Article

An Improved Estimation Algorithm of Space Targets Pose Based on Multi-Modal Feature Fusion

Jiang Hua ¹, Tonglin Hao ², Liangcai Zeng ¹ and Gui Yu ^{1,3,*}

¹ Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China; hua Jiang@wust.edu.cn (J.H.); zengliangcai@wust.edu.cn (L.Z.)

² School of Automation, Wuhan University of Technology, Wuhan 430081, China; Tonglin@foxmail.com

³ School of Mechanical and Electrical Engineering, Huanggang Normal University, Huanggang 438000, China

* Correspondence: Yugui@hgnu.edu.cn

Abstract: The traditional estimation methods of space targets pose are based on artificial features to match the transformation relationship between the image and the object model. With the explosion of deep learning technology, the approach based on deep neural networks (DNN) has significantly improved the performance of pose estimation. However, the current methods still have problems such as complex calculation, low accuracy, and poor real-time performance. Therefore, a new pose estimation algorithm is proposed in this paper. Firstly, the mask image of the target is obtained by the instance segmentation algorithm, and its point cloud image is obtained based on a depth map combined with camera parameters. Finally, the correlation among points is established to realize the prediction of pose based on multi-modal feature fusion. Experimental results in the YCB-Video dataset show that the proposed algorithm can recognize complex images at a speed of about 24 images per second with an accuracy of more than 80%. In conclusion, the proposed algorithm can realize fast pose estimation for complex stacked objects and has strong stability for different objects.

Keywords: space targets pose; deep neural networks; feature fusion; fast pose estimation; strong stability

Citation: Hua, J.; Hao, T.; Zeng, L.; Yu, G. An Improved Estimation Algorithm of Space Targets Pose Based on Multi-Modal Feature Fusion. *Mathematics* **2021**, *9*, 2085. <https://doi.org/10.3390/math9172085>

Academic Editor: Amir Mosavi

Received: 1 July 2021

Accepted: 26 August 2021

Published: 29 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Targets pose estimation refers to the establishment of the transformation relationship between the world coordinate system and the camera coordinate system, which can accurately estimate the 3D position and pose of the object. In this way, more flexible space robot grasping and manipulation can be realized. The traditional pose estimation method is mainly based on a local feature operator [1]. This method is only suitable for objects with texture information, but it cannot do anything for weak texture objects. Aiming at the problem of pose estimation of weak texture objects, researchers propose a two-dimensional plane detection method based on edge extraction, but the method relies heavily on key point matching, target detection, and a known three-dimensional model [2].

With the development of deep learning technology in the field of two-dimensional images, the method of remote pose estimating from images directly based on CNN architecture has been proposed multiple times [3]. For example, the BB8 algorithm [4] first segments the pixels of the object from the two-dimensional image; then, it uses another CNN network to regress the eight corners of the object and the three-dimensional bounding box and finally optimizes the 6D pose of the object through the PNP algorithm. The SSD-6D algorithm [5] generates the position and rotation parameters on account of the target detection algorithm SSD. First, a convolutional neural network is used to estimate the initial 6D pose of the target. Finally, the ICP algorithm is used for iterative correction. The common point of these methods is to improve the existing target detection algorithm

and estimate the 6D pose of the object in an RGB image by calculating the offset of the object and bounding box. However, these algorithms extract information from RGB images separately. The lack of depth information seriously limits the application of these algorithms in complex environments [6].

With the popularity of 3D cameras, the point cloud data based on depth information can not only express the basic shape features of the object but also contain the texture features of the object [7]. Therefore, a pose estimation method based on RGB-D is proposed that can combine the local features of the image with the 3D point cloud to build a template library with 3D information, which can effectively improve the robustness of the algorithm. In order to avoid the interference of illumination, environment, and other factors, the representative algorithm Linemod [8] based on RGB-D extracts image boundary features and surface normal features of depth map as templates, which greatly improves the stability of the algorithm. After a binary coding, these features can be achieved by a parallel template search, which improves the processing speed of the algorithm. The PoseCNN algorithm [9] is the first end-to-end 6D pose estimation network, which directly extracts features and then completes the output by adding three branch networks of image segmentation, position, and pose. Finally, the ICP algorithm iteratively optimizes the pose. However, the defect of this algorithm is that it needs to know the 3D model of the target, and in order to improve the accuracy of pose estimation, many algorithms need to add a post-processing operation, which is bound to fail to meet the real-time requirements of space robot grasping [10].

Accurate remote estimation of 6D pose is the key factor for the space robot to grasp successfully. It can provide the 3D translation coordinates and 3D rotation vectors of the object in the camera coordinate system. Traditional methods of pose estimation have disadvantages such as complex calculation, low accuracy, insufficient robustness to occlusion, and poor generalization. Therefore, in such complex scenes as space robot grasping, the method based on deep learning has unparalleled advantages. In addition, compared with the RGB-based pose estimation method, the RGB-D method contains more modal information and is not limited to the target texture information. To solve the above problems, this paper, based on the instance segmentation algorithm, obtains three-dimensional data of the object from the image, which extracts the semantic features and point cloud features of two heterogeneous kinds of data and fuses multi-modal features at the pixel level. In addition, the post-processing module for pose fine tuning is integrated into the deep learning framework to further improve the accuracy of the algorithm without increasing computation too much. Experiments on a common dataset show that the algorithm has high accuracy and significantly improves the real-time performance of pose estimation.

The main contributions of the proposed algorithm framework are as follows. Firstly, a multi-modal feature fusion method is proposed, which integrates the two-dimensional information contained in each pixel into the corresponding three-dimensional point cloud of the model, effectively preserving the spatial structure information of the model. Secondly, the network structure based on posture estimation is integrated into a refine module and integrated into the end-to-end neural network to avoid the dependence on ICP and other post-processing methods, which effectively improves the processing speed of the model.

The remainder of the study is arranged as follows. In Section 2, we introduce the related work of pose estimation. Section 3 focuses on the construction of a new neural network that can estimate the pose of the object based on feature fusion. In Section 4, the performance of the proposed neural network is discussed according to the experiment result. Finally, the conclusion that the new method has faster processing speed and better generalizability is given, and future research is described.

2. Related Work

The traditional architecture of a pose estimation network is to extract local or template features first and then classify the extracted features to realize the recognition of different targets [11]. However, the performance of the algorithm is seriously constrained by the artificially designed features and fixed matching process. Therefore, the pose estimation

methods based on CNN architecture appear repeatedly, such as PoseNet and Pynet [12]. However, due to the lack of depth information, pose estimation is still a difficult problem, so the method of remote pose estimation based on 3D data has become a trend.

The traditional method is based on RGB-D to extract 3D features, which integrates the depth information as an additional channel of the image into the CNN-based architecture [13]. Recent algorithms transform depth information into 3D data, such as voxels and point clouds, and then estimate the target pose from the 3D data. For example, Song et al. [14] estimated the target pose based on the voxelized input of a 3D convolution network and achieved a good result. However, these methods still need expensive post-processing steps to make full use of 3D data, resulting in low real-time performance of the algorithm.

The two-dimensional image can only reflect the projection information of the target at a certain angle in the three-dimensional space, so the representation of the related attributes of the object is not accurate, and the three-dimensional data can accurately represent the real shape and geometric structure of the target [15]. Previous methods are based on CNNs, which treat depth information as an additional image channel, but these methods ignore the inherent three-dimensional structure of depth information. Therefore, this paper chooses to process 3D data directly. The representation methods of 3D data mainly include point clouds, voxels, meshes, and multi-view images [16]. A point cloud is composed by a number of three-dimensional discrete points. Voxels are the individual cells of a cubical grid, which are combined into objects in a standard way. Meshes mean that polygon surfaces are formed by the relationship between point pairs. Multi-view images are several images from different viewpoints of the same scene to form a complete three-dimensional model. As a result of the disorder and unstructured characteristics of a 3D point cloud, the robustness of the method based on a point cloud is higher, but the feature extraction and characterization of a point cloud are more difficult.

At present, many methods choose to voxelize the point cloud data and then extract features based on deep learning, but these methods have the problem of information loss, which is not suitable for real-time requirements [17]. The multi-view image method to predict the pose of the target is based on two-dimensional image object detection to learn from multiple perspectives. However, this method is limited to the number of views, and it needs a scene without occlusion, so it is not suitable for complex scenes. Similarly, the grid-based method transforms the point cloud into regular 3D data for processing, which inevitably has defects [18].

3. Pose Estimation Based on Feature Fusion

The objects to be located by our algorithm are defined by RGB-D color data and 3D point clouds which segmented their associated 3D bounding boxes from images. When presented with a new RGB-D image, our goal is to recognize these objects and determine their 3D positions and orientations as needed in a space robot grasping task. To solve these problems, according to the known camera internal function, the depth information obtained from segmentation can be converted into 3D point cloud data, and then the point cloud data of the model can be directly processed. According to the characteristics of point cloud data, a symmetric function is proposed to extract the point cloud information, and the corresponding position and category of each point can be obtained directly, as shown in Formula (1).

$$f(\{x_1, \dots, x_n\}) = \max(h(x_{P(1)}, x_{P(2)}, \dots, x_{P(n)}), P \in S(n) \quad (1)$$

where $S(n)$ is the symmetric group of all permutations of n points, and h is a multi-layer perception. f is the final function, $x_{P(n)}$ means that the point cloud data are processed by a multi-layer perceptron. The disorder of the point cloud will not affect the arrangement of final function f after being treated by a symmetric function. This function is to extract the multi-dimensional features of a 3D point cloud model through a neural network and then take the maximum value of each dimension feature, so as to effectively avoid the disorder

problem. At the same time, in order to improve the accuracy of point cloud processing, a three-dimensional spatial transform network (STN) is added for the invariance problem. A rotation matrix that is most conducive to network processing can be learned from the pose information of point cloud, as shown in Formula (2).

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \phi_x) k(y_i^s - n; \phi_y) \quad (2)$$

where H, W denotes the height and weight of the image, and the network is the process of converting the original coordinates x_i^s, y_i^s to the target image. k is the bilinear interpolation function, ϕ_x, ϕ_y are the parameters of the bilinear interpolation function, and U_{nm}^c is the color of (n, m) in the image channel C . After adjusting the point cloud in space, the point cloud can be rotated to a more appropriate angle for processing. However, for the scene segmentation problem, the relationship between local point pairs is not learned, and the uneven density in the feature extraction process also restricts the effect of the network model.

Aiming at the problem of missing local features, this paper establishes hierarchical grouping of points based on a hierarchical point cloud feature learning method to extract abstract local features. First, the local part of the 3D model is randomly sampled by the farthest point sampling (FPS), which is sampled in a range. Then, in the aggregation layer, the sampled points are taken as the center, and the points in the aggregation sphere form a subset of the point cloud. Finally, the global features of the point cloud subset are extracted based on the PointNet network. After many iterations of such operations, the number of points remaining in the point cloud becomes less, and the local feature extracted from the point cloud contains more information. The overall process of the hierarchical point cloud feature learning method is shown in Figure 1.

Figure 1. Hierarchical point cloud feature learning method. Point cloud subsets are generated in the aggregation layer through random sampling, and abstract local features can be extracted after multiple iterations.

Finally, we need to fuse the point cloud information and the 2D information of the image through the ResNet network, which processes the point cloud data through PointNet and fuses the two sub-networks to get the 3D boundary box. In order to avoid the influence of different sizes of the target, the concatenation technique in the image target detection algorithm is used for reference, and the point cloud features and 2D features are spliced together. Finally, the offset value and confidence of each point compared with the center of the bounding box are obtained. The loss function of the network is shown in Formula (3).

$$L = \frac{1}{N} \sum_i \text{smoothL1}(x_{offset}^{i*}, x_{offset}^i) + L_{score} + L_{stn} \quad (3)$$

Here, x_{offset}^{i*} is the real corner position, x_{offset}^i is the predicted corner position, L_{stn} is the loss of spatial transformation regularization, which can strengthen the orthogonality of the spatial transformation matrix, and L_{score} is the loss of fractional function, which can make the network learn the offset from the point cloud to the nearest target box [19].

It is expected that the algorithm can be robust to noise and changing illumination in complex environments and meet the real-time requirements. This paper proposes an end-to-end deep learning method, which estimates the 6D pose of objects based on RGB-D.

The core is to fuse image features and point cloud features at the pixel level. This fusion method can clearly explain the local appearance and geometric information, and it can effectively improve the accuracy of pose estimation. At the same time, integrating the pose fine-tuning module into the end-to-end learning framework can not only improve the performance of the model but also meet the requirements of real-time tasks.

First, the segmented objects are obtained based on the instance segmentation algorithm PSPNet [20]. At the same time, the depth pixels of the mask are converted into 3D point clouds containing spatial structure information. Then, for each segmented object, the 2D features of the image are extracted based on the CNN network, and the geometric feature vector of the 3D point cloud is extracted based on the PointNet network. Such separate processing can effectively preserve the data information. Finally, multi-modal features are fused at the pixel level to estimate the 6D pose of the target. The refine module based on this network framework can iterate repeatedly to optimize the pose estimation results so as to improve the accuracy of the algorithm.

3.1. Feature Extraction

RGB and depth information are located in different spaces. How to extract features from heterogeneous data sources and fuse them properly is the key technology to be studied in this paper. Previous algorithms either use the high-cost 3D convolution algorithm or only use the depth information as an additional channel outside the RGB channel, resulting in a low accuracy of pose estimation. In this paper, RGB information and depth information are processed separately. First, the input image is segmented to obtain the target block diagram and mask image. At the same time, the depth pixels of the mask are converted into three-dimensional point clouds containing spatial structure information. Then, for each segmented object, 2D features of the image are extracted based on a CNN network, and geometric feature vectors of the 3D point cloud are extracted based on the PointNet network. The whole process is shown in Figure 2.

Figure 2. Framework of feature extraction algorithm. Objects are segmented from the instances in the images, and then 2D semantic features and 3D point cloud features are extracted respectively based on deep learning methods.

As can be seen from the above figure, the first step of the algorithm is to segment the region of interest in the image, and then, we need to extract information from RGB and depth channels and fuse them. Although the structures of these two information sources are similar, they reside in different spaces. The RGB-D image was a collection of pixels, each with RGB, and the mask for the segmented object of interest was a subset of these pixels. In this case, there are as many 3D points as RGB pixels, so there is no discrepancy in size. In this paper, we choose to process the two data sources separately so as to retain the internal structure information of the data source. The CNN network for extracting two-dimensional image features is pyramid scene parsing. At the same time, the up-sampling structure of the network should make the size of two-dimensional features and three-dimensional point cloud features consistent, which is convenient for pixel-by-pixel fusion. The main body of PSPNet (Pyramid Scene Parsing Network) uses ResNet18 as the basic network,

which can extract information of various scales in different regions, greatly improving the scene understanding ability of the algorithm, as shown in Figure 3.

Figure 3. Two-dimensional feature extraction based on PSPNet, which integrates four features of different pyramid scales, and then, these features are upsampled to obtain feature maps, which are finally concatenate with the original feature map.

As can be seen from the above figure, the core of the two-dimensional feature extraction algorithm based on PSPNet is Spatial Pyramid Pooling (SPP), which can understand the scene more accurately through the combination of local and global features. Point cloud data processing can be accomplished in the PointCNN algorithm, mainly using a symmetric function to solve the problem of translation invariance in unordered point cloud processing.

3.2. Multi-Modal Data Fusion

After the feature extraction of RGB and point cloud using different networks, in order to ensure the maximum retention of target information, the features of different levels in the network are extracted and fused, and finally, they are spliced into a pixel-level fusion feature, as shown in Figure 4.

Figure 4. Schematic diagram of the feature fusion algorithm. Three red boxes can be obtained to represent the feature fusion of the three stages, and the final global feature can be obtained by combining these three features.

In order to keep consistent with the dimension of the point cloud, the image features should adjust the dimension. Then, the two layers of features in the network are concatenated on the basis of pixels to generate fusion features. After average pooling of fusion feature 2, fusion feature 3 is repeated to ensure consistency with fusion feature 1 and fusion feature 2. Finally, a multi-layer fusion feature is obtained to ensure that the final feature contains multiple size information. Then, the preliminary prediction of 6D pose

can be realized based on this fusion feature. At the same time, three groups of parameters based on the fusion features can be obtained, including a rotation parameter, translation parameter, and the degree of confidence. In this pose prediction model, the loss function is the least square method between the real points in three-dimensional space and the corresponding points of the model after pose regression, as shown in Formula (4).

$$L_i(\bar{q}_i, q) = \frac{1}{2m} \sum_{x \in M} \| (R(\bar{q}_i)x + \bar{T} - (R(q)x + T) \|^2 \quad (4)$$

Here, $\bar{q}_i, q$ denotes the real point and the predicted point, m is the number of points in the point set, $R(q)x, R(\bar{q}_i)x$ is a matrix of the predicted rotation parameters and real rotation parameters, and $T, \bar{T}$ is a vector of predicted translation parameters and real translation parameters. However, this loss function is not suitable for centrosymmetric or axisymmetric objects, because when the loss function takes a value, it will correspond to many situations, which is not conducive to the stability of the network. Therefore, the loss function of a symmetric target is modified to Formula (5).

$$L_i(\bar{q}_i, q) = \frac{1}{2m} \sum_{x_1 \in M} \min_{x_2 \in M} \| (R(\bar{q}_i)x_1 + \bar{T} - (R(q)x_2 + T) \|^2 \quad (5)$$

The total loss function of the whole network combined with confidence is Formula (6).

$$L = \frac{1}{N} \sum_{i=1}^N (c_i L_i(\bar{q}_i, q) - \alpha \log(c_i)) \quad (6)$$

In the formula, i denotes the current point, N is the sum of the number of points, c_i is the confidence, and α is the super parameter. Finally, the final 6D pose parameters are determined according to the maximum confidence based on an unsupervised method.

3.3. Pose Refine Network

After the initial pose is obtained, in order to improve the accuracy of space robot grasping, the final pose parameters need to be optimized to realize the fine matching of 6D pose. Based on the idea of the ICP algorithm, this paper proposes a precise registration method integrated into the whole network, which can not only improve the accuracy of the algorithm, but also ensure the real-time performance of the model. This module first needs to process the target point cloud according to the predicted pose to achieve the network input of iterative fine matching, as shown in Figure 5.

Figure 5. Pose estimation of a fine matching network. The input point cloud and the real pose are transformed according to the inverse matrix of the feature fusion network to get the new position and corresponding pose of the target.

It can be seen from the figure that the predicted pose has been obtained from the target point cloud, but the predicted pose must be deviated from the real pose. Therefore, the inverse matrix H_e^{-1} can be obtained by inverting the predicted pose. Then, the new

position of the target point cloud can be obtained by multiplying the target point cloud by the inverse matrix, and the real pose corresponding to the new position of the target point cloud can be obtained by multiplying the real pose by the inverse matrix. Based on this new location, a new point cloud can be obtained, and the geometric features of the point cloud can be updated, and then, the new fusion features can be obtained, as shown in Figure 6.

Figure 6. Feature fusion in the refine model. The fine-tuning network takes the result of the last iteration as the input point cloud, and the main network is mostly the same as the previous matching network.

As can be seen from the figure, taking the last iteration result as the input point cloud, the RGB feature and the geometric structure feature of the point cloud can be extracted again, and then the fusion feature is obtained to predict the 6D pose parameters in refine mode. The difference between this part of network and the main network is that there is no output of confidence, but the category parameters are still included in the pose parameters. This is the first iterative process of fine matching. According to the output pose, the new position can be obtained by inversion, and so on until convergence. From the experiments presented later in this paper, it can be seen from that this iterative method can effectively improve the accuracy of 6D pose estimation, and the increased amount of calculation is very low.

3.4. Network Model Optimization

This section mainly studies the problems in the process of network model training, as well as the optimization for the network, so as to ensure that the algorithm achieves the desired effect. First of all, the point cloud in the YCB-Video dataset [21] focuses more on the boundary information of the object, so the 3D model can well describe the shape and key point information of the object, which is very helpful for the shape extraction of the object. However, this will lead to an uneven distribution of the point cloud and the low recognition of the model, which will affect the training results. To solve this problem, this paper expands the number of point cloud samples from 3000 to 10,000, and it adjusts the distribution of the template point cloud to make it evenly distributed [22], as shown in Figure 7.

In the upper row of the figure, we can see that the distribution of the original 3D model point cloud is uneven, which leads to the lack of surface information in many parts. The lower row adjusts the distribution of the point cloud, and the 3D model can take the contour information and texture information of the object into account, so as to effectively improve the recognition accuracy of the algorithm.

Figure 7. Point cloud of original dataset and adjusted image. The upper row is the model provided by the dataset, which is not highly identifiable. After adjusting the distribution of the template point cloud, a more accurate model can be obtained.

4. Experiment Results and Discussion

The experimental workstation is configured with i7 CPU, 16GB memory, and an NVIDIA GTX 1080Ti graphics card. In order to study the effect of the pose estimation network in the space robot grasping application, this paper chooses to evaluate the algorithm in the YCB-Video dataset. It contains 133,827 independent scenes, depicts various object instances, and has complete annotation information. This dataset is a typical dataset with wide coverage and difficult synthesis in the field of pose estimation, which is very close to an actual scene of space robot grasping. The dataset contains real data and synthetic data, including asymmetric objects, centrosymmetric objects, and axisymmetric objects, and it contains a variety of background surfaces.

Each example contains an RGB image, depth image, and instance segmentation annotation image. After obtaining the 6D pose estimation results, we need to use a quantitative evaluation index to evaluate the algorithm, so as to intuitively compare the algorithm effect. Since 6D pose estimation is a regression problem, the error metrics used in classification scenarios, such as accuracy and recall, are not applicable. For 6D pose estimation, it is necessary to define the error measure as the standard to evaluate the performance of the model. In this paper, the average distance metric (ADD) is selected as the error measurement standard of 6D pose estimation. It is a measurement method with invariable ambiguity, which represents the two-norm average of the three-dimensional distance between the predicted 6D pose and the real 6D pose, as shown in Formula (7).

$$ADD = \frac{1}{m} \sum_{x \in M} \|K_c(\hat{R}x + \hat{t}) - K_c(Rx + t)\|^2 \quad (7)$$

According to the formula, ADD is to calculate the average distance from each 3D model point of estimated pose $[\hat{R}, \hat{t}]$ to the nearest domain point of real pose $[R, t]$. K_c represents the degree of confidence of the pose estimation. In order to avoid deviation, the real pose and the estimated pose can be projected into the camera coordinate system, and then, the average distance of the symmetrical object can be calculated. In this paper, ADD is used to integrate symmetric and asymmetric objects into the overall evaluation.

By comparing the experimental results of two models in the YCB-Video dataset, the effectiveness of the proposed pose estimation algorithm is demonstrated. After the initial pose estimation of the whole network and the iterative refining of the refine network, we can see how the pose estimation process gradually calibrates the pose of the 3D model, as shown in Figure 8.

Figure 8. Initial estimated pose and fine adjusted pose of the example input point cloud. The blue point cloud represents the real pose of the target, and the red point cloud represents the estimated pose. The upper row objects are the result of the training of the previous modulation network that still has a certain error with the real value. The lower row is the result of fine-tuning network training, which obviously improves the accuracy of estimation effectively.

The blue point cloud represents the real pose of the target, and the red point cloud represents the recognized pose. It can be seen that there are still some errors between the initial pose and the real pose after the network training. However, through the repeated iteration of the fine-tuning network, it is obvious that most of the final point cloud models come close to the real point cloud. Therefore, the effectiveness of the pose estimation algorithm proposed in this paper can be repetitious and proved. In Table 1, the algorithm is compared with the most advanced pose estimation methods, including PoseCNN based on RGB and Densefusion based on RGB-D. In order to maintain consistency, all algorithms are configured with the same training set and test set. The comparison of ADD values in the quantitative evaluation of a 6D pose of the YCB-Video dataset is shown in Table 1.

Table 1. Comparison of ADD values of 6D pose estimation in the YCB-Video dataset.

Methods	BB8 [15]	PoseCNN [9]	Densefusion [7]	Ours
004_sugar_box	54.6	80.6	85.4	80.5
005_tomato_soup_can	44.3	82.4	73.6	75.5
007_tuna_fish_can	45.5	76.5	80.5	83.1
011_banana	47.4	78.3	82.3	84.3
024_bowl	46.6	85.1	73.2	80.4
025_mug	50.6	77.4	78.5	76.5
035_power_drill	48.2	85.6	82.6	84.2
037_scissors	57.8	89.3	80.7	82.3
MEAN	49.4	81.7	83.2	82.5

It can be seen from the table that the effect of PoseCNN based on RGB is not as good as that of Densefusion based on RGB-D, which proves that depth data are indispensable information in the process of pose estimation. The effect of the proposed method is similar to that of Densefusion, indicating that the multi-mode data fusion module plays a major role in improving the accuracy. Through the actual scene test, it can be found that the calculation speed of the proposed algorithm can reach about 24 pieces per second, while the accuracy can still remain above 80%. More importantly, compared with the existing methods, it can better solve the problem of grasping when the objects are mutually occluded. At the same time, in order to prove the advantages of the proposed algorithm in the space robot grasping, the three algorithm models were deployed on the actual platform respectively, and the running time of each step was counted, as shown in Table 2.

Table 2. Average running time of three 6D pose estimation algorithms.

PoseCNN [9] (ms)	Semantic Segmentation 30	Pose Estimation 170	ICP 850	Total Time 1050
Densefusion [7] (ms)	Instance Segmentation 56.4	Pose Estimation 17	ICP 560	Total Time 633.4
Ours (ms)	Instance Segmentation 36	Pose Estimation 15.6	Refine 10	Total Time 61.6

As can be seen from Table 2, compared with the RGB-based method PoseCNN, the proposed algorithm in this paper has a great improvement in the real-time performance of remote pose estimation. The previous two methods spend most of their effort on the post-processing, while the new algorithm integrates the fine-tuning module into the whole network, which improves the performance without much increase in computation. Although the proposed pose estimation algorithm may not be the most outstanding in average distance, the real-time performance of the model is of great value for space robot grasping.

Traditional 6D pose estimation requires either 3D convolution or a 3D model for ICP registration, which requires a great amount of computation. Generally, the speed of these algorithms can only process a part in 20 s, which is obviously not in line with production requirements. On the other hand, it is difficult for an algorithm calculating an object pose based on RGB images to deal with the complicated stacking of parts. We believe that the depth map contains the spatial pose information of the object, and the RGB map contains the category and boundary information of the object. Therefore, the feature information of these two modes can be extracted separately and fused to maximize the use of depth information. At the same time, the depth map is transformed into the point cloud map, which greatly reduces the calculation and does not reduce the running speed. The proposed algorithm can achieve a 6D pose quickly and accurately when dealing with the accumulation of complex parts. More importantly, it is robust enough for complex industrial environments, which has important practical significance for industrial production.

5. Conclusions

Aiming at the problems of low accuracy and slow speed of current pose estimation algorithms, an end-to-end pose estimation algorithm based on multi-mode feature fusion is proposed and verified in this paper. First, the proposed algorithm segments the RGB information and depth information of the target at the same time, and it effectively preserves the spatial information of the data by extracting semantic features and point cloud features of the two heterogeneous data respectively. Then, multi-modal features are fused at the pixel level to obtain a new global feature, which is used to complete the task of remote pose estimation. At the same time, the post-processing module of pose fine tuning is integrated into the end-to-end framework, so that the algorithm can improve the accuracy without adding too much computation.

Experimental results show that the pose estimation algorithm based on RGB-D input can realize the real-time detection of a 6D pose while maintaining sufficient detection accuracy. This algorithm is different from previous methods that treat depth maps as a channel of RGB maps to process images. Instead, RGB and point cloud features are extracted respectively; then, image features with the strongest correlation are screened out for stitching in one dimension. In the future work, I will continue to optimize the accuracy and calculation speed of the algorithm so that it can be applied to the actual production in a more complicated environment.

Author Contributions: Conceptualization, J.H.; Data curation, J.H. and L.Z.; Formal analysis, J.H.; Funding acquisition, L.Z.; Investigation, J.H. and T.H.; Methodology, J.H. and T.H.; Supervision, L.Z. and G.Y.; Validation, L.Z. and G.Y.; Writing original draft, J.H.; Writing review and editing, J.H. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support from the National Natural Science Foundation of China (grant no. 51975425).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Please contact Jiang Hua (huajiang@wust.edu.cn).

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

References

1. He, Y.; Sun, W.; Huang, H.; Liu, J.; Fan, H.; Sun, J. PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–19 June 2020; IEEE: Seattle, WA, USA; pp. 11629–11638.
2. Liu, F.; Fang, P.; Yao, Z.; Fan, R.; Pan, Z.; Sheng, W.; Yang, H. Recovering 6D Object Pose from RGB Indoor Image Based on Two-Stage Detection Network with Multi-Task Loss. *Neurocomputing* **2019**, *337*, 15–23. [\[CrossRef\]](#)
3. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Long Beach, CA, USA; pp. 4556–4565.
4. Borghi, G.; Fabbri, M.; Vezzani, R.; Calderara, S.; Cucchiara, R. Face-from-Depth for Head Pose Estimation on Depth Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 596–609. [\[CrossRef\]](#) [\[PubMed\]](#)
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 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Venice, Italy; pp. 1530–1538.
6. Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Brucker, M.; Triebel, R. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. *Int. J. Comput. Vis.* **2019**, *128*, 714–729. [\[CrossRef\]](#)
7. 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. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
8. Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Triebel, R. Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection. *Int. J. Comput. Vis.* **2020**, *128*, 714–729. [\[CrossRef\]](#)
9. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In Proceedings of the Robotics: Science and Systems XIV. Robotics: Science and Systems Foundation, Freiburg im Breisgau, Germany, 26 June 2018.
10. Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T.S.; Zhang, L. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Seattle, WA, USA; pp. 5385–5394.
11. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.-E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Gao, Q.; Liu, J.; Ju, Z.; Zhang, X. Dual-Hand Detection for Human-Robot Interaction by a Parallel Network Based on Hand Detection and Body Pose Estimation. *IEEE Trans. Ind. Electron.* **2019**. [\[CrossRef\]](#)
13. Ke, L.; Chang, M.-C.; Qi, H.; Lyu, S. Multi-Scale Structure-Aware Network for Human Pose Estimation. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Germany, 2018; Volume 11206, pp. 731–746. ISBN 9783030012151.
14. Haase, D.; Amthor, M. Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets. *arXiv* **2020**, arXiv:2003.13549.
15. Du, G.; Wang, K.; Lian, S.; Zhao, K. Vision-Based Robotic Grasping from Object Localization, Object Pose Estimation to Grasp Estimation for Parallel Grippers: A Review. *Artif. Intell. Rev.* **2020**. [\[CrossRef\]](#)
16. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *arXiv* **2019**, arXiv:1912.13192.
17. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7156–7165.

18. Fieraru, M.; Khoreva, A.; Pishchulin, L.; Schiele, B. Learning to Refine Human Pose Estimation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 318–31809.
19. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995.
20. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21 July–26 July 2017; pp. 2881–2890.
21. Liu, J.; Ding, H.; Shahroudy, A.; Duan, L.Y.; Jiang, X.; Wang, G.; Kot, A.C. Feature Boosting Network For 3D Pose Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 494–501. [[CrossRef](#)] [[PubMed](#)]
22. Ni, P.; Zhang, W.; Zhu, X.; Cao, Q. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3619–3625.