applied sciences

Article

Vision-Guided Object Recognition and 6D Pose Estimation System Based on Deep Neural Network for Unmanned Aerial Vehicles towards Intelligent Logistics

Sijin Luo [1,2,†], Yu Liang [1,2,†], Zhehao Luo [1,2], Guoyuan Liang [1,3,*], Can Wang [1] and Xinyu Wu [1]

[1] Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
* Correspondence: gy.liang@siat.ac.cn; Tel.: +86-755-86392138
† These authors contributed equally to this work.

Abstract: Unmanned aerial vehicle (UAV) express delivery is facing a period of rapid development and continues to promote the aviation logistics industry due to its advantages of elevated delivery efficiency and low labor costs. Automatic detection, localization, and estimation of 6D poses of targets in dynamic environments are key prerequisites for UAV intelligent logistics. In this study, we proposed a novel vision system based on deep neural networks to locate targets and estimate their 6D pose parameters from 2D color images and 3D point clouds captured by an RGB-D sensor mounted on a UAV. The workflow of this system can be summarized as follows: detect the targets and locate them, separate the object region from the background using a segmentation network, and estimate the 6D pose parameters from a regression network. The proposed system provides a solid foundation for various complex operations for UAVs. To better verify the performance of the proposed system, we built a small dataset called SIAT comprising some household staff. Comparative experiments with several state-of-the-art networks on the YCB-Video dataset and SIAT dataset verified the effectiveness, robustness, and superior performance of the proposed method, indicating its promising applications in UAV-based delivery tasks.

Keywords: UAV; vision system; semantic segmentation; 6D pose estimation; intelligent logistics

check for updates

Citation: Luo, S.; Liang, Y.; Luo, Z.; Liang, G.; Wang, C.; Wu, X. Vision-Guided Object Recognition and 6D Pose Estimation System Based on Deep Neural Network for Unmanned Aerial Vehicles towards Intelligent Logistics. *Appl. Sci.* **2023**, *13*, 115. https://doi.org/10.3390/app13010115

Academic Editor: Amerigo Capria

Received: 4 November 2022
Revised: 15 December 2022
Accepted: 20 December 2022
Published: 22 December 2022

1. Introduction

Recent advances in areas such as computer vision, automation, and artificial intelligence have created new opportunities in the aviation logistics industry. Future improvements require research on the automatic, unmanned, and informational directions that will power the efficiency and quality of delivery services. Therefore, the demand for unmanned aerial vehicle (UAV)-based automatic express delivery systems has increased rapidly over the past several decades. Currently, UAVs are widely used in the fields of surveillance, aero photography, military, and others [1,2], while most research still focuses on control algorithms and path planning [3,4].

Owing to the importance and widespread adoption of target localization and recognition in complex scenes, we proposed an efficient vision system, including object detection, tracking, segmentation, and 6D pose estimation, and deployed it on a UAV platform to demonstrate its potential in the applications of automatic express delivery.

The proposed vision system can detect, locate, and track targets of interest without manual intervention by calculating precise position and pose parameters. In addition, we built a dataset to evaluate the performance of our algorithm, called the SIAT dataset, which contains 20 video sequences with a total of 13,161 frames. The ground truth of our SIAT dataset comprised segmentation masks and 6D pose parameters for all targets in each frame. As shown in Figure 1, the workflow for the entire system consists of the following stages:

- **Object detection**: The RGB-D camera on the UAV can continuously capture images during flights and then transmit them to the server via the WIFI system. The target of interest can be detected and located by the single shot multibox detector (SSD) [5] algorithm.
- **Object tracking**: Once the target is detected, the UAV will continue to track and approach it steadily.
- **Semantic Segmentation**: The semantic segmentation network processes the image combined with the color and depth information and outputs an accurate segmentation mask. Furthermore, for objects with the same textures but different 3D geometries, as described in the subsection about object classification, we introduced a classification network to distinguish them.
- **6D Pose Estimation**: The 6D pose estimation network calculates the pose parameters of the target in the segmented image and transmits them to the UAV.

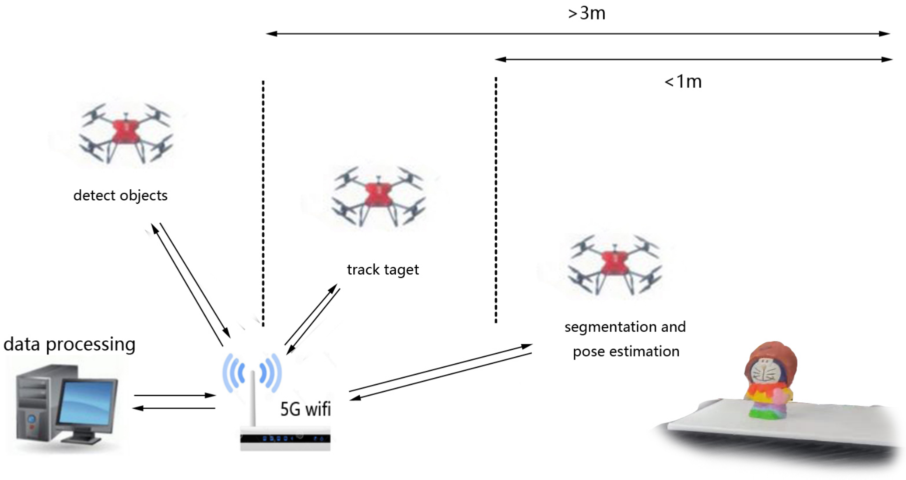


Figure 1. The hardware setup and workflow of the proposed vision system. (1) UAV: the UAV obtains pictures through the RGB-D camera, then transmits the data to the server via a WIFI system. (2) Server: the images are processed to detect, segment, classify the target of interest, and estimate its pose parameters. (3) Workflow: object detection: detect the target of interest in flight, the distance from the target is usually greater than 3 m; object tracking: approach and track the target, the distance to the target ranges from 1 to 3 m; semantic segmentation: segment the target pixels from the background and classify the target; 6D pose estimation: estimate the 6D pose parameters pixel-by-pixel in the target region. For segmentation and pose estimation, the distance is usually less than 1 m.

The details of the above are described in Section 3. The main contributions of this study are as follows.

- We built a practical vision system and reliable visual assistance of express delivery that expands collaborative work between humans and UAVs by enabling accurate localization and 6D pose parameters of the targets.
- We proposed a semantic segmentation network with a novel feature fusion structure, which provides more comprehensive semantic information for segmentation by connecting different features at different layers to fuse color and depth information.
- We proposed an innovative 6D pose estimation network that uses the pyramid pooling module (PPM) and the refined residual block (RRB) module in the color feature extraction backbone to enhance features to accurately generate 6D pose parameters for the target.

- We constructed a dataset with the ground truth of segmentation masks and 6D pose parameters to evaluate the performance of the algorithm of the system, namely the SIAT dataset.

The remainder of this paper is organized as follows: Section 2 reviews related work. The details of the system and methodology are described in Section 3. Section 4 introduces the experimental setup and indicators, followed by a discussion of comparative experimental results to prove the effectiveness and robustness of the proposed system. Section 5 concludes the paper.

2. Related Work

Automatic detection, localization, and estimation of the 6D pose of the target are virtual steps in UAV intelligent logistics. Therefore, it is important to design an accurate and efficient intelligent-vision system. Previous related research [6,7] mainly focused on target detection, obstacle avoidance, visual navigation, and environmental awareness for UAVs; however, few focused on intelligent logistics tasks. To fill this gap, our research mainly aims to customize a vision-guided system that involves object detection, target tracking, semantic segmentation, object classification, and 6D object pose estimation for UAV intelligent logistics. We review the related research on these vision tasks.

Object detection: Object detection is one of the fundamental tasks in computer vision. Traditional methods usually extract features [8,9] and use AdaBoost [10] to classify and detect targets. With the sharp development of deep learning, object detection methods based on neural networks have rapidly emerged, and the frameworks of these methods can mainly be categorized into two types: two-stage region-based methods, including [11–13], and one-stage region-free methods, including [14–17]. Two-stage region-based methods are characterized by the extraction of anchor boxes through the Region Proposal. A large number of extracted anchor boxes will cause an imbalance between positive and negative samples and affect the training and inference speed of the model. One-stage region-free methods complete category prediction and location regression simultaneously to improve the speed and reduce memory consumption. Considering the real-time advantage, we adopted the SSD algorithm in our system, which is highly efficient in detecting objects of different sizes and shapes and is robust because of the multi-scale feature maps used for prediction.

Object tracking: Object tracking is a valuable task that is widely used in sports broadcasting, security monitoring, unmanned vehicles, robots, and other fields. Traditional object tracking methods include discriminative correlation filters (DCF) [18], Kernel tracking [19], silhouette tracking [20], and point tracking [21]. Object tracking methods based on deep learning have progressively replaced traditional methods and have become mainstream in recent years. SiamFC [22] and SANet [23] are both successful representatives of deep-learning-based methods. With the proposition of attention methods, the transformer-based model shows excellent capabilities in various fields, such as natural language processing (NLP) and computer vision. TransTrack [24] and TrackFormer [25] have achieved strong performance in object tracking as a representation of the transformer-based model.

Semantic segmentation: Semantic segmentation is a pixel-level classification task that aims to segment an image into different parts. Traditional semantic segmentation methods are based on clustering approaches with additional information from the edges and contours [26]. After the full convolution neural network (FCN) [27] was proposed, methods based on deep learning have gradually dominated the semantic segmentation field. U-Net [28] defined an encoder network to extract semantic information from images and a decoder network to output pixel-level classification results. Similarly, U-Net adds skip connections to fuse the complementary information of different layers. The pyramid pooling module (PPM) is recommended in PSPNet [29] to aggregate context information from different scales to upgrade the network's ability to acquire global information. To segment objects at multiple scales robustly, various spatial pyramid pooling (ASPP) methods have been proposed by DeepLab [30]. Transformer-based research [31] on semantic segmentation is also in the limelight.

Object classification: Classification is a classic problem in the field of computer vision. Since Alexnet [32] was proposed, an era of deep learning has been proposed. With further development of the classification network, basic network structures, such as Vgg [33] and ResNet [34] have been proposed to rapidly mature the object classification field. A recently published study [35] trained a lightweight CNN network to generate mid-level features. In this study, the main task of our classification network is to deal with objects that have comparable contexts but different geometry structures; therefore, two parallel branches are designed to extract color features and depth features, respectively, and then fuse features to classify the objects.

The 6D object pose estimation: The traditional method of object pose estimation is represented by template matching [36]; however, it is difficult to apply in a messy environment. In recent years, transformational changes in deep learning hold the promise of estimating accurate, stable, and high-quality object pose parameters and encourage pose estimation methods based on neural networks to become mainstream. These methods can be roughly divided into two types: keypoint-based methods [5,37,38], which use neural networks to obtain key points first, and then compute 6D poses using the PnP algorithm [39], and regression-based methods [40,41], which directly design a single neural network to regress the 6D poses of the objects. Another group of approaches addressed the CAD model, which resulted in significant geometry and structure priors. Mask2CAD [42] learns via contrastive loss between the positive and negative pairs of image CAD. Pathch2CAD [43] builds on Mask2CAD to solve the challenges of occlusions and new perspectives. The method [44] proposed a semantic segmentation network that fuses multi-scale features in a densely connected manner. The last proposed method, called FS6D [45], uses object pose estimation as a feature-matching problem based on the feature-matching technique SuperGLUE [46].

3. System and Methodology

3.1. Hardware Setup

The staple hardware devices in the vision system include a server with NVIDIA GTX1080Ti GPU and 3.6 GHz Intel i7-6850K CPU, router, micro UAV(DJI M100), and Percipio RGB-D camera (FM830-I) mounted on the UAV. In addition to being compact, lightweight, and easy to mount and unmount, the Percipio RGB-D camera simultaneously combines structured light and binocular vision to generate depth maps with an accuracy of up to 1 mm. Note that the RGB-D camera also acts as a depth sensor to measure the distance from the UAV to the target. According to the parameters given by Tuyang, the effective range of the Percipio FM830-I RGB-D camera is 0.5–6 m. Figure 2 shows the UAV and Percipio RGB-D camera used in our vision system.

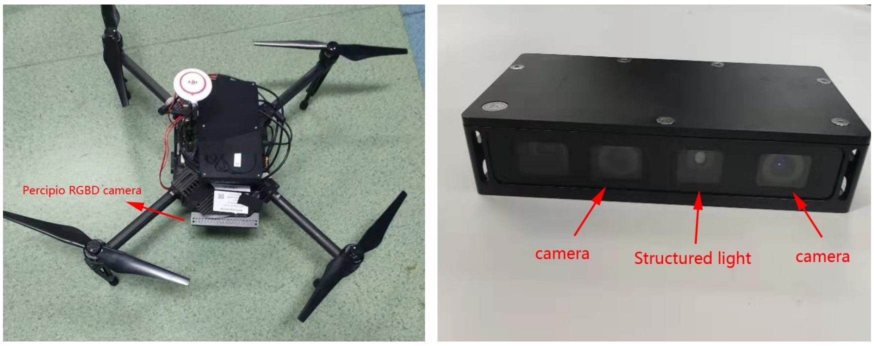


Figure 2. UAV (**left**), the Percipio RGB-D camera (**right**).

3.2. System Overview

The overall framework of the proposed system is illustrated in Figure 1. The RGB-D camera equipped on the UAV takes the image every 0.5 s and then transmits the image to the server via WIFI. We applied the Mercury-MW150R router to transmit the signal from the UAV to the server, which theoretically has an effective communication range of 300 m indoors and 800 m outdoors. On the server, these images are used to detect, track, segment objects, and estimate 6D poses. The position, size, and 6D pose parameters of the target were transmitted back to the UAV to provide visual assistance for handling tasks. The following sections describe the detection, segmentation, and object 6D pose estimation algorithms.

3.3. Object Detection

The system was applied to UAVs, which have high real-time requirements. In addition, for safety reasons, in an actual scenario, the UAVs need to maintain a certain distance from the target, resulting in a limited region of interest obtained by the detector. To deal with these challenges, we proposed a strategy: cropping the image from 960 × 1280 to a small size during the object detection stage to improve the inference efficiency of the network.

We selected the classic SSD algorithm as the object detector because of its high efficiency in detecting objects of different sizes and shapes and its robustness because of the multi-scale feature maps used for prediction. As an excellent single-stage object detection algorithm with high accuracy, compared with the previous two-stage method [47], SSD significantly improves the detection speed by canceling the bounding box proposals and consequent feature resampling. We split the original image of size 960 × 1280 into sub-images of size 300 × 300. The size of the sub-images is an empirical choice, considering the inference speed of each sub-image and the number of sub-images that need to be detected by SSD. Hereafter, multi-scale convolutional feature maps generated at different scales in SSD can predict the bounding boxes of objects in these sub-images with gratifying accuracy. Finally, the bounding boxes predicted by SSD were mapped to the original image, as shown in Figure 3.

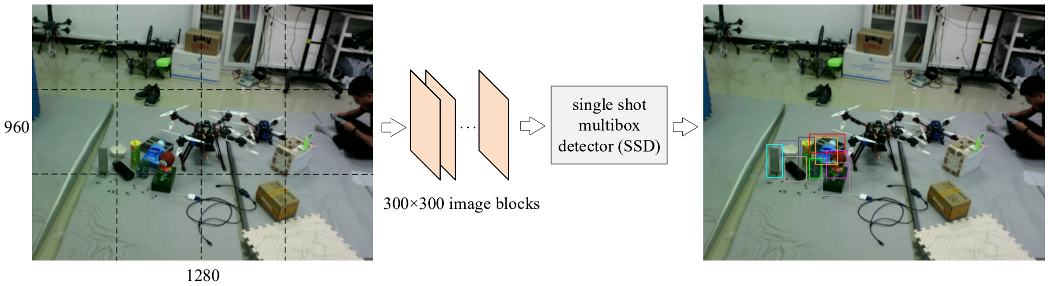


Figure 3. The process of object detection. The original image is cut into several 300 × 300 sub-images, and then the SSD algorithm is applied to each sub-image, respectively. Finally, these bounding boxes are mapped to the original image.

3.4. Object Tracking

In the tracking phase, the UAV moves to the target according to the detection results. To maintain real-time detection while tracking, with full consideration of the correlation in sequence frames, we propose an innovative strategy in which the current detection region is centered on the detected area of the previous frame and expands it twice. Finally, the size of the cropped image was adjusted to 300 × 300 pixels as the input of the detector to output the current real-time detection results of each frame.

3.5. Semantic Segmentation

As a significant component of the proposed visual system, the semantic segmentation algorithm is a system comprising multiple networks. Figure 4 illustrates this framework. The segmentation system distinguishes detected objects from the background at the pixel level to facilitate subsequent pose estimation. The black plate with three round marks in addition to the object is a calibration plate used to generate the SIAT datasets and works in the same way in all figures in the following text.

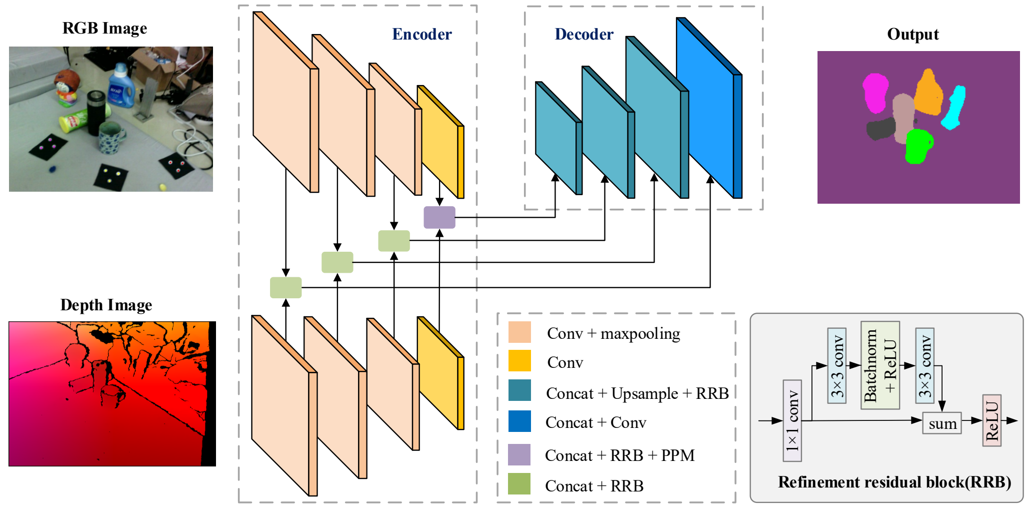


Figure 4. The overall framework of the segmentation network. The encoder network uses convolutional layers and MaxPooling layers to extract features of different scales from RGB and depth images. The decoder layer uses convolution and Upsampling to output pixel-level classification results.

During tracking, the RGB-D camera acted as a depth sensor to measure the distance from the UAV to the server. Semantic segmentation began when the UAV approached the target within 1 m. We introduced a novel segmentation algorithm based on an encoder-decoder structure to obtain a precise segmentation mask. Our semantic segmentation network was inspired by the classical segmentation network U-Net [28] and consisted of an encoder network and a decoder network. The convolutional and max-pooling layers compose the encoder network to extract features at different scales, whereas the decoder layer applies convolution and upsampling to output the pixel-level classification results. In addition, because the features of the encoder and decoder layers contain complementary information from different layers in the convolutional network, a skip network is applied to fuse this information to improve the network performance.

Compared with the prior network, the innovation of our network is reflected in the following two points:

- To better utilize the geometric information of the environment, in addition to the RGB images, we input the depth images into the segmentation network. Therefore, our encoder network consists of two branches, with one branch extracting color features and the other extracting geometric features. Next, the color and geometric features of the Max-Pooling layer will be concatenated at each downsampling stage to reinforce the expressiveness of the features.
- We introduced the PPM [29] to aggregate contextual information at different scales to reduce computation. In addition, to strengthen the recognition ability of each stage, we added RRB [48] to refine the feature map, the details of which are shown in Figure 4.

In the training stage, cross-entropy is used as the loss function of the segmentation network and is expressed as follows:

$$L = -\sum_{i=1}^{H}\sum_{j=1}^{W} y'_{i,j} \log y_{i,j}, \tag{1}$$

where $y_{i,j} \in \{1, 2, \ldots, C\}$ is the ground truth of each pixel and $y'_{i,j}$ is the prediction. H and W denote the height and width of the image, respectively.

In summary, the proposed semantic segmentation network can achieve pixel-level classification by fusing color and depth information, which significantly improves the accuracy with a slight increase in the computational cost. However, the segmentation network performs unsatisfactorily on objects with similar or the same textures but diverse geometries. Therefore, we leveraged a classification network to address this problem.

3.6. Object Classification

As mentioned in the segmentation network section, segmentation networks often cannot accurately predict segmentation masks for objects with similar textures but different geometries.

Figure 5 shows that the left image is a cup with a handle, and the right image is the other side of the cup, in which the handle cannot be seen. Although the handle of the cup has the same pattern as the cup body, it cannot be separated using color information alone. We designed a new classification network that introduced depth information to classify two objects.

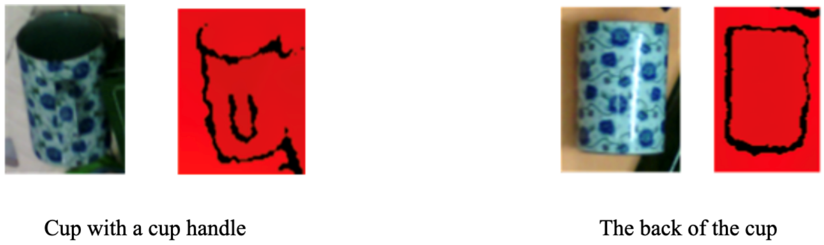


Figure 5. Objects with the same textures and different shapes, the **left** object is a cup with a handle and the **right** object is another side of the cup with the handle hidden. The depth image is next to the color image.

According to the segmentation results of the previous step, the target regions were cropped as inputs into the classification network. The structure of the classification is shown in Figure 6: two convolutional networks extracting color and geometric features, a fuse layer for the two types of features, and a fully connected layer to obtain the result. In each branch, the features were first extracted through the convolutional and max-pooling layers. Subsequently, after the fourth layer of convolution, the color feature vector and depth feature vector of the same dimension are obtained. They are multiplied element by element to fuse the final features and feed them into a fully connected layer. The outputs were concatenated and eventually fed into the pose predictor.

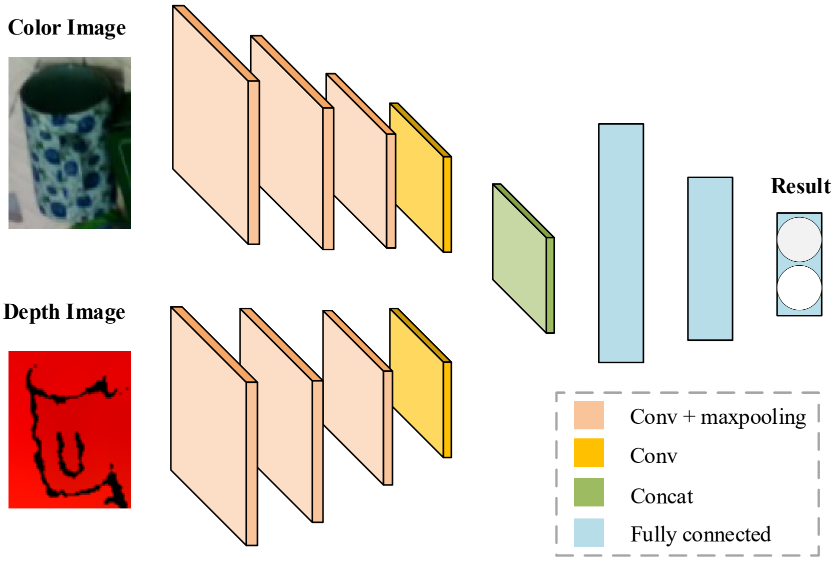


Figure 6. The overall framework of the classification network. Two convolutional networks are designed to extract color and geometric features, respectively, then fuse those features, and utilize fully connected layers to obtain results.

3.7. Object 6D Pose Estimation

We estimate the object 6D poses through the transformation matrix between the object coordinate system and the camera coordinate system, which comprises the rotation matrix R and translation vector t. This concept is illustrated in Figure 7.

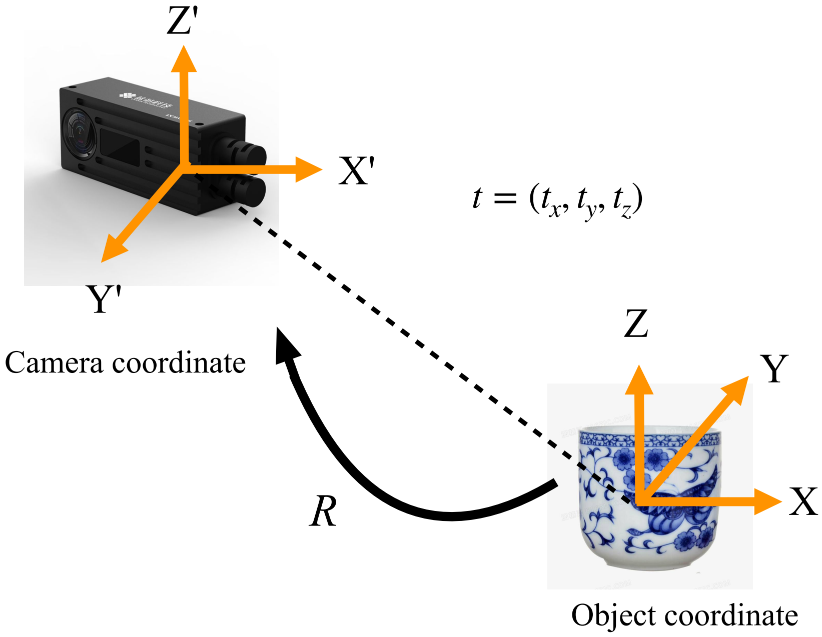


Figure 7. Illustration of object 6D pose estimation. The pose transformation matrix between the object coordinate system and the camera coordinate system is composed of the rotation matrix R and the translation vector t.

To solve the rotation matrix R and translation vector t requires sufficient geometric information; we converted the region of the target in the depth map to a point cloud according to the following formula:

$$\begin{aligned} z &= d/s \\ x &= (x' - c_x) * z/f_x \\ y &= (y' - c_y) * z/f_y \end{aligned}$$

where d and s denote depth and scale factors, respectively. x' and y' represent the coordinates of pixels in the depth map, and c_x, c_y, f_x, f_y represent the intrinsic parameters of the camera, and (x, y, z) are the corresponding point coordinates calculated. By converting each pixel in the depth map, as described above, the data of the point cloud can be obtained. The trimmed RGB image and its corresponding point cloud data are then fed together into the 6D Pose Estimation network.

The overall framework of our network is shown in Figure 8. We use a feature extracting backbone based on ResNet18 to map pixels to the color feature embeddings. Our backbone has made some improvements to ResNet18; for details, the output of layer2, layer3, and layer4 and the output of PPM go through an RRB module to strengthen the features. The features from different scales are then fused, which is beneficial for pose estimation. Meanwhile, a geometric feature extraction backbone based on PointNet is also constructed to map 3D points to geometric feature embeddings. With the pixel-wise fused color and geometric feature embeddings, the convolution layers will regress the 6D pose parameters $T_i(i = 1, 2, \ldots, N)$ and confidence scores $c_i(i = 1, 2, \ldots, N)$ for each pixel. The pose with the maximum confidence score is regarded as prediction T_{pre}. To improve the pose-estimation accuracy, we use the iterative refinement network proposed by denseFusion [41] to obtain the refined pose.

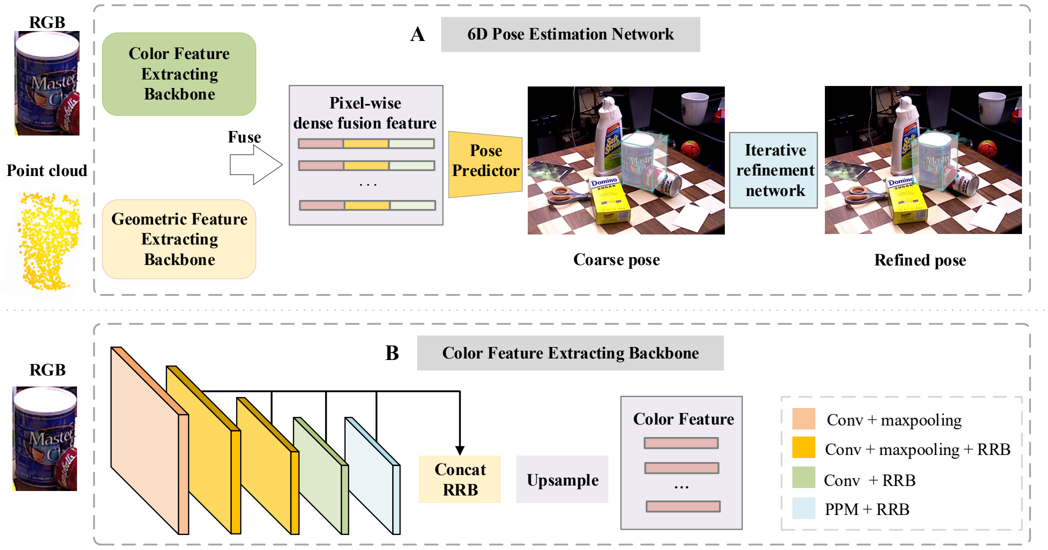

RGB
Point cloud
A
6D Pose Estimation Network
Color Feature Extracting Backbone
Geometric Feature Extracting Backbone
Fuse
Pixel-wise dense fusion feature
Pose Predictor
Coarse pose
Iterative refinement network
Refined pose
RGB
B
Color Feature Extracting Backbone
Concat RRB
Upsample
Color Feature
Conv + maxpooling
Conv + maxpooling + RRB
Conv + RRB
PPM + RRB

Figure 8. The overall framework of the 6D pose estimation network. (**A**) 6D pose estimation network: Firstly, the color and geometric features are extracted from two backbones and fused at the pixel level. Then, a predictor is used to regress the pose. Finally, the iterative refinement network is used to refine the pose. (**B**) Color Feature Extracting Backbone: Our backbone is based on ResNet18, for detail, the output of layer2, layer3, and layer4 and the output of PPM go through an RRB module to strengthen the features. Then the features are fused from different scales to improve the expressiveness of features.

We designed two loss functions in the training stage to deal with symmetric and asymmetric targets. For asymmetric ones, each pose corresponds to the unique shape or texture of the object; therefore, we define the loss as the average distance between the points sampled on the object model transformed by the ground truth pose and corresponding points transformed by the predicted pose. The loss function is defined as follows:

$$L_i = \frac{1}{M}\sum_{j}\left\|(R_{gt}x_j + t_{gt}) - (R^i_{pred}x_j + t^i_{gt})\right\|$$

where x_j represents the homogeneous form of the j-th point of M randomly selected points from the 3D model of the object. $p_{gt} = [R_{gt}|t_{gt}]$ is the ground truth pose, and R_{gt} and t_{gt} are the rotation matrix and translation vector, respectively. In addition, $p_{pred}i = [R_{pred}i|t_{pred}i]$ is the current predicted pose of the i-th pixel.

For targets with a symmetric structure, one texture or shape can theoretically be associated with multiple or infinite poses. Therefore, to avoid ambiguous learning objectives, loss is defined as the average distance between the points transformed by the predicted pose and their closest points on the object model transformed by the ground truth pose. The loss function is defined as follows:

$$L_i = \frac{1}{M}\sum_{j}\min_{0<k<M}\left\|(R_{gt}x_j + t_{gt}) - (R^i_{pred}x_k + t^i_{pred})\right\|$$

4. Experiments

4.1. Datasets

YCB-Video Dataset The YCB-Video dataset [49] is a benchmark for the task of object 6D pose estimation, which contains 92 RGB-D video sequences and 133,827 frames of 21 items with different shapes and textures. Each frame includes one segmentation mask and a 6D pose of all target objects in the image. For a fair comparison with previous work, we set the same training and testing data, where 80 video sequences for training and 2949 keyframes selected from the remaining 12 video sequences were used for evaluation. All methods mentioned below are based on the same training data and test data to ensure fairness. In addition, 80,000 synthetic images released by [49] were used to train our network.

SIAT Dataset To verify the robustness of the proposed method, we used an RGB-D video dataset, the SIAT dataset. As shown in Figure 9, the SIAT dataset comprises 20 video sequences of 10 items with different textures and shapes, with 18 for training and the remaining two for evaluation. Each video includes hundreds of frames captured by a Percipio RGB-D camera (FM830-I). The 3D model of each item in the SIAT dataset was obtained using the method in [50]. Using a motion capture system, OptiTrack, an accurate 6D pose between the camera coordinate system and object coordinate system could be obtained. An experimental site equipped with OptiTrack and its corresponding software interface is shown in Figure 10. Once the 6D pose parameter of an object is obtained, the segmentation mask of the object can be obtained using the camera model.

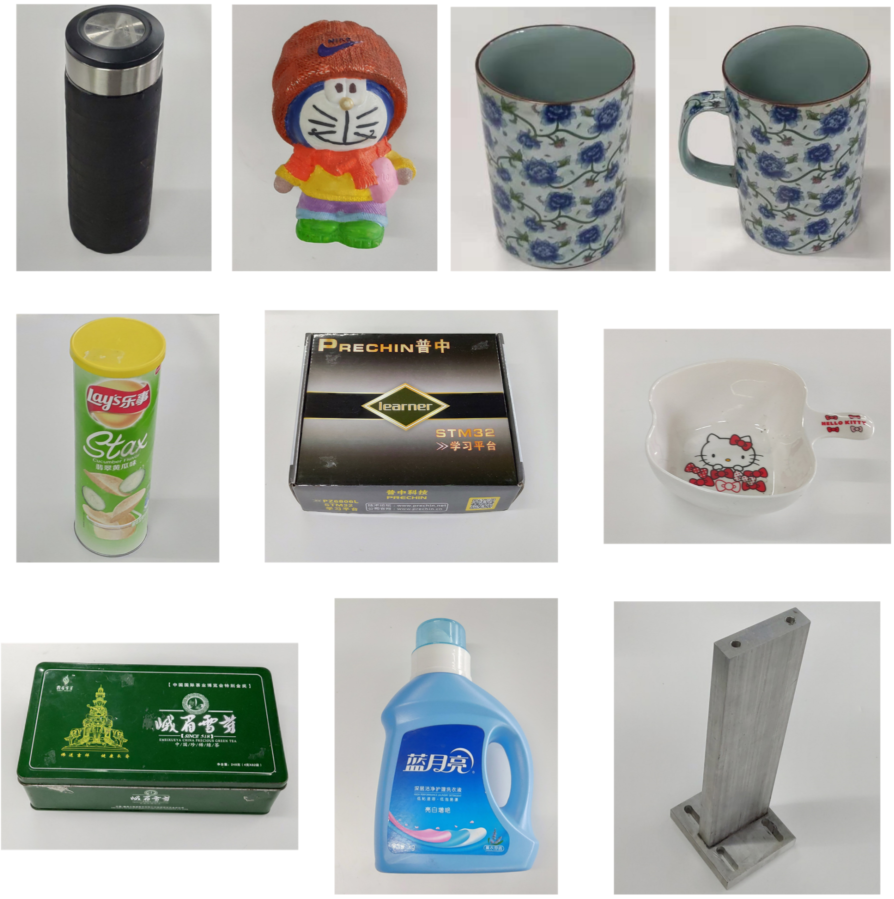
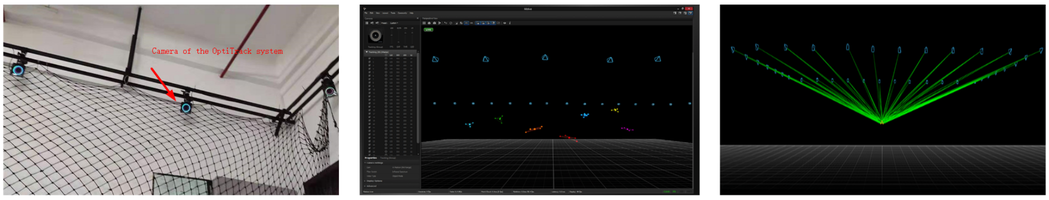


Figure 9. The objects present in the SIAT dataset.



The Motion Capture System

Software interface of the OptiTrack system

The fluorescent ball is located by the camera of the OptiTrack system

Figure 10. The Motion Capture System, OptiTrack, and its corresponding software interface.

4.2. Metrics

Metrics for Segmentation Two widely used metrics, mean intersection over union (mIOU) and mean pixel accuracy (mPA), were utilized to evaluate the segmentation network proposed in this study. mIOU is the average of IOUs of all classes, in which the IOU of each class is calculated by dividing the intersection of the predicted area and actual ground truth area by the union of the predicted and ground truth areas. The mPA is the average pixel accuracy of all classes, in which the pixel accuracy of each class is defined as the proportion of accurate pixels to the total pixels.

Metrics for Pose Estimation ADD-S (average closest distance) and AUC (area under the ADD-S curve) were adopted to evaluate the performance of the 6D pose estimation network proposed in the study on the SIAT dataset and YCB-Video dataset. The ADD-S is defined as the average distance between each point of the 3D object model transformed by the ground truth pose P_{gt} and the nearest point of the 3D object model transformed by P_{pred}. Accuracy was defined as the percentage of testing samples with an ADD-S below a certain threshold. In this study, we report the accuracy of an ADD-S of less than 2 cm. In addition, an accuracy threshold curve of the ADD-S can be obtained by adjusting the threshold. The AUC is defined as the area under the curve from 0 to a certain threshold. For a fair comparison, we followed previous work to set this threshold to 0.1 m.

4.3. Experiments on the SIAT Dataset

Semantic segmentation Table 1 shows the segmentation accuracy of the proposed segmentation network and other state-of-the-art methods on the SIAT dataset. As shown in Table 1, the proposed segmentation network is much better than the other methods for both mIOU and mPA metrics, which proves the overall performance of our network. Figure 11 shows some qualitative results of our segmentation network and other methods, from which it can be observed that the results of our method are more accurate than those of other methods, especially at the edges of the objects. Furthermore, for the texture-less object in the first and second columns, it is difficult to obtain accurate and complete predictions using the previous state-of-the-art methods. Our segmentation results after fusing the depth information can better fit the ground truth.

Table 1. Quantitative evaluation of segmentation on SIAT Dataset.

	mIOU	**mPA**
U-Net [28]	57.2	94.2
DeepLabV3 [51]	61.3	95.0
PSPNet [29]	65.2	95.6
Ours	70.6	96.1

Classfication The accuracy of our classification network reached 96.4%, which proves its robustness in dealing with similar objects, especially for objects with the same textures but different geometrical structures, such as cups from different perspectives, which cannot be distinguished from other segmented networks.

6D Pose estimation Table 2 shows the accuracy comparison of our 6D pose estimation network and DenseFusion on the SIAT dataset, where Dense(per) represents the DenseFusion method without an iterative refinement network and Dense(iter) is the DenseFusion method using an iterative refinement network. Ours uses the same method as abbreviated. As shown in Table 2, our results without iterative refinement are close to the refined results of DenseFusion, and our refined result is significantly better than that of DenseFusion, which proves the accuracy of our method. Some quantitative results of DenseFusion and our pose-estimation network are shown in Figure 12. It can be observed that our method can obtain better results overall, especially for the untextured object in the fourth column of images. The estimation of DenseFusion has obvious errors, and our method can yield more accurate results.

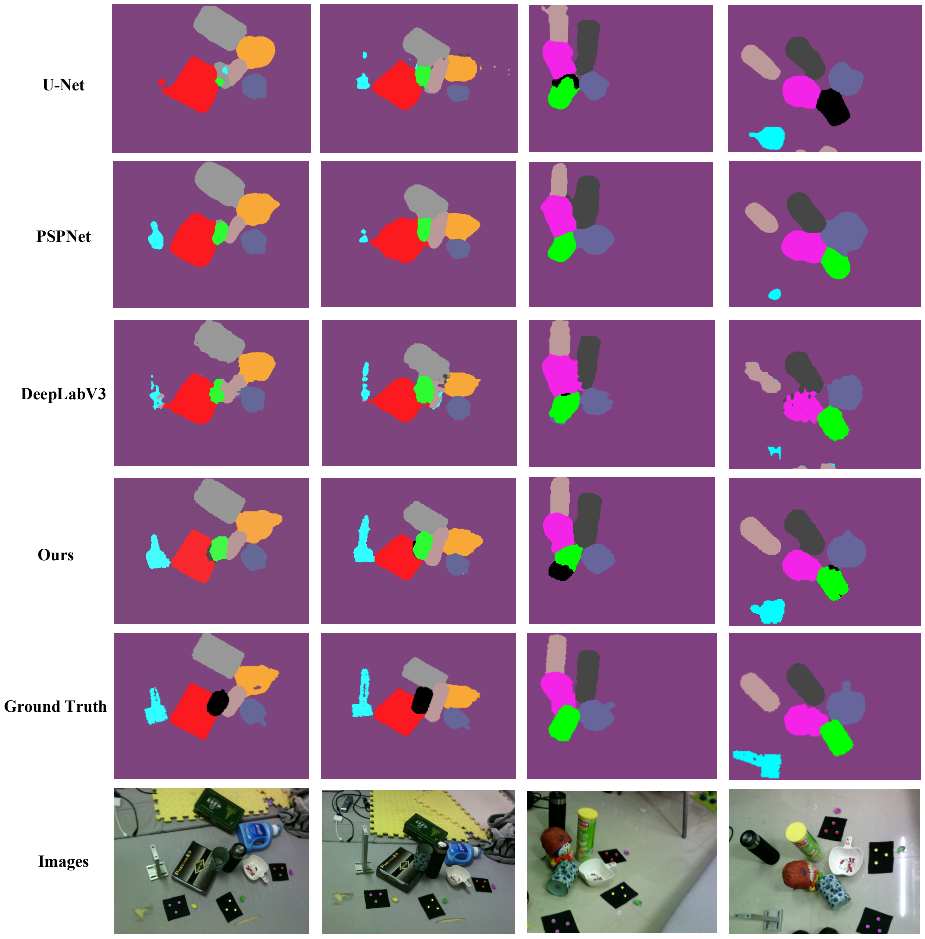
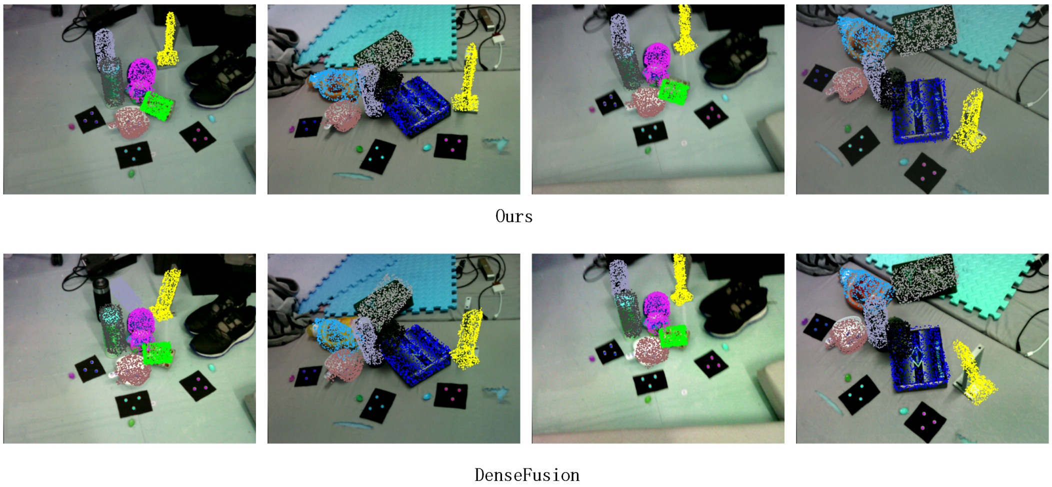


Figure 11. Some qualitative results of our segmentation network and other state-of-the-art segmentation networks. Different colors represent different categories.

Table 2. Quantitative evaluation of 6D pose on SIAT Dataset.

	Dense(per)		**Dense(iter)**		**Ours(per)**		**Ours(iter)**	
	AUC	**ADD-S**	**AUC**	**ADD-S**	**AUC**	**ADD-S**	**AUC**	**ADD-S**
toy	97.4	88.6	97.4	88.6	88.8	97.2	88.8	97.8
Lay's	68.0	73.8	72.7	75.3	76.0	74.0	78.6	82.2
bowl	97.4	91.5	97.4	91.5	92.2	97.7	92.3	99.1
Thermos cup	50.0	58.9	50.0	58.9	73.6	61.6	73.6	61.6
Tea box	69.2	82.0	69.2	82.0	75.2	68.8	79.5	68.8
Blue moon	52.2	75.7	60.8	76.4	78.1	62.5	84.0	73.1
Metal block	64.3	78.5	64.5	78.5	76.7	64.4	81.4	76.9
Carton	71.7	83.4	71.7	83.4	75.1	66.1	80.1	75.3
cup	96.3	85.9	97.6	87.6	87.9	97.7	88.9	99.5
back of cup	92.7	88.2	92.7	88.2	87.7	94.0	90.1	98.1
MEAN	75.7	79.5	77.3	81.0	81.4	78.7	83.7	83.2



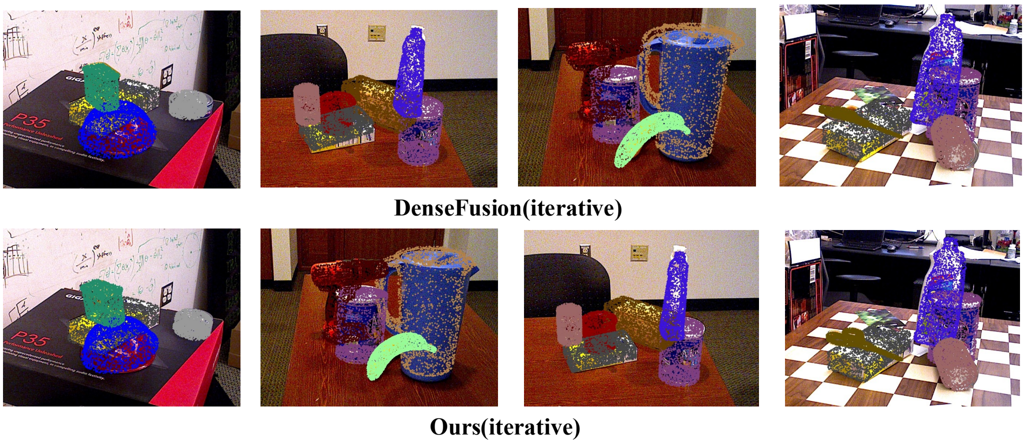
Ours

DenseFusion

Figure 12. Quantitative evaluation of DenseFusion and the pose estimation network proposed in the paper on SIAT pose dataset.

4.4. Experiments on the YCB-Video Dataset

The 6D Pose estimation Table 3 shows the accuracy comparison of the proposed 6D pose estimation network in this study and other state-of-the-art methods on the YCB-Video dataset, where Dense(per) denotes the DenseFusion method without an iterative refinement network, Dense(iter) is DenseFusion using an iterative refinement network, and our method is also abbreviated similarly. It can be observed that our method significantly outperforms other methods on the ADD-S and AUC metrics, which confirms the advantages of our method. Figure 13 shows some quantitative results of our network and the state-of-the-art method DenseFusion on the YCB-Video dataset. The points in the image are formed by projecting the 3D points of the object into a 2D image after transformation with the predicted pose parameters, in which different colors represent different objects. The figure (Figure 13) shows that our method can estimate more accurate results, particularly in cluttered scenes, demonstrating the robustness of our network. Table 4 shows the execution time of our system. It can be seen that the transmission task takes more time than the visual processing task.



DenseFusion(iterative)

Ours(iterative)

Figure 13. In order to obtain a more intuitive comparison, some qualitative results are shown here. All results are predicted based on the same segmentation mask as PoseCNN. The dots of different colors represent objects of different categories. The name below the image indicates the name of the method.

Table 3. Quantitative evaluation of pose estimation on YCB-Video dataset.

	PoseCNN + ICP		**Dense(per)**		**Dense(iter)**		**Ours(per)**		**Ours(iter)**	
	AUC	**ADD-S**	**AUC**	**ADD-S**	**AUC**	**ADD-S**	**AUC**	**ADD-S**	**AUC**	**ADD-S**
002 master chef can	95.8	100.0	95.2	100.0	96.4	100.0	95.3	100.0	96.3	100.0
003 cracker box	92.7	93.0	92.5	99.3	95.5	99.5	92.6	100.0	96.3	100.0
004 sugar box	98.2	100	95.1	100.0	97.5	100.0	95.5	100.0	97.7	100.0
005 tomato soup can	94.5	96.8	93.7	96.9	94.6	96.9	96.8	100.0	97.7	100.0
006 mustard bottle	98.6	100.0	95.9	100.0	97.2	100.0	96.0	100.0	97.8	100.0
007 tuna fish can	97.1	97.9	94.9	100.0	96.6	100.0	96.0	100.0	97.2	100.0
008 pudding box	97.9	100.0	94.7	100.0	96.5	100.0	94.3	100.0	96.8	100.0
009 gelatin box	98.8	100.0	95.8	100.0	98.1	100.0	97.3	100.0	98.2	100.0
010 potted meat can	92.7	97.2	90.1	93.1	91.3	93.1	93.0	95.4	94.0	95.3
011 banana	97.1	99.7	91.5	93.9	96.6	100.0	93.5	96.8	97.1	100.0
019 pitcher base	97.8	100.0	94.6	100.0	97.1	100.0	93.4	99.5	97.9	100.0
021 bleach cleanser	96.9	99.9	94.3	99.8	95.8	100.0	95.0	99.7	96.7	100.0
024 bowl	81.0	58.8	86.6	69.5	88.2	98.8	84.4	73.9	88.8	96.8
025 mug	95.0	99.5	95.5	100.0	97.1	100.0	96.0	100.0	97.3	100.0
035 power drill	98.2	99.9	92.4	97.1	96.0	98.7	92.9	97.3	96.1	98.3
036 wood block	87.6	82.6	85.5	93.4	89.7	94.6	85.8	84.3	91.7	96.7
037 scissors	91.7	100	96.4	100.0	95.2	100.0	96.6	100.0	93.1	99.5
040 large marker	97.2	98.0	94.7	99.2	97.5	100.0	95.9	99.7	97.8	100.0
051 large clamp	75.2	75.6	71.6	78.5	72.9	79.2	73.7	79.2	75.7	80.1
052 extra large clamp	64.4	55.6	69.0	69.5	69.8	76.3	83.4	83.6	83.3	88.9
061 foam brick	97.2	99.6	92.4	100.0	92.5	100.0	94.8	100.0	96.4	100.0
MEAN	93.0	93.1	91.2	95.3	93.1	96.8	92.8	96.5	94.8	97.9

Table 4. Running time of all networks used in the vision system and the time of image transmission.

	Detection	**Segmentation**	**Classification**	**6D Pose Estimation**	**Image Transmission**
Time (s)	0.049	0.02	0.002	0.023	0.11

5. Conclusions

In this study, a vision system is proposed for the object-handling tasks of UAVs. The system combines 2D images and 3D point cloud information to accurately detect and segment various objects in complex scenes and realize 6D object pose estimation, which can provide a solid foundation for intelligent picking or other object handling tasks of UAVs. In addition, to evaluate the performance of the system more comprehensively, we contribute a 6D pose estimation dataset named the SIAT dataset. Experiments conducted on the SIAT dataset and benchmark YCB-Video dataset demonstrate the robustness of our system. Currently, the proposed method experiences slow image transmission. Upgrading the hardware to improve image transmission speed and researching more lightweight networks and algorithms to carry vision systems on UAVs to complete offline 6D pose estimation will be considered in future works.

Author Contributions: Conceptualization, G.L., X.W. and C.W.; methodology, G.L., S.L. and Y.L.; validation, S.L., Y.L. and Z.L.; formal analysis, S.L.; data curation, Y.L.; writing—original draft preparation, S.L.; writing—review and editing, S.L. and Y.L.; visualization, S.L. and Y.L.; supervision, C.W. and X.W.; project administration, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Joint Fund of the National Natural Science Foundation of China with Shenzhen City under grant No. U1813209, No. U2013212, and No. U1813222; in part by the Shenzhen Science and Technology Innovation Committee under grant No. JSGG20201102155000001; and the Shenzhen Key Fundamental Research Project under grant No. JCYJ20210324120400003.

Data Availability Statement: The two benchmark datasets LineMod and YCB-Video analyzed for this study can be found at http://campar.in.tum.de/Main/StefanHinterstoisser (accessed on 2 February 2022) and https://rse-lab.cs.washington.edu/projects/posecnn/ (accessed on 15 March 2022). The SIAT dataset used in this paper is built by us and will be made available to any qualified researcher. Further inquiries can be directed to the corresponding author.

Acknowledgments: Fan Chen made important contributions to an earlier version of this article. The authors would also like to thank Haoyao Chen with Harbin Institute of Technology at Shenzhen for providing the motion capture system for the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, Q.; Ye, H.; Huang, K.; Zha, Y.; Shi, L. Estimation of leaf area index of sugarcane using crop surface model based on UAV image. *Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 104–111.
2. Viguier, R.; Lin, C.C.; Aliakbarpour, H.; Bunyak, F.; Pankanti, S.; Seetharaman, G.; Palaniappan, K. Automatic Video Content Summarization Using Geospatial Mosaics of Aerial Imagery. In Proceedings of the 2015 IEEE International Symposium on Multimedia (ISM), Miami, FL, USA, 14–16 December 2015.
3. Thomas, J.; Loianno, G.; Daniilidis, K.; Kumar, V. The role of vision in perching and grasping for MAVs. In Proceedings of the Micro- & Nanotechnology Sensors, Systems, & Applications VIII, Baltimore, MD, USA, 17–21 April 2016.
4. Thomas, J.; Loianno, G.; Daniilidis, K.; Kumar, V. Visual Servoing of Quadrotors for Perching by Hanging from Cylindrical Objects. *IEEE Robot. Autom. Lett.* **2016**, *1*, 57–64. [CrossRef]
5. Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1521–1529.
6. Smolyanskiy, N.; Kamenev, A.; Smith, J.; Birchfield, S. Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4241–4247.
7. Kainuma, A.; Madokoro, H.; Sato, K.; Shimoi, N. Occlusion-robust segmentation for multiple objects using a micro air vehicle. In Proceedings of the 2016 16th International Conference on Control, Automation and Systems (ICCAS), Gyeongju, Republic of Korea, 16–19 October 2016.
8. Yan, K.; Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. *Proc. CVPR* **2004**, *2*, 506–513.
9. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
10. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
11. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
12. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 386-397. [CrossRef]
13. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1483–1498. [CrossRef]
14. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *ECCV 2016: Computer Vision—ECCV 2016, Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016;* Springer: Cham, Switzerland, 2016; pp. 21–37.
16. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
17. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef]

19. Bruni, V.; Vitulano, D. An improvement of kernel-based object tracking based on human perception. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 1474–1485. [CrossRef]
20. Xiao, C.; Yilmaz, A. Efficient tracking with distinctive target colors and silhouette. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2728–2733.
21. Lychkov, I.I.; Alfimtsev, A.N.; Sakulin, S.A. Tracking of moving objects with regeneration of object feature points. In Proceedings of the 2018 Global Smart Industry Conference (GloSIC), Chelyabinsk, Russia, 13–15 November 2018; pp. 1–6.
22. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016.
23. Fan, H.; Ling, H. SANet: Structure-Aware Network for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017.
24. Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; Alameda-Pineda, X. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv* **2021**, arXiv:2103.15145.
25. Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; Feichtenhofer, C. Trackformer: Multi-object tracking with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8844–8854.
26. Ilea, D.E.; Whelan, P.F. Image segmentation based on the integration of colour–texture descriptors—A review. *Pattern Recognit.* **2011**, *44*, 2479–2501. [CrossRef]
27. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
28. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241.
29. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
30. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
31. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 7262–7272.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Nejatishahidin, N.; Fayyazsanavi, P.; Kosecka, J. Object pose estimation using mid-level visual representations. *arXiv* **2022**, arXiv:2203.01449.
36. Zhu, M.; Derpanis, K.G.; Yang, Y.; Brahmbhatt, S.; Zhang, M.; Phillips, C.; Lecce, M.; Daniilidis, K. Single image 3D object detection and pose estimation for grasping. In Proceedings of the IEEE International Conference on Robotics & Automation, Hong Kong, Chia, 31 May–5 June 2014.
37. Tekin, B.; Sinha, S.N.; Fua, P. Real-time seamless single shot 6d object pose prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 292–301.
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. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3828–3836.
39. Lepetit, V.; Moreno-Noguer, F.; Fua, P. Epnp: An accurate o (n) solution to the pnp problem. *Int. J. Comput. Vis.* **2009**, *81*, 155. [CrossRef]
40. Doumanoglou, A.; Kouskouridas, R.; Malassiotis, S.; Kim, T.K. Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
41. Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Fei-Fei, L.; Savarese, S. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. *arXiv* **2019**, arXiv:1901.04780.
42. Kuo, W.; Angelova, A.; Lin, T.Y.; Dai, A. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *Computer Vision—ECCV 2020, Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; pp. 260–277.
43. Kuo, W.; Angelova, A.; Lin, T.Y.; Dai, A. Patch2CAD: Patchwise Embedding Learning for In-the-Wild Shape Retrieval from a Single Image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 12589–12599.
44. Liang, G.; Chen, F.; Liang, Y.; Feng, Y.; Wang, C.; Wu, X. A manufacturing-oriented intelligent vision system based on deep neural network for object recognition and 6d pose estimation. *Front. Neurorobot.* **2021**, *14*, 616775. [CrossRef]
45. He, Y.; Wang, Y.; Fan, H.; Sun, J.; Chen, Q. FS6D: Few-Shot 6D Pose Estimation of Novel Objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6814–6824.

46. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems, Proceedings of the Annual Conference on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019*; NeurIPS: Vancouver, BC, Canada, 2019; Volume 32.
47. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
48. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Learning a discriminative feature network for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1857–1866.
49. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv* **2017**, arXiv:1711.00199.
50. Zhan, S.; Chung, R.; Zhang, X.T. An Accurate and Robust Strip-Edge-Based Structured Light Means for Shiny Surface Micromeasurement in 3-D. *IEEE Trans. Ind. Electron.* **2013**, *60*, 1023–1032.
51. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.