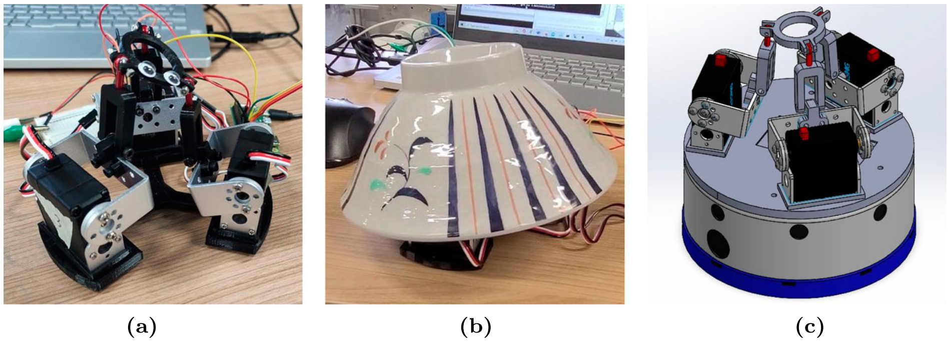

Figure 5. Three-axis platform. (a) Three-axis platform. (b) Three-axis platform with dish. (c) Khepera with a three-axis platform.

process was very rapid with the help of a high-frame-rate camera capture and lighting environment that changed alternately every frame. This enabled us to capture a total of 2300 images in around 140 s. The 2300 images consisted of 1150 pairs of white-light-lit and UV-light-lit images.

Since the speed of the mobile robots was kept low to avoid any collision or damage, and the capture rate of the camera was high, we captured many image-pairs when the object was in the same location. Thus, after deleting any duplicate image-pairs, we obtained around 300 image-pairs in 140 s. We choose to filter out the duplicate image-pairs in the post-processing step rather than delaying the image capture because carrying out post-processing is much faster, i.e. it did not increase our cycle-time for data collection, and the time required for the process was negligible.

3.3.3. Grasp-position annotation and bounding-box annotation

The white-light-lit image is the primary image without annotation. The invisible marker clearly stands out under the UV light. Its pixel coordinate is easily obtained by thresholding the RGB value of the color of the marker. This coordinate was used for the grasp-position annotation in our use case.

We then drew a square bounding box with the grasp-position coordinate as the center, by using the following equation:

$$\begin{aligned} (x_{\mathrm{TL}},\ y_{\mathrm{TL}}) &= (x_c - \mathrm{w}/2), \quad (y_c - \mathrm{h}/2) \\ (x_{\mathrm{BR}},\ y_{\mathrm{BR}}) &= (x_c + \mathrm{w}/2), \quad (y_c + \mathrm{h}/2) \end{aligned} \tag{1}$$

where $(x_{\mathrm{TL}},\ y_{\mathrm{TL}})$ and $(x_{\mathrm{BR}},\ y_{\mathrm{BR}})$ denote the top-left and bottom-right coordinates of the bounding box,

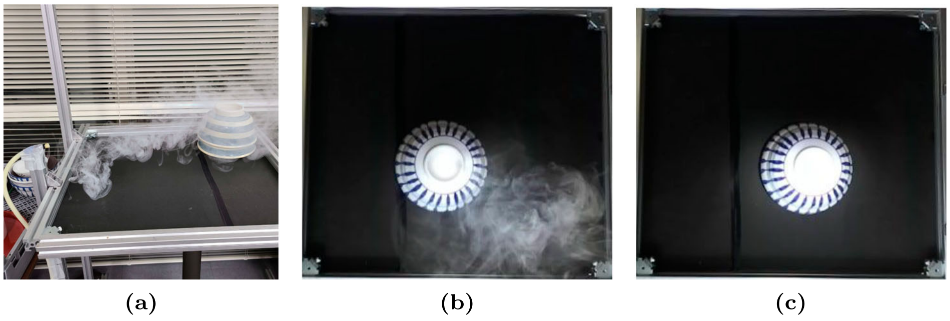

Figure 6. Environmental noise.

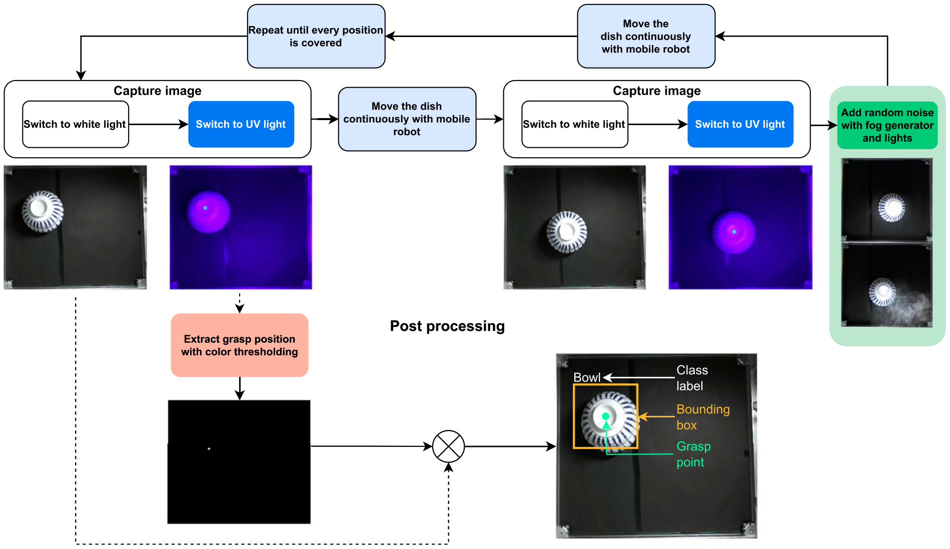
Repeat until every position is covered
Move the dish continuously with mobile robot
Capture image
Switch to white light
Switch to UV light
Move the dish continuously with mobile robot
Capture image
Switch to white light
Switch to UV light
Add random noise with fog generator and lights
Extract grasp position with color thresholding
Post processing
Bowl
Class label
Bounding box
Grasp point

Figure 7. Automatic data-collection and annotation workflow.

respectively, and (x_c, y_c) is the center of the bounding box. The *width* and *height* of the bounding box are denoted by **w** and **h** in the equation, which can be adjusted according to the size of the object. Although the bounding box drawn using the above equation is imprecise and does not fit the object tightly, it performed well to classify the dishes.

3.4. Other applications

Although we considered only symmetric bowls due to the constraints of our target application, our data-collection setup can be used to collect data for asymmetric objects as well, as shown in Figure 8.

Invisible markers can also be used to annotate key points of an object, as shown in Figure 9. These key points can be used to estimate the orientation of the dish with respect to the camera, which we did not explore in this study.

Table 2 provides a comparison of our method against the methods presented in Section 2 in terms of cost, annotation and time required for data collection. De Gregorio et al. took around 7.2 s to capture an image [17]. We estimated the time required for data collection by the other two methods [18], and [19] by simulating robot motion and the average time for the robot to calculate inverse kinematics and move to a new position for image capture. For estimating the time required for annotating objects with Rennie et al.'s method [18], we used the data provided by Su et al. [14], i.e. an average of 50 s for drawing one bounding box per image. Similarly, we extrapolated the time required for 6D annotation with De Gregorio et al.'s method [17] to estimate the time required to annotate images with Fang et al.'s method [19].

4. Experiments

To check the efficacy of our data-collection method, we conducted two comparison tests.

We first trained an object-detection model with the data collected with our method. We trained another instance of the same model with synthetically generated data. The test data in both cases were hand-annotated

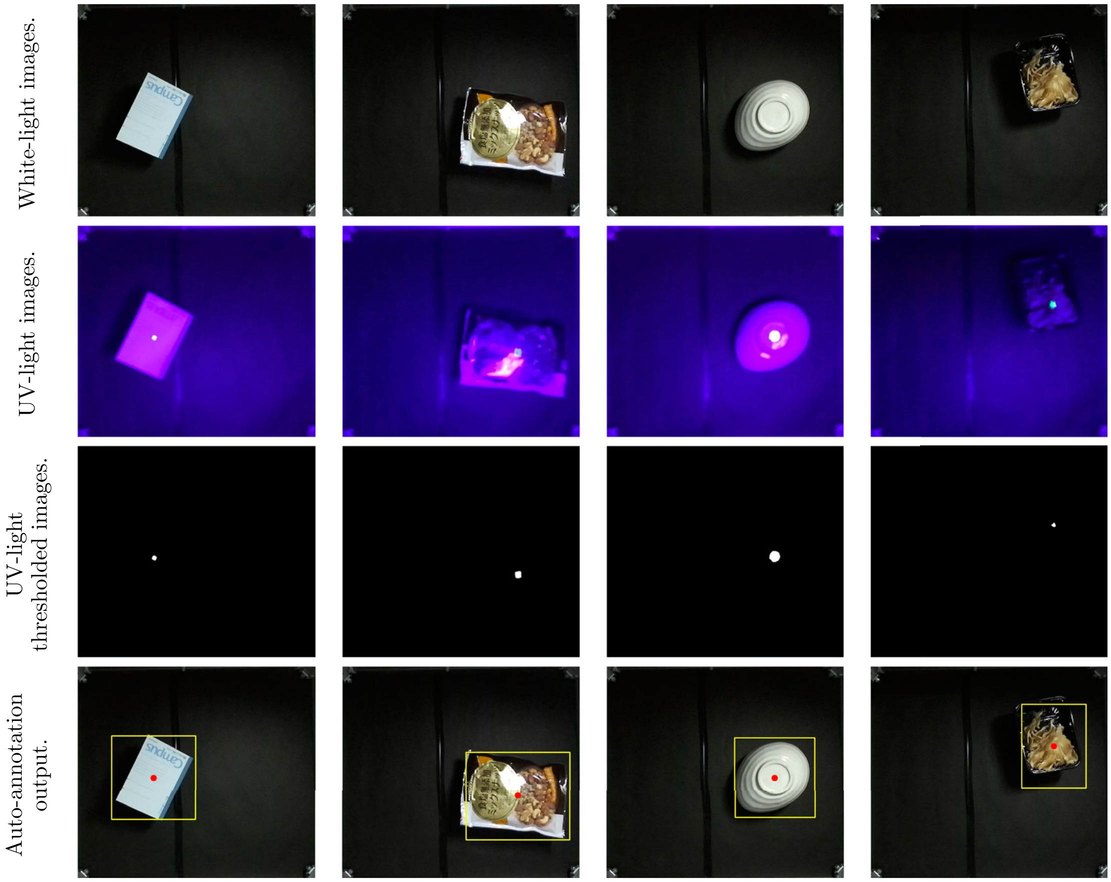

Figure 8. Data collection of objects of various shapes.

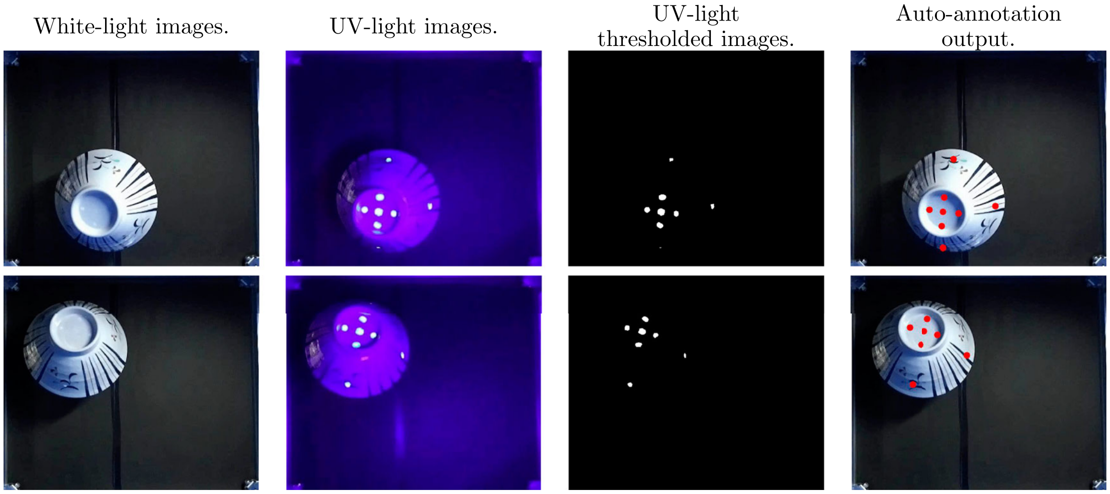

Figure 9. Key-point data collection with three-axis platform.

data. We further expanded this comparison by training an instance of the model with both the data collected with our method and synthetically generated data. We compared the mean Intersection over Union (IoU) for all dish classes between the three instances of the model.

We then compared the following three data-collection methods: (1) manual data collection, (2) synthetic data collection, and (3) automated data collection with mobile robots and invisible markers (proposed). The metrics of comparison were procedure of data collection and annotation, cost, and time required.

4.1. Object detection performance comparison

The objects for detection were dishes of various classes shown in Figure 10. We randomly selected twelve dishes

Table 2. Comparison of data-collection methods in terms of robot cost and annotation time.

	Robot	Cost (USD)	Annotation	Annotation method	Time for data collection and annotation of one image
De Gregorio et al. [17]	Universal Robots UR5	30,000	2D/6D	AR pen + Camera projection (Semi-automatic)	Data collection: 7.2 s. Annotation: ∼ 300 s for first image in sequence. Negligible time for following images.
Rennie et al. [18]	Motoman SDA10F	100,000	6D	Human annotation (Manual)	Data collection: 5–10 s. Annotation: ∼ 50 s.
Fang et al. [19]	Universal Robots UR5	30,000	6D	Human annotation + Camera projection (Semi-automatic)	Data collection: 5–10 s. Annotation: ∼ 300 s for first image in sequence. Negligible time for following images.
Mobile robot and invisible marker data collection (Proposed)	Khepera IV and E-puck2	5,000	2D	Invisible markers (Automatic)	Data collection + Annotation : 0.4 s

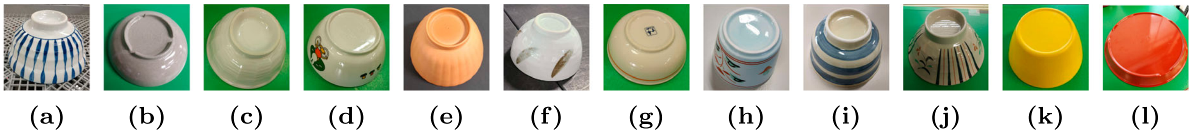

Figure 10. Dish classes.

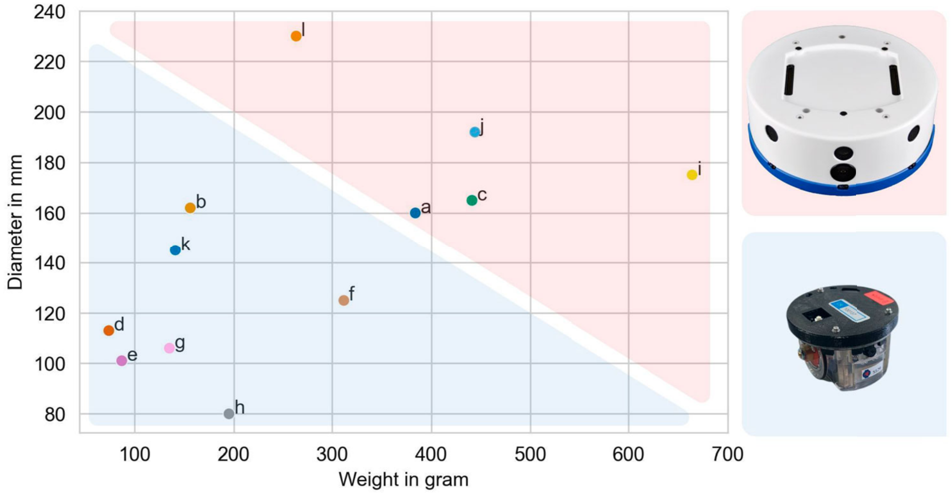

Figure 11. Mobile robots assigned to dish based on its dimensions and weight.

from the commercial restaurant dishes available to us. Figure 11 describes which robot was assigned to each dish according to its dimension and weight.

4.1.1. Data collected using mobile robots and invisible markers

For automatic data collection with our method, we used three backgrounds: green, black and dishwasher rack. Figure 12 shows a sample of data collected with our method.

We collected around 1500 images for each dish type with the images distributed over the three backgrounds mentioned above. We were able to collect more than 18,000 images. The total time required to collect and annotate the images was around 2 h, leaving out the time required to setup the objects and changing the backgrounds, which was only a few minutes.

4.1.2. Synthetically generated data

To create synthetically generated data, we first created accurate 3D models with realistic textures of our target dishes using a 3D scanner and hiring a professional digital artist. Figure 13 shows one of the 3D models.

We used Unreal Engine 4 (UE4) [34] and NVIDIA Deep learning Dataset Synthesizer (NDDS) [35] tool to

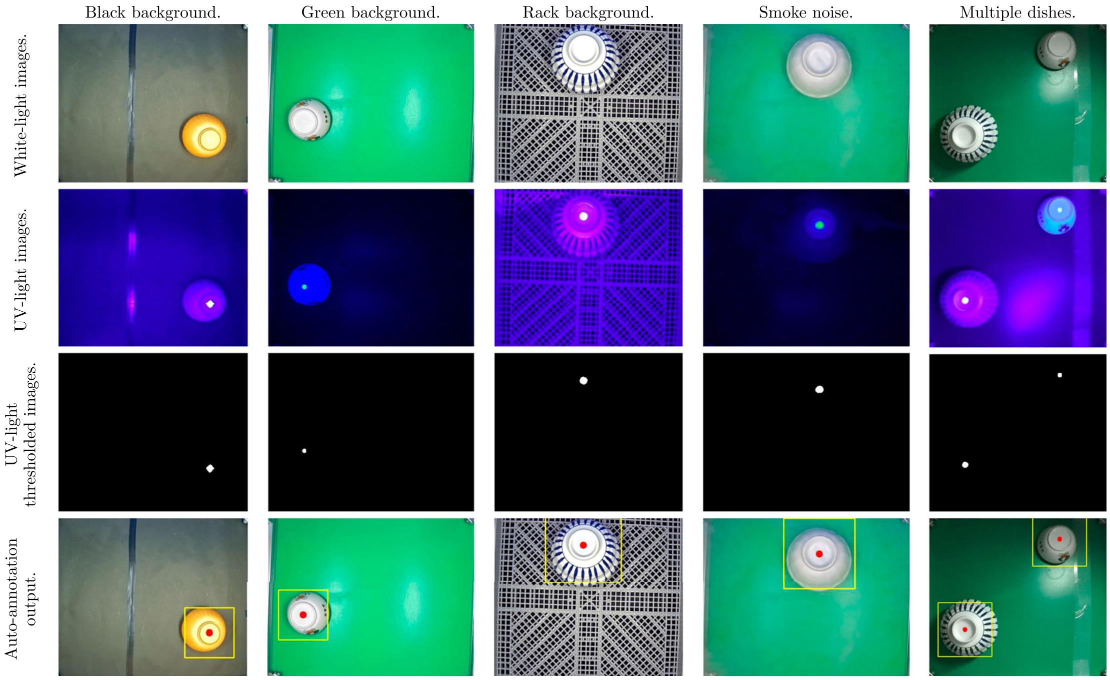

Figure 12. Data sample from the proposed method.

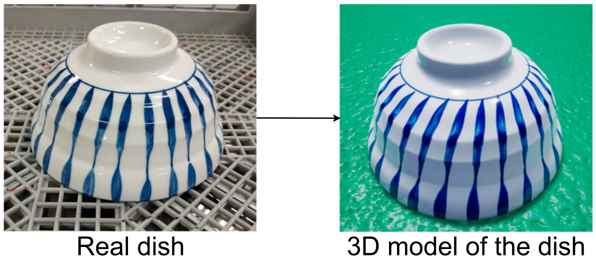

Figure 13. Real dish converted to 3D model.

generate our synthetic data. Our simulation setup in UE4 is shown in Figure 14.

The NDDS tool enabled us to move the dishes randomly in the scene. It also provided a random background generator for domain randomization that helped make the model robust against any background not seen during training. Domain randomization had previously been used to bridge the reality gap of models trained using only synthetic data [36] and increase the accuracy of object detection for indoor objects [37].

We made a few changes to the NDDS tool to restrict the rotation of our symmetric objects in each axis to be controlled by the user as rotation around the z-axis is unfavorable for symmetric objects [38].

We generated synthetic data that had a mixture of images *where* each dish is present individually and images *where* all the dishes are mixed together. The individual-dish images contained three dishes of the same type. Five thousand images for each of the 12 dishes were generated

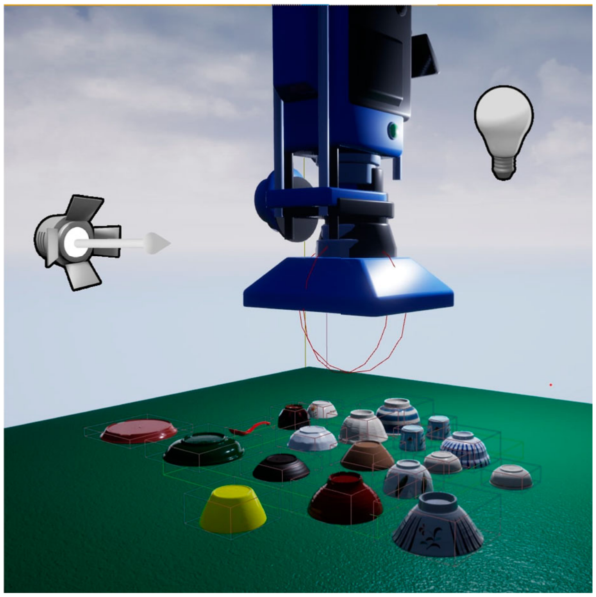

Figure 14. Unreal Engine 4 environment.

amounting to 60,000 images. A further 5000 mixed-dish images brought the total to 65,000 synthetic images.

The 5000 mixed-dish images also contained 1000 images to which we added synthetic noise such as water droplets, smoke, and dirt. Some of these images are shown in Figure 15. A sample of synthetically generated data is shown in Figure 15.

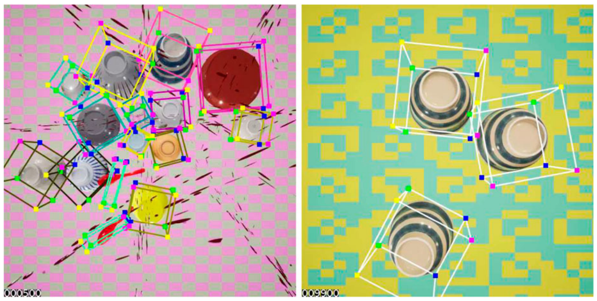

Figure 15. Synthetic data sample.

4.1.3. Test data

The test data used for all three model instances was a hand-labeled dataset of multiple dishes in various real-world backgrounds. We collected and hand-annotated around 1500 images containing different combinations of our dish classes. A sample of our test data is shown in Figure 16. The distributions of the training and test data for both models are shown in Figure 17.

4.1.4. Experiment results

We used the mean IoU metric for comparing the performances of each model instance. The model instance trained with synthetic data only achieved a mean IoU for all dishes of 73.45%. The model instance trained with the data collected with our method instead achieved a mean IoU for all dishes of 94.33%. We also ran an ablation test to see how much of an impact the extra noise had, and the mean IoU dropped to only 93.62%. This was due to a lack of a significant number of edge case images with noise in the test data.

We further combined the two datasets, synthetically generated and the data collected with our method with added noise, to train a third instance of the model. It achieved a mean IoU for all dishes of 97.21%.

Table 3 summarizes our test results and Figure 18 shows the IoU score for each dish class by each model instance. As we can see, synthetic data by itself does not make the model robust to inference on real images.

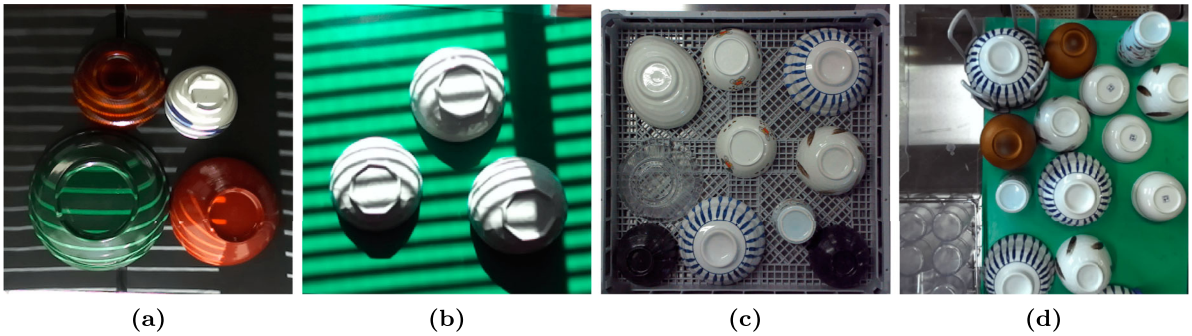

Figure 16. Test images.

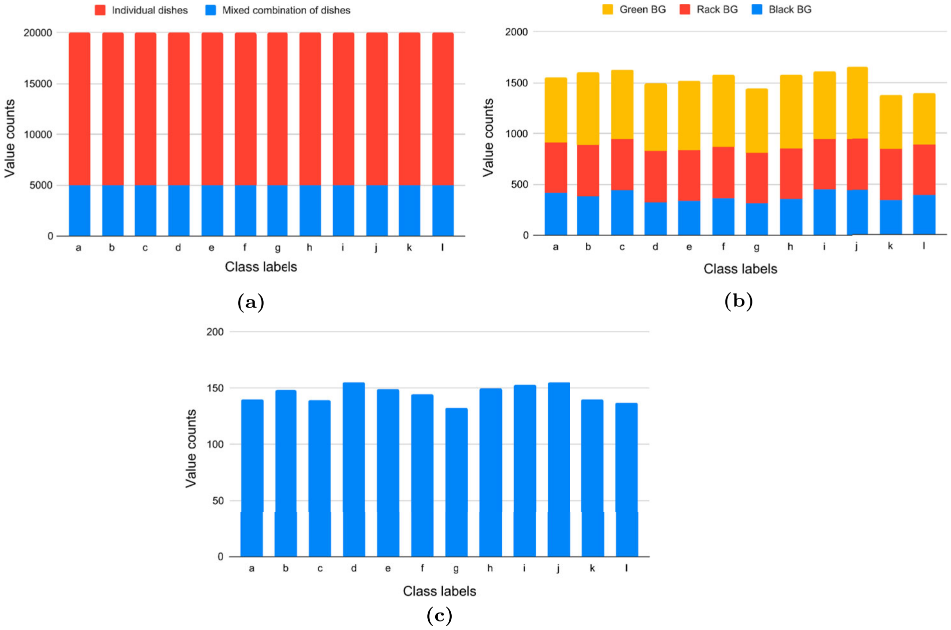

Figure 17. Distributions of training and test set data. (a) Distribution of synthetic-training data. (b) Distribution of mobile-robot and invisible-marker data. (c) Distribution of test data.

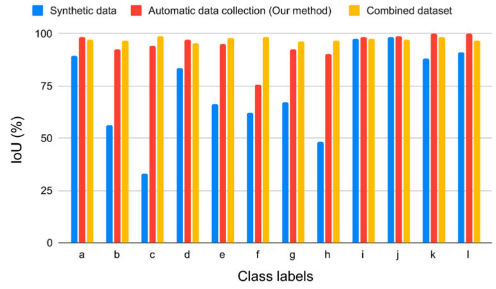

Figure 18. Comparison of IoU scores for all dishes.

Table 3. Comparison of mean IoU scores on test data.

	Mean IoU on test data (threshold > 0.5)
Synthetic data	73.45%
Automatically collected data – with noise (proposed)	94.33%
Automatically collected data – without noise	93.62%
Combined dataset	97.21%

However, when combined with data collected by our method, it helps to increase the robustness of object-detection model by contributing with data on edge cases which would have been difficult to collect manually.

4.2. Comparison among methods of data collection and annotation

4.2.1. Time taken to collect images and annotate

Manual data collection is highly laborious and time-consuming. It took us around 30 h for manual data collection and annotation of around 1500 images, i.e. around 70–80 s per image. The time required increased as the size of the dataset increased. One could reduce the time by distributing the work among more workers, but the cost would also increase.

Synthetic data generation is a much faster process compared with manual data collection. However, creating accurate 3D assets of objects is an arduous task and requires technical expertise. Setting up the simulation environment also requires domain knowledge and expertise of the tool. Once the 3D models are available and environment is setup, however, the process is very fast.

It took us less than 5 min to generate 10,000 images and a little bit more than an hour to generate the whole training dataset of 65,000 images. This equals around 0.03 s per image. The time required for data generation can be further reduced by using better computing resources. We used an NVIDIA GTX 1050 GPU and an Intel i7 processor for the synthetic-data-generation process.

Data collection using mobile robots and invisible markers is a much less laborious task compared with manual data collection. We were able to collect and annotate around 300 images in a little bit over 2 min and our whole dataset of 18,000 training images in around 2 h.

4.2.2. Cost

As manual data collection was done by in-house engineers, we estimated the monetary cost for collecting the images to be around $0.45. The cost of manual data annotation was estimated using the current fee listed in Amazon Mechanical Turk, which is around $0.80 per

Table 4. Comparison of data-collection methods.

	Manual data collection	Synthetic data collection	Automated data collection with mobile robot (Proposed)
Data collection	– Images are captured manually	– Images of objects in synthetic environment are captured automatically	– Images are captured automatically
	– Variation in object position, background, lighting, and other environment variables depend on worker's intuition and knowledge	– Object position, background, lighting, and other environment variables can be randomized programmatically	– Object position, lighting, and other environment variables can be randomized programmatically. Scene background can be changed manually or modified in post-processing
Annotation	– Objects in image and their grasp-position are annotated manually	– Objects in image and their grasp-position are automatically annotated during data generation	– Objects in image and their grasp positions are automatically annotated using corresponding invisible-marker image during operation
	– Quality and consistency of bounding-box and grasp- position annotation depends on worker's skill level	– Quality and consistency of bounding-box and grasp-position annotation is high	– Quality of grasp-position annotation is consistent, but the bounding-box annotations are not precise
Cost for data collection + annotation	– Man hour cost for data collection + annotation: 1500–2000 USD – Repeat costs for every new dataset	– Average cost of 3D model: 250 USD – Negligible cost for any data generated for same objects	– Cost for mobile robots and other equipment: $\sim$ 5000 USD – Negligible cost for any data collected for any object
Time required for data collection and annotation of one image (seconds)	$\sim$ 70–80	$\sim$ 0.03	$\sim$ 0.4

image, thus a total of around $1.4 per image. This cost could be reduced by outsourcing both the data collection and annotation tasks, but one should consider the cost of quality and training how to collect the data in that scenario.

In synthetic data generation, although the cost of generating a new dataset is negligible, we need to create an accurate 3D model of the objects beforehand. We used an external service to construct accurate 3D models of our objects, which cost us up to $250 for each object.

Data collection and annotation with our method has an upfront cost of around $5000 for the mobile robots, three-axis platform, invisible markers and other equipment. Once the setup is complete, the cost of collecting data and annotating an image is very small. Table 4 provides a comparison among the various data-collection methods used in our experiments.

5. Conclusions

We proposed a method for data collection and annotation using mobile robots equipped with a tilt platform and invisible markers. Such mobile robots enable us to capture images of objects at every possible location inside the camera's field of view. A high-frame-rate image capture combined with switching between white light and UV light enables us to obtain an image pair of white-light-lit and UV-light-lit images. The invisible marker in the UV-light-lit image helps obtain annotation data during data collection.

The data collected with our method are much more useful in making deep learning models robust against real-world conditions compared with synthetically generated data. We conclude that our proposed method can create large datasets for deep learning models that are consistent, realistic, less time-consuming and inexpensive. Other than object detection, our method can also be used to collect data for grasp-position-estimation and key-point-estimation tasks.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by Connected Robotics Inc.

Notes on contributors

Suraj Prakash Pattar received his Bachelor's degree in Mechatronics Engineering from Visvesvaraya Technological University, India in 2014, Master's in Robotics Engineering from the University of Genoa, Italy in 2017, and Master's in Advanced Robotics from Ecole Centrale de Nantes, France in 2018. He has been a Ph.D. student at Chubu University since 2020 and has worked as a Robotics AI Research Engineer at Connected Robotics Inc., Tokyo, since 2019. His research interests include robotics, computer vision and machine learning.

Thomas Killus received his Master's degree in mechanical engineering specializing in mechatronics from Technical University Brunswick, Germany, in 2020. Since 2021, he has worked at Connected Robotics, focusing on hardware development and robot motion control. His research interests include reinforcement learning and robot motion control.

Tsubasa Hirakawa received his Ph.D. degree in Computer Science from Hiroshima University, Japan in 2017. From 2017 to 2019, he was a research fellow at Chubu University. He was a designated associate professor at the Chubu Institute for Advanced Studies, Chubu University, Japan from 2019 to 2021. He has been a lecturer at Chubu University, Japan, since 2021. He was a Fellow of the Japan Society for the Promotion of Science from 2014 to 2017 and a visiting researcher at ESIEE Paris, France, in 2014 and 2015.

Takayoshi Yamashita received his Ph.D. in Computer Science, Chubu University, Japan in 2011. He worked at OMRON Corporation from 2002 to 2014 black and has been a lecturer at the Department of Computer Science, Chubu University, Japan since 2014. His current research interests include object detection, object tracking, human activity understanding, pattern recognition and machine learning. He is a member of the IEEE, the IEICE and the IPSJ.

Tetsuya Sawanobori received his Bachelor's degree in Engineering from the University of Tokyo and his Master's in Engineering from Kyoto University. He worked for one of Japan's major restaurant companies and then joined Soft Servo System, Inc., which is a motion control development company originating from MIT. He later started his own company, Connected Robotics Inc., which is involved in food industry automation with robotics, especially using robotic arms on unique robot control technology combining machine learning technology.

Hironobu Fujiyoshi received his Ph.D. in Electrical Engineering from Chubu University, Japan, in 1997. From 1997 to 2000, he was a post-doctoral fellow at the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA, USA, *where* he worked on the DARPA Video Surveillance and Monitoring (VSAM) effort and the humanoid vision project for the HONDA Humanoid Robot. He is currently a professor at the Department of Robotics, Chubu University, Japan. From 2005 to 2006, he was a visiting researcher at the Robotics Institute, Carnegie Mellon University. His research interests include computer vision, video understanding, and pattern recognition. He is a member of the IEEE, the IEICE, and the IPSJ.

ORCID

Suraj Prakash Pattar http://orcid.org/0000-0002-7480-4074
Tsubasa Hirakawa http://orcid.org/0000-0003-3851-5221
Takayoshi Yamashita http://orcid.org/0000-0003-2631-9856
Hironobu Fujiyoshi http://orcid.org/0000-0001-7391-4725

References

[1] He K, Gkioxari G, Dollár P, et al. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). IEEE; 2017. p. 2961–2969.

[2] Farhadi A, Redmon J. YOLOv3: an incremental improvement. arXiv preprint.arXiv; 2018.

[3] Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. In: European Conference on Computer Vision. Springer; 2016. p. 21–37.

[4] Yu J, Weng K, Liang G, et al. A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation. In: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE; 2013. p. 1175–1180.

[5] Kumra S, Kanan C. Robotic grasp detection using deep convolutional neural networks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2017. p. 769–776.

[6] Caldera S, Rassau A, Chai D. Review of deep learning methods in robotic grasp detection. Multimodal Technol Interact. 2018;2(3):57.

[7] Zeng A, Yu K-T, Song S, et al. Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2017. p. 1386–1383.

[8] Xiang Y, Schmidt T, Narayanan V, et al. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. Proceedings of Robotics: Science and Systems. 2018. DOI:10.15607/RSS.2018.XIV.019.

[9] Kehl W, Manhardt F, Tombari F, et al. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. Venice, Italy. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE; 2017. p. 1530–1538. DOI:10.1109/ICCV.2017.169.

[10] Zhao Z-Q, Zheng P, Xu S-T, et al. Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst. 2019;30(11):3212–3232.

[11] Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009. p. 248–255.

[12] Oh S, Hoogs A, Perera A, et al. A large-scale benchmark dataset for event recognition in surveillance video. In: CVPR 2011. IEEE; 2011. p. 3153–3160.

[13] Sorokin A, Forsyth D. Utility data annotation with Amazon mechanical turk. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE; 2008. p. 1–8.

[14] Su H, Deng J, Fei-Fei L. Crowdsourcing annotations for visual object detection. Toronto, Ontario, Canada. In: Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence. AAAI; 2012.

[15] Yao A, Gall J, Leistner C, et al. Interactive object detection. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2012. p. 3242–3249.

[16] Connected Robotics Inc. Dishwasher robot. [Online]. Available from: https://connected-robotics.com/products/dishwashing-bowl/.

[17] De Gregorio D, Tonioni A, Palli G, et al. Semiautomatic labeling for deep learning in robotics. IEEE Trans Autom Sci Eng. 2019;17(2):611–620.

[18] Rennie C, Shome R, Bekris KE, et al. A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place. IEEE Robot Autom Lett. 2016;1(2):1179–1185.

[19] Fang H-S, Wang C, Gou M, et al. GraspNet-1Billion: a large-scale benchmark for general object grasping. Seattle, WA, USA. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2018. p. 11441–11450. DOI:10.1109/CVPR42600.2020.01146.

[20] Mitash C, Bekris KE, Boularias A. A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2017. p. 545–551.

[21] Tremblay J, To T, Sundaralingam B, et al. Deep object pose estimation for semantic robotic grasping of household objects. In: Conference on Robot Learning (CoRL); 2018. [Online]. Available from: https://arxiv.org/abs/1809.10790.

[22] Planche B, Wu Z, Ma K, et al. Depthsynth: real-time realistic synthetic data generation from cad models for 2.5D recognition. In: 2017 International Conference on 3D Vision (3DV). IEEE; 2017. p. 1–10.

[23] Hodaň T, Vineet V, Gal R, et al. Photorealistic image synthesis for object instance detection. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE; 2019. p. 66–70.

[24] Sela M, Xu P, He J, et al. Gazegan – unpaired adversarial image generation for gaze estimation. CoRR; 2017. [Online]. Available from: http://arxiv.org/abs/1711.09767.

[25] Borrego J, Dehban A, Figueiredo R, et al. Applying domain randomization to synthetic data for object category detection. CoRR; 2018. [Online]. Available from: http://arxiv.org/abs/1807.09834.

[26] Adhikari B, Peltomaki J, Puura J, et al. Faster bounding box annotation for object detection in indoor scenes. In: 2018 7th European Workshop on Visual Information Processing (EUVIP). IEEE; 2018. p. 1–6.

[27] Adhikari B, Huttunen H. Iterative bounding box annotation for object detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE; 2021. p. 4040–4046.

[28] Chen Y, Liu L, Tao J, et al. The image annotation algorithm using convolutional features from intermediate layer of deep learning. Multimed Tools Appl. 2021;80(3):4237–4261.

[29] Depierre A, Dellandréa E, Chen L. Jacquard: A large scale dataset for robotic grasp detection. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2018. p. 3511–3516.

[30] Yan X, Hsu J, Khansari M, et al. Learning 6-DOF grasping interaction via deep geometry-aware 3D representations. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2018. p. 3766–3773.

[31] Cao H, Fang H-S, Liu W, et al. SuctionNet-1Billion: a large-scale benchmark for suction grasping. IEEE Robot Autom Lett. 2021;6(4):8718–8725.

[32] Baker S, Scharstein D, Lewis J, et al. A database and evaluation methodology for optical flow. Int J Comput Vis. 2011;92(1):1–31.

[33] Takahashi K, Yonekura K. Invisible marker: automatic annotation of segmentation masks for object manipulation. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2020. p. 8431–8438.

[34] Epic Games. Unreal engine. [Online]. Available from: https://www.unrealengine.com.

[35] To T, Tremblay J, McKay D, et al. NDDS: NVIDIA deep learning dataset synthesizer. 2018. Available from: https://github.com/NVIDIA/Dataset_Synthesizer.

[36] Tremblay J, Prakash A, Acuna D, et al. Training deep networks with synthetic data: bridging the reality gap by domain randomization. Salt Lake City, UT, USA. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE; 2018. p. 969–977. DOI:10.1109/CVPRW.2018.00143.

[37] Dehban A, Borrego J, Figueiredo R, et al. The impact of domain randomization on object detection: a case study on parametric shapes and synthetic textures. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2019. p. 2593–2600.

[38] Rad M, Lepetit V. BB8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. Venice, Italy. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE; 2017. p. 3848–3856. DOI:10.1109/ICCV.2017.413.