

Received January 12, 2021, accepted January 20, 2021, date of publication January 25, 2021, date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054493

# Object Pose Estimation Incorporating Projection Loss and Discriminative Refinement

JUN-KAI YOU, CHEN-CHIEN JAMES HSU<sup>1</sup>, (Senior Member, IEEE),  
WEI-YEN WANG<sup>1</sup>, (Fellow, IEEE), AND SHAO-KANG HUANG

Department of Electrical Engineering, National Taiwan Normal University, Taipei 106, Taiwan

Corresponding author: Chen-Chien James Hsu (jhsu@ntnu.edu.tw)

This work was supported in part by the Chinese Language and Technology Center of the National Taiwan Normal University (NTNU) through the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan, and in part by the Ministry of Science and Technology, Taiwan, through the Pervasive Artificial Intelligence Research (PAIR) Labs, under Grant MOST 109-2634-F-003-006 and Grant MOST 109-2634-F-003-007.

**ABSTRACT** The accurate estimation of three-dimensional (3D) object pose is important in a wide range of applications, such as robotics and augmented reality. The key to estimate object poses is matching feature points in the captured image with predefined ones of the 3D model of the object. Existing learning-based pose estimation systems utilize a voting strategy to estimate the feature points in a vector space for improving the accuracy of the estimated pose. However, the loss function of such approaches only takes account of the direction of the vector, resulting in an error-prone localization of feature points. Therefore, this paper considers a projection loss function dealing with the error of the vector field and incorporates a refinement network to revise the predicted pose to obtain a good final output. Experimental results show that the proposed methods outperform the state-of-the-art methods in ADD(-S) metric on the LINEMOD and Occlusion LINEMOD datasets. Moreover, the proposed method can be applied to real-world practical scenarios in real time to simultaneously estimate the poses of multiple objects.

**INDEX TERMS** Object pose estimation, LINEMOD, occlusion LINEMOD, deep learning, convolutional neural network.

## I. INTRODUCTION

The main purpose of object pose estimation is to describe the relationship between the object and world coordinates. Specifically, its goal is to obtain the rotation angle and translation distance of the object according to the captured image. The accurate estimation of three-dimensional (3D) object pose is vital in a wide range of applications, such as pick-and-placing robotic applications and virtual reality, thus attracting considerable attention in the field of computer vision.

Traditional object pose estimation methods can be roughly categorized into two-dimensional (2D) [1]–[3] and 3D [4]–[6] methods. The former detects the feature points from 2D images and solves the rotation matrix and translation matrix using Perspective- $n$ -Point ( $PnP$ ) algorithms. The latter matches the point clouds between a predefined template model and depth image information using ICP algorithms [7]. Due to the rapid growth of the graphics

processing unit (GPU), learning-based techniques have significantly improved the overall performance of object pose estimation in recent years. These approaches [8]–[24] can be classified into two different types based on the input data: RGB-D-based and RGB-based methods. RGB-D-based methods are useful in predicting the translation distance with available depth information. However, low-quality depth data are likely to incur negative impacts on pose estimation. Conversely, RGB-based methods estimate object pose by matching the feature points in the captured image against predefined ones of the corresponding 3D model of the object. Hence, it is important to determine 2D feature points in images that are related to the designated 3D model points for RGB-based methods. Such process is termed in the literature as the localization of 2D–3D correspondence. To provide a 2D–3D correspondence for the pose estimator, BB8 [11] aimed at localizing feature points projected from the eight corners of a 3D bounding box by restricting the range of pose data for training. Some approaches [19] localized feature points projected from the designated vertices of an object point cloud model. However, such localization approaches

The associate editor coordinating the review of this manuscript and approving it for publication was Charith Abhayaratne<sup>1</sup>.

are likely to encounter a mismatch problem when dealing with occluded objects. If the feature points are not sufficiently accurate, then the  $PnP$  solver will lead to an inaccurate estimate of the object pose. In order to solve this problem, one of the typical ways is to utilize voting strategies to determine the feature points in a vector space. If the selected feature points are determined from the highest hypothesis in the vector space, then the localization process can overcome the occlusion scenarios. However, the loss function of such approaches only takes into account the direction of a vector, resulting in an error-prone localization of feature points.

Thus, this paper considers a projection loss function to deal with the errors in a vector field and incorporates a pose refinement network to revise the predicted pose so as to obtain a good performance. Our proposed pose estimation systems make the following main contributions: 1) A projection loss function is designed to deal with the error of the feature point localization in a vector space, providing accurate feature points for the  $PnP$  solver. 2) The proposed system integrates a refinement network controlled by a novel discriminative strategy for improving the accuracy of pose estimation. 3) Our proposed pose estimation systems have better performance in terms of the average distance of model points (ADD) metric on both the LINEMOD [4] and Occlusion LINEMOD [6] datasets, outperforming state-of-the-art methods. 4) The proposed method can be applied to real-world practical scenarios in real time to simultaneously estimate the poses of multiple objects.

The rest of the paper is organized as follows: Section II introduces related works, Section III presents the proposed pose estimation method, Section IV shows the experimental results, and Section V concludes our work.

## II. RELATED WORKS

### A. TRADITIONAL METHODS

Traditional object pose estimation depends on matching feature points in an image against those of a 3D object model. Popular methods that extract feature points from RGB images, such as SIFT [25], FAST [26], SURF [27], and ORB [28], have been designed to handle the feature-extracting task under various environmental factors such as different camera viewpoints, lighting conditions, and noisy images. After the feature extraction, object pose can then be estimated using  $PnP$  algorithms. However, traditional methods may encounter various difficulties. First, the problem of occlusion scenarios is not well addressed because features points cannot be successfully detected. Next, the feature points of texture-less objects cannot be easily localized. Third, these methods are ad-hoc-designed to serve certain scenarios that cannot bring out a general model for various environments. Accordingly, in recent years, deep learning techniques have been introduced to solve these problems.

### B. DEEP LEARNING-BASED METHODS WITH RGB-D DATA

Depth information provides an effective solution to deal with texture-less objects. With a deep learning model, the relationship between 2D and 3D are learned using the available depth

map or point cloud. Among them, PoseCNN [20] utilized depth information to improve the accuracy of the predicted pose with ICP. DenseFusion [8], [24] integrated a pose refinement network into the pose estimation system, where the pose is estimated with fused RGB-D information and then refined by an iterative refinement network for a better performance. However, utilizing the depth information to estimate object poses faces different challenges. First, the precision of the depth map depends on the quality of the camera. If the precision of the captured depth is not sufficiently high, then the accuracy of the pose estimation will not be appealing. Next, RGB-D-based methods encounter prediction difficulties when the target object is occluded. This is because the available depth information cannot be completely matched with the entire 3D model of the object. Some of the depth data may also make the pose estimator confused with other objects, resulting in difficulties to obtain an accurate pose estimate.

### C. DEEP LEARNING-BASED METHODS WITH RGB DATA

The task of RGB-based pose estimation can generally be divided into two parts: object detection and pose estimation. The former locates the position of target object in images and collects the required features from the object, whereas the latter matches the features against those of the 3D object model to estimate poses. To make an accurate pose estimation, the most important task is to extract adequate feature points, which best describe the 2D–3D correspondence for the pose estimator. Various methods have been introduced in the literature to extract features points. SSD-6D [12] and BB8 [11] focused on extracting the projection position of the corner vertices of an object's bounding box. SSD-6D [12] classified objects from different viewpoints and directly regressed the 3D bounding box. Thus, it is unreliable when facing the occlusion problem. BB8 [11] first estimated the 2D segmentation mask, then predicted the bounding box via the mask, and finally estimated the pose by the  $PnP$  algorithm. PvNet [9] collected the feature points on the object's surface. Pix2Pose [23] obtained the feature points by directly regressing the 3D coordinates of a 3D object model. However, directly regressing the feature points from the image will encounter estimation problems in occlusion scenarios, which makes the estimation unreliable. To deal with this problem, PoseCNN [20] firstly proposed a voting strategy to locate the center point of the object in the vector space to strength the 2D–3D correspondence when occlusion occurs. Subsequently, PvNet [9] also adopted a voting strategy to select feature points from the vector field. This strategy provides better robustness for pose estimation than the regression approaches, because it allows the feature points to be selected from the occluded area of the image. However, such a strategy requires an accurate vector field estimation. Errors in the vector field will cause a large deviation between the selected feature points and the ground truth. Such an impact will definitely degrade the performance of the pose estimation results.

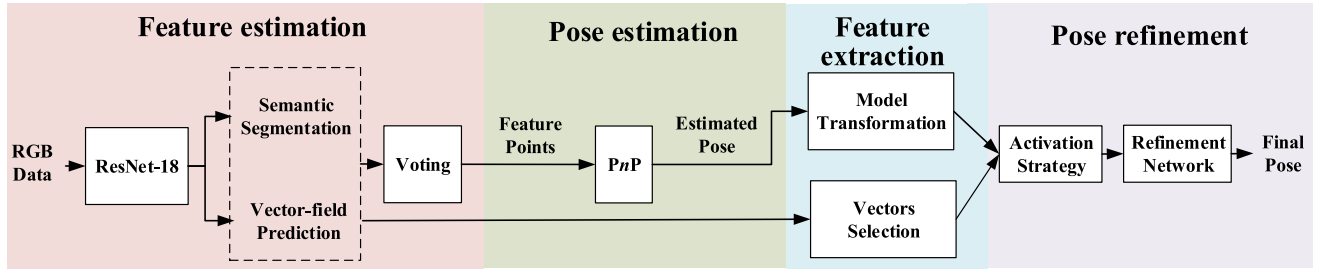


FIGURE 1. Architecture of the proposed pose estimation system.

### III. PROPOSED METHOD

In this paper, we proposed a pose estimation system built on top of PvNet [9], taking into consideration of a projection loss and a discriminative refinement network to obtain a good performance. Figure 1 shows the architecture of the pose estimation system that contains 4 stages, including a feature estimation, pose estimation, feature extraction, and pose refinement stage. First, the captured RGB image is fed into ResNet-18 for extracting semantic segmentation and vector-field data. The projection loss is integrated to ResNet-18 to increase the accuracy of vector-field data. Then, a voting strategy [9] is applied to determine the feature points. Next, a PnP solver makes an initial pose estimate according to the feature points. Third, the estimated pose, available object 3D model, and vector-field prediction are utilized to generate the input data for the refinement network at the second feature extraction stage. Specifically, the pose data and 3D model are utilized to generate a transformed 3D model of the object, and the corresponding vectors of the object are selected based on the vector-field prediction and the semantic segmentation. Lastly, an activation strategy determines whether or not to launch the pose refinement network. In the following subsections, we will provide more details on the design of the projection loss, total loss function with a dynamic weight, pose refinement process, and activation strategy.

#### A. PROJECTION LOSS FUNCTION

One of the objectives of this work is to provide accurate feature points for the PnP solver to estimate object poses. When a  $640 \times 480$  image is fed into ResNet-18, we obtain a semantic segmentation for the object and a vector-field prediction. Here the key to improve the localization results of feature points relies upon the design of the loss function of the vector field. However, the original loss function of the vector field in PvNet only considers the similarity between the predicted and ground-truth vectors. The loss gradient of this design seriously degrades when the epoch number is sufficiently large. To improve the learning effectiveness in such a condition, we add an extra loss function by considering the distance error  $d$  between the position of the predicted vector  $\vec{p}_{ik}$  and the ground-truth vector  $\vec{g}_{ik}$ , as illustrated in Figure 2, where we intend to minimize the distance error for increasing the gradient of loss at the final training stage. Therefore,

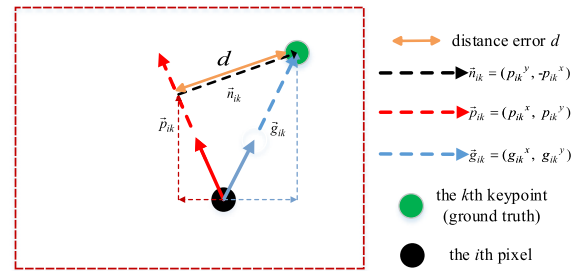


FIGURE 2. Illustration of the distance error between the predicted vector and ground truth. By minimizing the distance error, the predicted vector can get closer to the ground truth.

we propose a loss function to reduce the distance error as follows:

$$L_{proj} = \sum_{k \in K} \sum_{i \in P} l_1 \left( w_i \frac{(\vec{g}_{ik} \cdot \vec{n}_{ik})}{\|\vec{n}_{ik}\|} \right), \quad (1)$$

where  $k \in K$  is the  $k$ th keypoint and  $i \in P$  represents the  $i$ th pixel in the image  $P$ .  $\vec{p}_{ik} = (p_{ik}^x, p_{ik}^y)$  is the predicted vector from the network,  $\vec{g}_{ik} = (g_{ik}^x, g_{ik}^y)$  is the ground truth vector, and  $\vec{n}_{ik} = (p_{ik}^y, -p_{ik}^x)$  is the normal vector of the predicted vector  $\vec{p}_{ik}$ , where  $x$  and  $y$  represent the horizontal and vertical coordinates, respectively.  $w_i$  represents the value of pixel  $i$  in the predicted mask to limit the vector to the region of interest, and  $l_1(\cdot)$  is the standard Pytorch smooth\_l1\_loss function. We calculate the distance error  $d$  by projecting the target vector onto the normal of predicted vector. With this loss function, we can avoid the training process from gradient vanishing condition after long learning epochs.

#### B. TOTAL LOSS FUNCTION WITH A DYNAMIC WEIGHT

By including the projection loss, the total loss function of ResNet-18 in the proposed system comprises three losses to determine the segmentation mask and vector-field prediction. To ensure that the three losses properly work together, we design a dynamic weight to balance the scale of the loss functions. According to the experimental results, the projection loss should be launched at longer learning epochs rather than the beginning of the training process. This is because the projection loss will be too large at the beginning of the training process and hence dominates the loss function. Therefore, we design a dynamic weight propor-

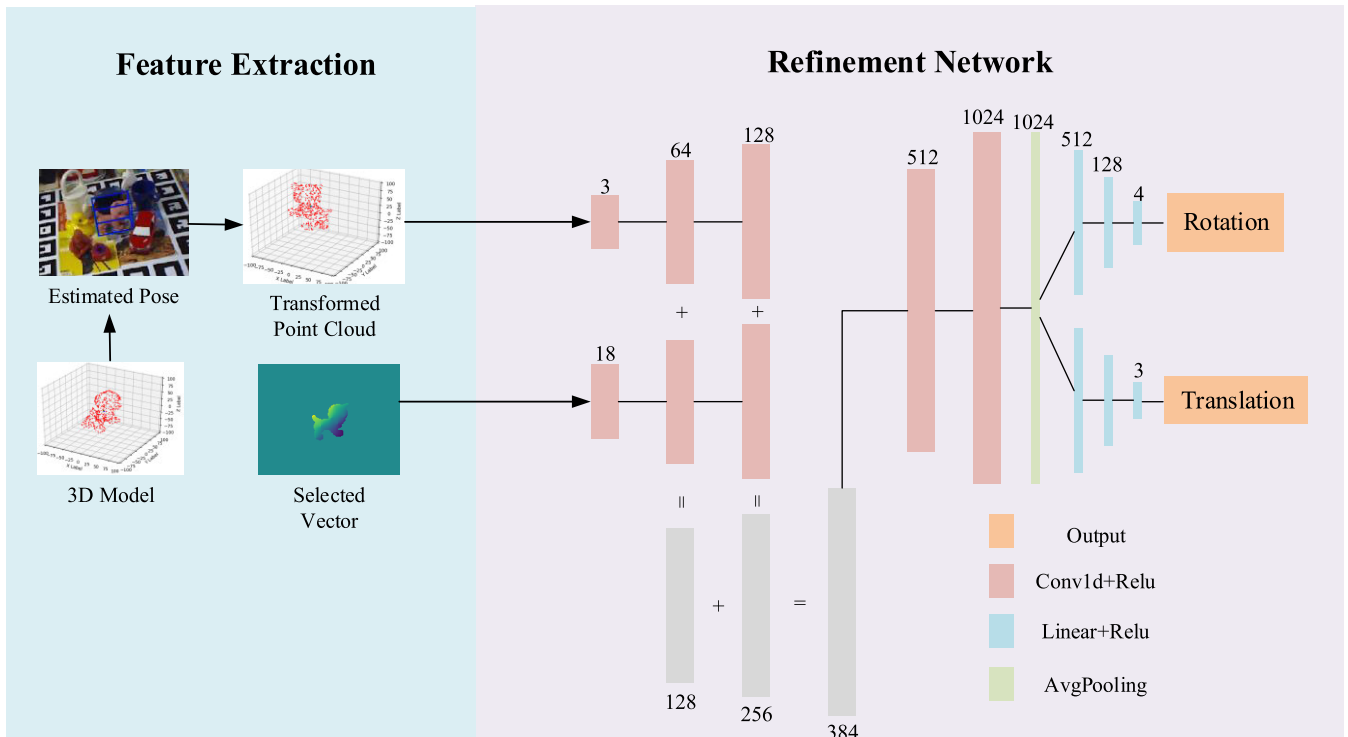


FIGURE 3. Structure of the refinement network with a feature extraction to explain the input data to the refinement network.

tional to the number of learning epoch for the total loss function:

$$L_{total} = L_{vf} + L_{seg} + \beta L_{proj}, \quad (2)$$

where  $L_{vf}$  and  $L_{seg}$  are the loss of vector-field prediction and semantic segmentation, respectively [9], and  $\beta$  is a weight of the projection loss, which we empirically determine. In the beginning, the weight starts from 0.01 to 0.05 with an increase of 0.001 every five epochs. With this strategy, the projection loss will dynamically increase during the training process, allowing the three losses working in an appropriate scale.

### C. POSE REFINEMENT

Motivated by DenseFusion [8], [24], 3D information seems to bring a strong 2D–3D correspondence for designing a learning-based pose estimator. Therefore, we integrate a pose refinement network into the pose estimation system. Because DenseFusion is an RGB-D-based approach, we only adopt the pose refinement network that does not require depth information.

Here, our rationale is to make the refinement network learn the pose refinement process rather than estimate a new pose. Figure 3 illustrates the structure of the proposed refinement network, where the input to the refinement network requires the 2D and 3D information of the object. However, the proposed RGB-based method can only provide 2D information from the previous pose estimation network. Thus, we have

to utilize the 3D features of the object 3D model. Particularly, a transformed point cloud can be obtained according to the estimated pose from the previous network, which is subsequently utilized as the 3D information for pose refinement. Furthermore, the selected vector of the object generated from the previous network is taken as the 2D information. After executing the refinement network, the predicted pose is improved and becomes closer to the ground-truth pose.

### D. REFINEMENT ACTIVATION STRATEGY

Although the refinement network can help improve the pose estimation performance, it cannot effectively work when the object was seriously occluded. In this circumstance, the refinement network cannot correctly perform. To solve this problem, we control the launch time of the refinement network through a discriminative strategy according to the area of the occlusion region, as illustrated in Figure 4. The process of the proposed activation strategy for the refinement network is shown in Figure 5. First, we project the point cloud of the object based on the predicted pose data. Then, we compare the projection with the mask generated from the semantic segmentation. If the overlapped region is greater than a pre-defined threshold, then the estimated pose is sufficiently correct to launch the refinement network. Figure 6 shows an example of the projection result of the object ‘Cat’ from the LINEMOD dataset, where the purple region is the mask from the semantic segmentation, the yellow points are the projected points of the object point cloud

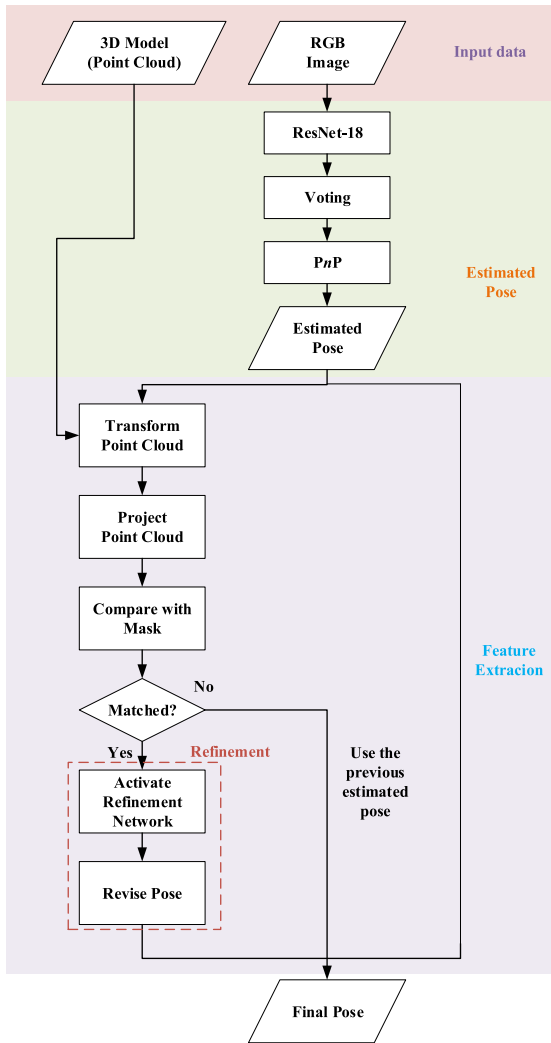


FIGURE 4. Overall process of the proposed pose estimation system.

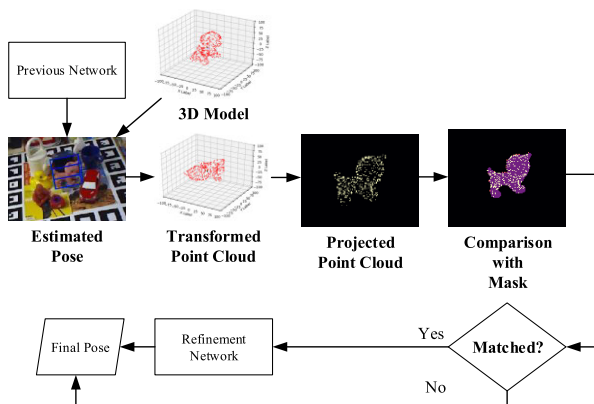


FIGURE 5. Activation strategy for the pose refinement network.

that overlap with the mask, and the red points are the ones that do not overlap with the mask. In this figure, most of the projected points are yellow, which means that the percentage of the overlapped region between the projection and the mask

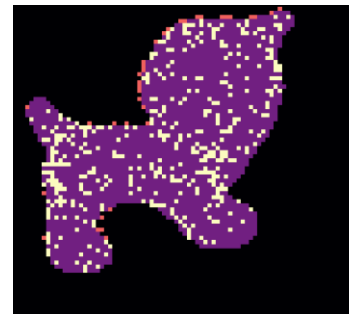


FIGURE 6. Projection of the object 'Cat' from the LINEMOD dataset in comparison with the mask according to Figure 5.

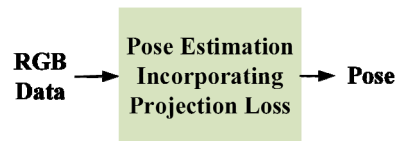


FIGURE 7. Object pose estimation incorporating the projection loss (OPEPL).

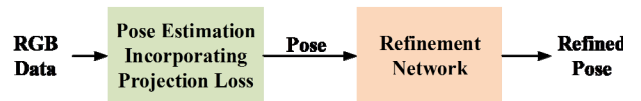


FIGURE 8. Object pose estimation incorporating the projection loss with refinement (OPEPL-R).

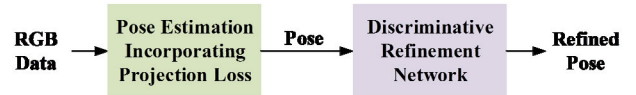


FIGURE 9. Object pose estimation incorporating the projection loss and discriminative refinement (OPEPL-DR).

is high. Hence, we will launch the refinement network in such case.

## IV. EXPERIMENTAL RESULTS

### A. EXPERIMENTAL SETUP

In this paper, we propose three pose estimation systems in different settings by considering the projection loss function, refinement network, and discriminative strategy. For easy understanding, we give each of the methods an abbreviation: 1) the base model with the projection loss function as the Object Pose Estimation Incorporating Projection Loss (OPEPL). 2) the integration of the refinement network with OPEPL as the Object Pose Estimation Incorporating Projection Loss with Refinement (OPEPL-R). 3) the refinement network controlled by a proposed discriminative strategy as Object Pose Estimation Incorporating Projection Loss and Discriminative Refinement (OPEPL-DR). For better clarity, Figures 7, 8, 9 shows the architecture of each method.

To evaluate the proposed pose estimation systems, we conduct our experiments on Intel (R) Core (TM) i7-9700 @

**TABLE 1.** Comparison between the proposed method and state-of-the-art approaches in the ADD metric on the LINEMOD dataset.

	State-of-the-Art				Ours		
	BB8[11]	PoseCNN[20]	DPOD[18]	PvNet[9]	OPEPL	OPEPL-R	OPEPL-DR
Ape	40.4	27.8	53.28	43.62	65.52	65.33	<b>65.52</b>
Bench vise	91.8	68.9	95.34	99.9	100	99.9	<b>100.00</b>
Cam	55.7	47.5	90.36	86.86	94.12	94.12	<b>94.22</b>
Can	64.1	71.4	94.1	95.47	<b>98.23</b>	97.64	97.74
Cat	62.6	56.7	60.38	79.34	86.93	<b>87.62</b>	87.52
Driller	74.7	65.4	97.72	96.43	97.82	98.41	<b>98.41</b>
Duck	44.3	42.8	66.01	52.58	66.38	<b>67.51</b>	67.23
Egg box	57.8	98.3	99.72	99.15	99.81	<b>100</b>	99.81
Glue	41.2	95.6	93.83	95.66	92.08	98.84	<b>98.84</b>
Hole p.	67.2	50.9	65.83	81.92	87.54	88.39	<b>88.39</b>
Iron	84.7	65.6	99.8	98.88	99.39	99.18	<b>99.39</b>
Lamp	76.5	70.3	88.11	99.33	99.42	99.52	<b>99.52</b>
Phone	54	54.6	74.24	92.41	94.81	<b>95.39</b>	95.29
Average	62.69	62.75	82.98	86.27	90.93	<b>91.68</b>	<b>91.68</b>

**TABLE 2.** Comparison between the proposed method and state-of-the-art approaches in the 2D projection metric on the LINEMOD dataset.

	State-of-the-Art			Ours		
	BB8[11]	PoseCNN[20]	PvNet[9]	OPEPL	OPEPL-R	OPEPL-DR
Ape	96.6	83	99.23	99.05	99.04	99.05
Bench vise	90.1	50	99.81	99.61	99.52	99.71
Cam	86	71.9	99.21	99.12	99.12	99.12
Can	91.2	69.8	99.9	99.8	99.7	99.70
Cat	98.8	92	99.3	99.4	99.8	99.90
Driller	80.9	43.6	96.92	97.82	97.62	97.62
Duck	92.2	91.8	98.02	99.06	99.06	99.06
Egg box	91	91.1	99.34	99.15	99.15	99.15
Glue	92.3	88	98.45	99.23	96.33	96.33
Hole p.	95.3	82.1	100	99.81	99.8	99.81
Iron	84.8	41.8	99.18	99.39	99.28	99.28
Lamp	75.8	48.4	98.27	97.7	97.79	97.70
Phone	85.3	58.8	99.42	99.33	99.33	99.33
Average	89.25	70.18	99.00	<b>99.11</b>	98.89	98.90

3.0GHz with an NVIDIA GeForce RTX 2070 graphic card, under Python 3.6 that utilizes PyTorch v1.1 and NVIDIA CUDA 10 library for parallel computation.

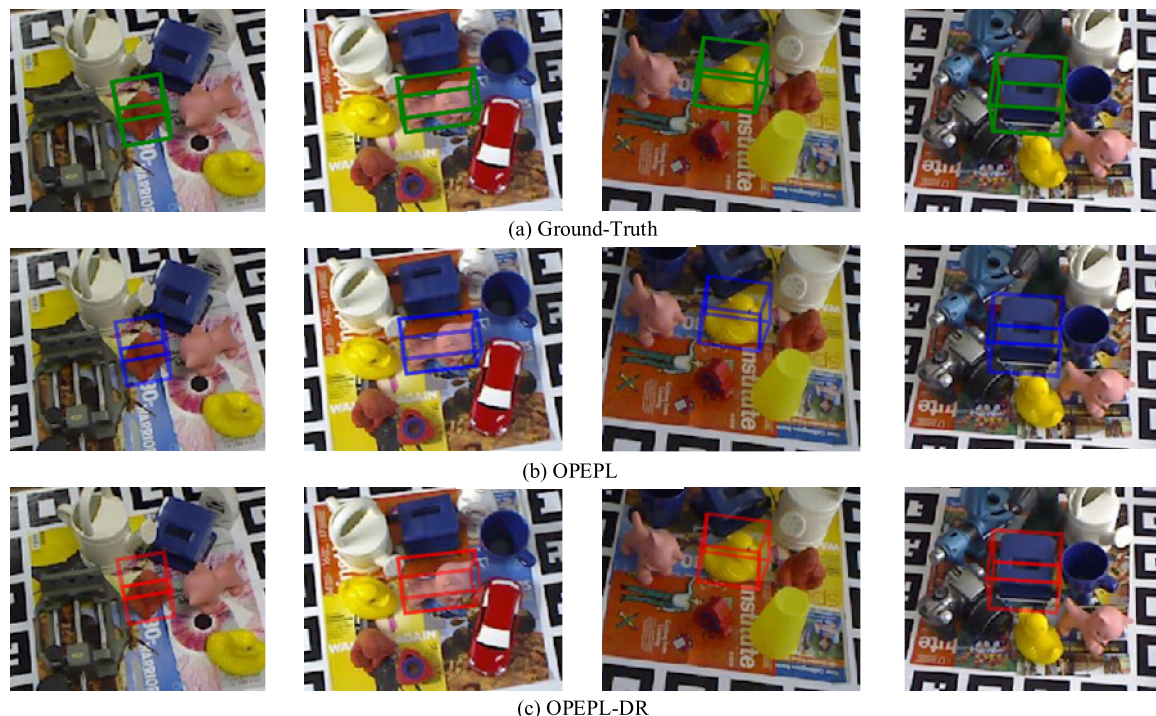
## B. DATASETS

We conduct experiments on two well-known 6-degree of freedom (DOF) pose estimation datasets, i.e., LINEMOD [4] and Occlusion LINEMOD [6] datasets. LINEMOD contains 13 objects and exhibits various estimation challenges, including texture-less objects and complex environments, whereas Occlusion LINEMOD uses part of the objects in LINEMOD to build up occlusion scenarios. We follow a prior work [9] to split the LINEMOD data: 15% of images for training and 85% for testing. In addition to the

original LINEMOD dataset, we also expanded the training data based on the LINEMOD dataset. For each single object, we rendered 10000 images with random backgrounds and transformed poses based on the available point cloud data of the LINEMOD dataset. Then, we generate 10000 fused images by randomly selecting multiple objects and randomly transforming the poses of the objects. Thus, the training data we used contain the original LINEMOD training set, the 10000 rendered images, and the 10000 fused images.

## C. EVALUATION METRICS

The ADD [4] and 2D Projection metric [29] are widely used metrics for evaluating the performance of the 6-DOF pose estimation. In this work, we adopt both of them to



**FIGURE 10.** Object pose estimation results represented by the bounding boxes are drawn in the corresponding scene images from the LINEMOD dataset by various methods. Target objects in the column from left to right are Ape, Cat, Duck, and Hole puncher.

evaluate the proposed pose estimation systems. The ADD metric measures the average 3D distance between the points transformed from the estimated pose and ground-truth pose. For symmetric objects, we adopt ADD-S [20] to calculate the distance. Once the distance is less than 10% of model’s diameter, the estimated pose is considered correct. The 2D projection metric measures the distance between the projection model of the estimated and ground-truth poses. If such a distance is less than 5 pixels, the estimated pose is considered correct.

**D. IMPLEMENTATION DETAILS**

The training data are expanded based on the LINEMOD dataset for better learning performance. To enhance the diversity of the training set, we apply data augmentation strategies, including random image cropping and rotation. We follow a prior work [9] by revising ResNet-18 as the feature estimation network, to work together with the refinement network shown in Figure 3. We set the initial learning rate as 0.0001 and halve it every 20 epochs in both networks. Each model of different objects is trained for 200 epochs in the OPEPL network. As for the refinement network, each object is trained for a manually selected epoch number until the training loss converges within an acceptable value.

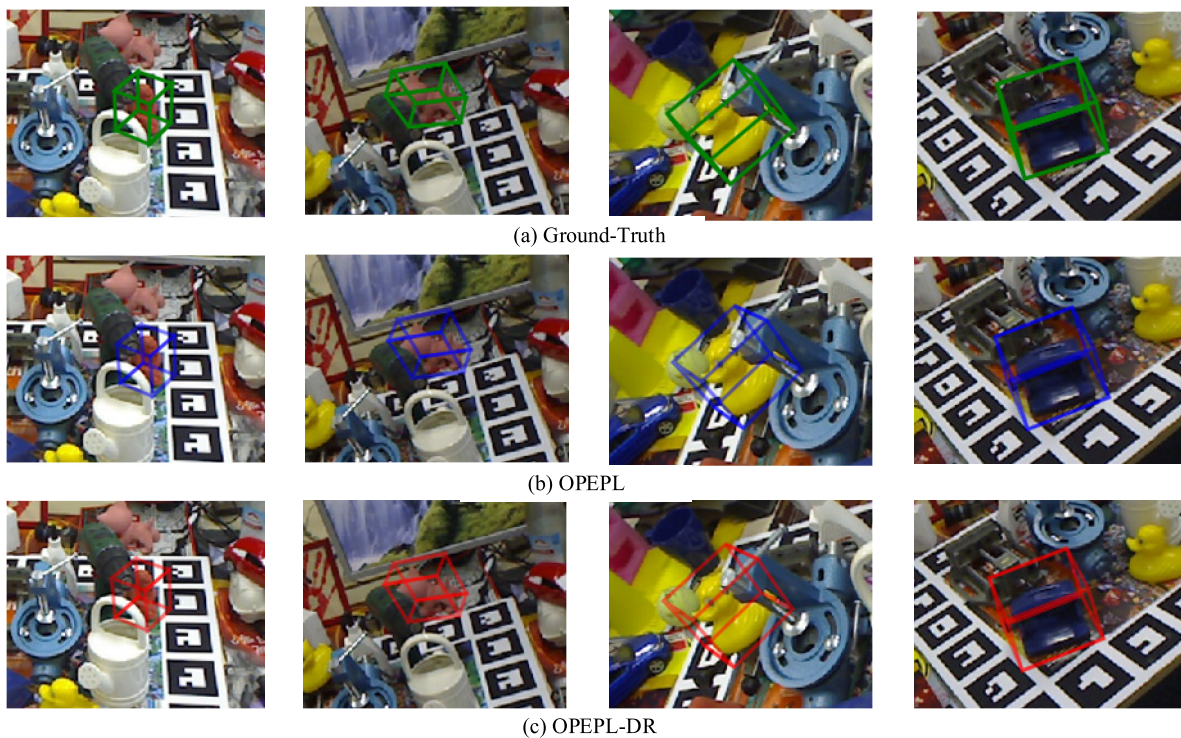
**E. COMPARISON RESULTS AGAINST THE STATE-OF-THE-ART METHODS**


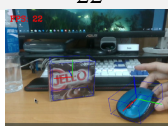

In Table 1 and Table 2, we compare our methods with BB8 [11], PoseCNN [20], DPOD [18], and PvNet [9] in

ADD and 2D projection metrics on the LINEMOD dataset, respectively. From Table 1, we can see that the accuracy of the proposed three methods are all better than those of the State-of-the-Art methods. OPEPL reaches 90.93% in ADD metric, which is better than PvNet [9]. This means that adding the projection loss does helps to improve the accuracy of pose estimation. Conversely, the OPEPL-R reaches 91.68%, indicating that the proposed refinement network successfully improves the final pose estimation. Note that the average accuracy of OPEPL-DR is same as that of OPEPL-R, because the pose refinement process is activated by the proposed discriminative strategy for most of the test scenes in the LINEMOD dataset. With reference to Table 2, although the performance of PvNet is slightly better than that of the proposed OPEPL-R and OPEPL-DR in 2D projection metrics, OPEPL with 99.11% accuracy is still the best among all the methods. To evaluate the proposed methods in occlusion scenarios, Table 3 shows a comparison of the proposed methods with YOLO6D [19], PoseCNN [20], and PvNet [9] in the ADD metric on the Occlusion LINEMOD dataset. From Table 3, we can see that OPEPL, OPEPL-R, and OPEPL-DR reach 41.71%, 42.21%, and 42.33% accuracy, respectively, outperforming the state-of-the-art methods. Moreover, the performance of OPEPL-DR is more accurate than that of OPEPL-R, indicating that the proposed discriminative strategy can make a better choice as to when to launch the refinement network for occlusion scenarios. Figures 10 and 11 show some of the pose estimation results using OPEPL and OPEPL-DR compared with the

**TABLE 3.** Comparison between the proposed method and state-of-the-art approaches in the ADD metric on the occlusion LINEMOD dataset.

	State-of-the-Art			Ours		
	YOLO6D[19]	PoseCNN[20]	PvNet[9]	OPEPL	OPEPL-R	OPEPL-DR
Ape	2.48	9.6	15.81	<b>22.39</b>	19.49	20.34
Can		45.2	63.6	67.36	67.69	<b>67.94</b>
Cat	0.67	0.93	16.68	20.22	19.97	<b>20.30</b>
Driller	7.66	41.4	65.65	66.47	66.39	<b>66.8</b>
Duck	1.14	19.6	25.24	28.48	<b>30.67</b>	30.15
Egg box		22	50.17	38.64	<b>38.81</b>	38.13
Glue	10.08	38.5	49.62	46.62	<b>51.05</b>	50.83
Hole p.	5.45	22.1	39.67	43.51	43.6	<b>44.18</b>
Average	6.42	24.92	40.81	41.71	42.21	<b>42.33</b>

**FIGURE 11.** Object pose estimation results represented by the bounding boxes are drawn in the corresponding scene images from the Occlusion LINEMOD dataset by various methods. Target objects in the column from left to right are Ape, Cat, Duck, and Hole puncher.**TABLE 4.** Running speed on multi-object pose estimation.

	Single Object	Two objects	Three Objects
<i>fps</i>	24	22	20
			

ground truth on the LINEMOD and Occlusion LINEMOD datasets, respectively. Moreover, OPEPL and OPEPL-DR can make appealing pose estimates closer to the ground truth.

#### F. PRACTICAL IMPLEMENTATION FOR REAL-WORLD SCENARIOS

To evaluate the proposed method in practical real-world scenarios, we create our own training data on 3 different objects, including a mouse and two boxes with different sizes. Moreover, we expand the proposed network dealing with a single object to train a pose estimation network for multiple objects by adding output layers to ResNet-18 to detect multiple objects via semantic segmentation. The experiment is conducted on a personal computer with Intel (R) Core (TM) i7-9700 @ 3.0GHz, an NVIDIA GeForce RTX 2070 graphic card, and a Logitech C920 webcam. Table 4 shows the running speed of the proposed pose estimation system for



multiple objects appearing on a desk. We can see that the proposed method can simultaneously estimate the objects in real time, achieving 24 fps and 20 fps in estimating a single object and 3 objects, respectively. Thus, the proposed method can estimate poses for multiple objects in practical real-world scenarios.

## V. CONCLUSION

In this work, we developed an object pose estimation system that integrates a projection loss and a pose refinement network to the RGB-based pose estimator. The projection loss reduces the position error of feature points by voting from vector spaces so that the PnP solver can provide an accurate pose estimate. Moreover, the refinement network controlled by the proposed discriminative strategy can dynamically improve the accuracy of the object pose with 2D and 3D information effectively. The experimental results show that the proposed method reaches 91.68% and 42.33% accuracy in the ADD(-S) metric on the LINEMOD and Occlusion LINEMOD datasets, respectively, outperforming the state-of-the-art methods. To solve practical real-world problems, an adaptive grasping strategy for robot arms based on the proposed pose estimation methods are currently under investigation.

## ACKNOWLEDGMENT

The authors are grateful to the National Center for High-Performance Computing for computer time and facilities to conduct this research.

## REFERENCES

- [1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.
- [2] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1510–1519.
- [3] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2011–2018.
- [4] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Nov. 2012, pp. 548–562.
- [5] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2441–2448.
- [6] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3d object coordinates," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland: Springer, 2014, pp. 536–551.
- [7] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot. Sci. Syst.*, Seattle, WA, USA, 2009.
- [8] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3338–3347.
- [9] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4561–4570.
- [10] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3109–3118.
- [11] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3828–3836.
- [12] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.
- [13] M. Sundermeyer, Z. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 699–715.
- [14] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 683–698.
- [15] P. Castro, A. Armagan, and T.-K. Kim, "Accurate 6D object pose estimation by pose conditioned mesh reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 4147–4151, doi: 10.1109/ICASSP40776.2020.9053627.
- [16] C.-M. Lin, C.-Y. Tsai, Y.-C. Lai, S.-A. Li, and C.-C. Wong, "Visual object recognition and pose estimation based on a deep semantic segmentation network," *IEEE Sensors J.*, vol. 18, no. 22, pp. 9370–9381, Nov. 2018.
- [17] A. Gadwe and H. Ren, "Real-time 6DOF pose estimation of endoscopic instruments using printable markers," *IEEE Sensors J.*, vol. 19, no. 6, pp. 2338–2346, Mar. 2019.
- [18] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1941–1950.
- [19] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.
- [20] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics, Science and System XIV*. Pittsburgh, PA, USA: Carnegie Mellon Univ., Jun. 2018, doi: 10.15607/RSS.2018.XIV.019.
- [21] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3385–3394.
- [22] Z. Zhao, G. Peng, H. Wang, H.-S. Fang, C. Li, and C. Lu, "Estimating 6D pose from localizing designated surface keypoints," 2018, *arXiv:1812.01387*. [Online]. Available: <http://arxiv.org/abs/1812.01387>
- [23] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7668–7677.
- [24] S.-K. Huang, C.-C. Hsu, W.-Y. Wang, and C.-H. Lin, "Iterative pose refinement for object pose estimation based on RGBD data," *Sensors*, vol. 20, no. 15, p. 4114, Jul. 2020, doi: 10.3390/s20154114.
- [25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [26] D. G. Viswanathan, "Features from accelerated segment test (FAST)," in *Proc. 10th Workshop Image Anal. Multimedia Interact. Services*, 2009, pp. 6–8.
- [27] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
- [28] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
- [29] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3364–3372.



**JIUN-KAI YOU** received the M.S. degree in electrical engineering from National Taiwan Normal University, Taipei, Taiwan, where he is currently pursuing the master's degree with the Department of Electrical Engineering. His research interests include computer vision and robotics.



**CHEN-CHIEN JAMES HSU** (Senior Member, IEEE) was born in Hsinchu, Taiwan. He received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1987, the M.S. degree in control engineering from National Chiao Tung University, Hsinchu, in 1989, and the Ph.D. degree from the School of Microelectronic Engineering, Griffith University, Brisbane, QLD, Australia, in 1997.

He was a Systems Engineer with IBM Corporation, Taipei, for three years, where he was responsible for information systems planning and application development, before commencing his Ph.D. studies. He joined the Department of Electronic Engineering, St. John's University, Taipei, as an Assistant Professor in 1997, and was appointed to an Associate Professor in 2004. From 2006 to 2009, he was with the Department of Electrical Engineering, Tamkang University, Taipei. He is currently a Professor with the Department of Electrical Engineering, National Taiwan Normal University, Taipei. He is the author or coauthor of more than 200 refereed journal and conference papers. His current research interests include digital control systems, evolutionary computation, vision-based measuring systems, sensor applications, and mobile robot navigation. He is a Fellow of IET.



**WEI-YEN WANG** (Fellow, IEEE) received the Diploma degree in electrical engineering from the National Taipei Institute of Technology, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1990 and 1994, respectively.

In 1994, he was appointed to an Associate Professor with the Department of Electronic Engineering, St. John's and St. Mary's Institute of Technology, Taiwan. In 2004, he became a Full Professor with the Department of Electronic Engineering, Fu Jen Catholic University. From 1990 to 2006, he worked concurrently as a Patent Screening Member of the National

Intellectual Property Office, Ministry of Economic Affairs, Taiwan. Since 2003, he has been certified as a Patent Attorney in Taiwan. In 2006, he was a Professor and the Director of the Computer Center, National Taipei University of Technology, Taiwan. From 2007 to 2014, he was a Professor with the Department of Applied Electronics Technology, National Taiwan Normal University, Taiwan. From 2011 to 2013, he was the Director of the Information Technology Center, National Taiwan Normal University, Taiwan, where he is currently a Professor with the Department of Electrical Engineering. His current research interests and publications include the areas of robot control, intelligent system design, neural networks, fuzzy logic control, robust adaptive control, computer-aided design, digital control, and CCD camera-based sensors. He has authored or coauthored over 200 refereed conference and journal papers in the above areas.

Dr. Wang was elevated to a Fellow of IEEE, in 2013, for his contributions to observer-based adaptive fuzzy-neural control for uncertain nonlinear systems. He has also been named as a Fellow of IET, CACS, and RST, and was a recipient of the Best Associate Editor Award of IEEE TRANSACTIONS ON CYBERNETICS. He is currently serving as the Editor-in-Chief of the *International Journal of Fuzzy Systems*, and an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS.



**SHAO-KANG HUANG** received the B.S. degree from the Department of Technology Application and Human Resource Development, National Taiwan Normal University, Taipei, Taiwan, in 2010, and the M.S. degree in electrical engineering from National Taiwan Normal University, in 2013, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. His research interests include computer vision and machine learning.

• • •