FFB6D: A Full Flow Bidirectional Fusion Network for 6D Pose Estimation

Yisheng He[1]   Haibin Huang[3]   Haoqiang Fan[2]   Qifeng Chen[1]   Jian Sun[2]
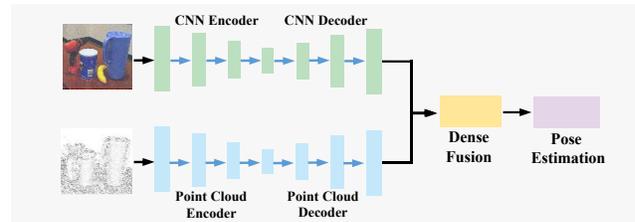
[1]Hong Kong University of Science and Technology   [2] Megvii Technology   [3]Kuaishou Technology

Abstract

*In this work, we present FFB6D, a **F**ull **F**low **B**idirectional fusion network designed for **6D** pose estimation from a single RGBD image. Our key insight is that appearance information in the RGB image and geometry information from the depth image are two complementary data sources, and it still remains unknown how to fully leverage them. Towards this end, we propose FFB6D, which learns to combine appearance and geometry information for representation learning as well as output representation selection. Specifically, at the representation learning stage, we build **bidirectional** fusion modules in the **full flow** of the two networks, where fusion is applied to each encoding and decoding layer. In this way, the two networks can leverage local and global complementary information from the other one to obtain better representations. Moreover, at the output representation stage, we designed a simple but effective 3D keypoints selection algorithm considering the texture and geometry information of objects, which simplifies keypoint localization for precise pose estimation. Experimental results show that our method outperforms the state-of-the-art by large margins on several benchmarks. Code and video are available at* `https://github.com/ethnhe/FFB6D.git`.

(a) **The DenseFusion [65] Network**. The two networks extract features from different modalities of data separately without any communication, util the final layers of the encoding-decoding architecture.
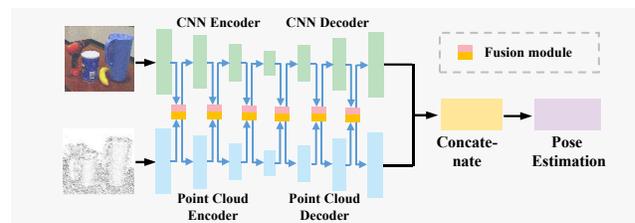
(b) **The Proposed Full Flow Bidirectional Fusion Network**. Bidirectional fusion modules are added as bridges for information communication in the full flow of the two networks, where fusion is applied on each encoding and decoding layers. Local and global supplementary information from each other is shared between the two networks for better appearance and geometry representation learning, which is crucial for 6D pose estimation.

Figure 1: **Network Comparison**

1. Introduction

6D Pose Estimation is an important component in lots of real-world applications, such as augmented reality [40], autonomous driving [13, 8, 69] and robotic grasping [9, 62, 17]. It has been proven a challenging problem due to sensor noise, varying lighting, and occlusion of scenes. Recently, the dramatic growth of deep learning techniques motivates several works to tackle this problem using convolution neural networks (CNNs) on RGB images [68, 46, 70, 34]. However, the loss of geometry information caused by perspective projection limits the performance of these approaches in challenging scenarios, such as poor lighting conditions, low-contrast scenes, and textureless objects. The recent advent of inexpensive RGBD sensors provides extra depth information to ease the problem [5, 19, 23, 22] and also leads

to an interesting research question: How to fully leverage the two data modalities effectively for better 6D pose estimation?

One line of existing works [68, 32] leverage the advantage of the two data sources within cascaded designs. These works first estimate an initial pose from RGB images and then refine it on point clouds using either the Iterative Closest Point (ICP) algorithm or multi-view hypothesis verification. Such refinement procedures are time-consuming and can not be optimized with the pose from RGB images end-to-end. On the other hand, works like [48, 69] apply a point cloud network (PCN) and a CNN to extract dense features from the cropped RGB image and point cloud respectively and the extracted dense features are then concatenated for pose estimation [65]. Recently, DenseFusion [65] proposed a better fusion strategy which replaced the naive concatena-

tion operation with a dense fusion module, shown in Figure 1(a), and delivered improved performance. However, both feature concatenation and DenseFusion suffers from performance degeneration due to the separation of CNN and PCN in several scenarios, including objects with similar appearance or with reflective surfaces. Such cases are challenging either for the isolated CNN or the PCN feature extraction.

In this work, we propose a *full flow bidirectional* fusion network that perform fusion on each encoding and decoding layers for representation learning from the RGBD image, shown in Figure 1(b). Our key insight is that appearance information in RGB and geometry information in point cloud can serve as complementary information during their feature extraction procedure. Specifically, during the CNN encoding-decoding procedure, it's hard for CNN to learn a distinctive representation for similar objects from the RGB image, which, however, is obvious in the PCN's view. On the other hand, the miss of depth caused by reflective surfaces of objects challenges the point cloud only geometry reasoning. Whereas, these objects are visible by CNN from RGB images. Hence, it's necessary to get through the two separated feature extraction branches in the early encoding-decoding stages and the proposed *full flow bidirectional* fusion mechanism bridges this information gap.

We further leverage the learned rich appearance and geometry representation for the pose estimation stage. We follow the pipeline proposed in PVN3D [17], which opens up new opportunities for the 3D keypoint based 6D pose estimation. However, it only considers the distance between keypoints for 3D keypoint selection. Some selected keypoints might appear in non-salient regions like smooth surfaces without distinctive texture, making it hard to locate. Instead, we take both the object texture and geometry information into account and propose the SIFT-FPS algorithm for automatic 3D keypoint selections. Salient keypoints filtered in this way are easier for the network to locate and the pose estimation performance is facilitated.

To fully evaluate our method, we conduct experiments on three popular benchmark datasets, the YCB-Video, LineMOD, and Occlusion LineMOD datasets. Experimental results show that the proposed approach without any time-consuming post-refinement procedure outperforms the state-of-the-art by a large margin.

To summarize, the main contributions of this work are:

- A novel full flow bidirectional fusion network for representation learning from a single scene RGBD image, which can be generalized to more applications, such as 3D object detection.

- A simple but effective 3D keypoint selection algorithm that leverages texture and geometry information of object models.

- State-of-the-art 6D pose estimation performance on the YCB-Video, LineMOD, and Occlusion LineMOD datasets.

- In-depth analysis to understand various design choices of the system.

2. Related Work

### 2.1. Pose Estimation with RGB Data

This line of works can be divided into three classes, holistic approaches, dense correspondence exploring, and 2D-keypoint-based. Holistic approaches [27, 15, 18, 68, 33, 63, 58, 60, 44, 54] directly output pose parameters from RGB images. The non-linearity of rotation space limit the generalization of these approaches. Instead, dense correspondence approaches [36, 59, 14, 2, 41, 28, 12, 34, 66, 6, 6, 21, 4] find the correspondence between image pixels and mesh vertexes and recover poses within Perspective-n-Point (PnP) manners. Though robust to occlusion, the large output space limits the prediction accuracy. Instead, 2D-keypoint-based [53, 52, 42, 29, 43, 46, 72, 38] detect 2D keypoints of objects to build the 2D-3D correspondence for pose estimation. However, the loss of geometry information due to perspective projections limit the performance of these RGB only methods.

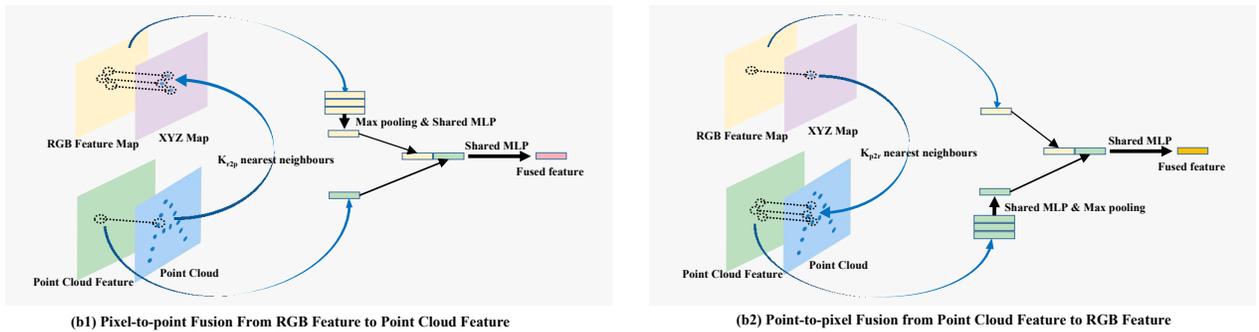### 2.2. Pose Estimation with Point Clouds

The development of depth sensors and point cloud representation learning techniques [49, 50] motivates several point clouds only approaches. These approaches either utilize 3D ConvNets [56, 57] or point cloud network [74, 47, 24] for feature extraction and 3D bounding box prediction. However, sparsity and non-texture of point cloud limit the performance of these approaches. Besides, objects with reflective surfaces can not be captured by depth sensors. Therefore, taking RGB images into account is necessary.

2.3. Pose Estimation with RGB-D Data

Traditional methods utilize hand-coded [19, 20, 51] templates or features optimized by surrogate objectives [2, 61, 67, 28] from RGBD data and perform correspondence grouping and hypothesis verification. Recent data-driven approaches propose initial pose from RGB images and refine it with point cloud using ICP [68] or MCN [32] algorithms. However, they are time-consuming and are not end-to-end optimizable. Instead, works like [31, 35] add appearance information from CNN on RGB images as complementary information for geometry reasoning on bird-eye-view (BEV) images of point clouds. But they neglect the help of geometry information for RGB representation learning, the BEV ignore the pitch and roll of object pose, and the regular 2D CNN is not good at contiguous geometry reasoning either. Instead, [17, 69, 48, 65, 73, 64] extract
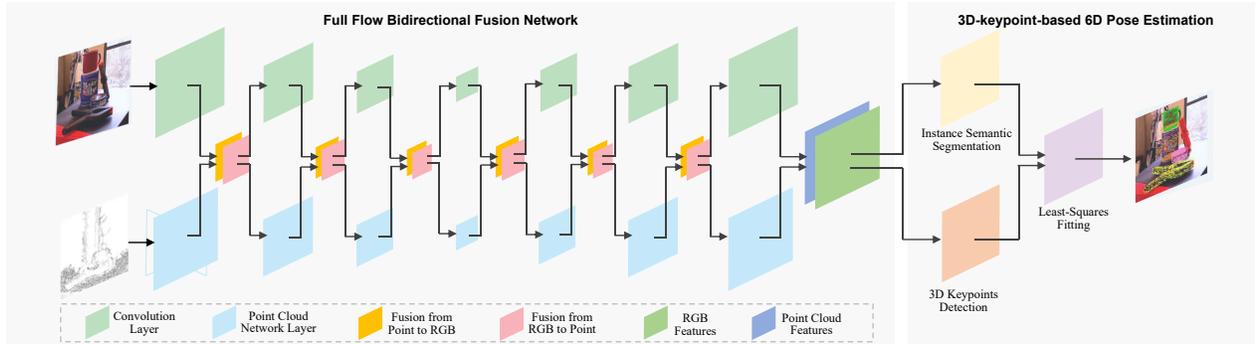
(a) **The pipeline of FFB6D**. A CNN and a point cloud network is utilized for representation learning of RGB image and point cloud respectively. In flow of the two networks, bidirectional fusion modules are added as communicate bridges. The extracted per-point features are then fed into an instance semantic segmentation and a 3D keypoint voting modules to obtain per-object 3D keypoints. Finally, the pose is recovered within a least-squares fitting algorithm.

(b1) Pixel-to-point Fusion From RGB Feature to Point Cloud Feature

(b2) Point-to-pixel Fusion from Point Cloud Feature to RGB Feature

(b) **Dense bidirectional fusion modules**. (1) The pixel-to-point fusion modules fuse RGB features to point cloud features. For each point, we find its K_{r2p} nearest neighbors in the XYZ map and gather their corresponding appearance features from the RGB feature map. These features are then processed by max pooling and a shared MLP to obtain the most significant appearance features. Finally, a shared MLP fuses the concatenation of the appearance and geometry features to obtain the fused point features. (2) The point-to-pixel fusion modules similarly obtain fused pixel features as the pixel-to-point fusion.

Figure 2: **Overview of FFB6D.**

features from RGB images and point clouds using CNN and point cloud network individually and then fused them for pose estimation. Such approaches are more effective and efficient. Nevertheless, since the appearance and geometry features are extracted separately, the two networks are not able to communicate and share information, and thus limit the expression ability of the learned representation. In this work, we add bidirectional fusion modules in the full network flow as communication bridges between the two networks. Assisted by supplementary information from another branch, better representation of appearance and geometry features are obtained for pose estimation.

3. Proposed Method

Given an RGBD image, the task of object 6D pose estimation aims to predict a transformation matrix that transforms the object from its coordinate system to the camera coordinate system. Such transformation consists of a rotation matrix $R \in SO(3)$ and a translation matrix $T \in \mathbb{R}^3$. To tackle the problem, pose estimation algorithms should fully explore the texture and the geometric information of

both the scene and the target object.

3.1. Overview

We propose a full flow bidirectional fusion network to solve the problem, as shown in Figure 2(a). The proposed framework first extracts the pointwise RGBD feature for per-object 3D keypoints localization. Then the pose parameters are recovered within a least-square fitting manner. More specifically, given an RGBD image as input, we utilize a CNN to extract appearance features from the RGB image, and a point cloud network to extract geometric features from point clouds. During the feature extraction flow of the two networks, point-to-pixel and pixel-to-point fusion modules are added into each layer as communication bridges. In this way, the two branches can utilize the extra appearance (geometric) information from the other to facilitate their own representation learning. The extracted pointwise features are then fed into an instance semantic segmentation and a 3D keypoint detection module to obtain per-object 3D keypoints in the scene. Finally, a least-square fitting algorithm is applied to recover the 6D pose parameters.

3.2. Full Flow Bidirectional Fusion Network

Given an aligned RGBD image, we first lift the depth image to a point cloud with the camera intrinsic matrix. A CNN and a point cloud network (PCN) is then applied for feature extraction from the RGB image and point cloud respectively. When the information flow through the two networks, point-to-pixel and pixel-to-point fusion modules are added for bidirectional communication. In this way, each branch can leverage local and global information from the other to facilitate their representation learning.

Pixel-to-point fusion from image features to point cloud features. These modules share the appearance information extracted from CNN to the PCN. One naive way is to generate a global feature from the RGB feature map and then concatenate it to each point cloud feature. However, since most of the pixels are background and there are multi objects in the scene, squeezing the RGB feature map globally would lose lots of detailed information and harm the following pose estimation module. Instead, we introduce a novel pixel to point feature fusion module. Since the given RGBD image is well-aligned, we can use the 3D point clouds as a bridge to connect the pixel-wise and the point-wise features. More specifically, we lift the depth of each pixel to its corresponding 3D point with the camera intrinsic matrix and get an XYZ map, aligned with the RGB map. As shown in the left part of Figure 2(b), for each point feature with its 3D point coordinate, we find its K_{r2p} nearest point in the XYZ map and gather their corresponding appearance features from the RGB feature map. We then use max polling to integrate these neighboring appearance features following [50], and apply shared Multi-Layer Perceptrons (MLPs) to squeeze it to the same channel size as the point cloud feature:

$$F_{r2p} = MLP(\max_{i=1}^{K_{r2p}} F_{r_i}), \tag{1}$$

where F_{r_i} is the i_{th} nearest pixel of RGB feature and F_{r2p} the integrated one. We then concatenate the integrated appearance feature F_{r2p} with the point feature F_{point} and use shared MLP to obtained the fused point feature:

$$F_{fused_p} = MLP(F_{point} \oplus F_{r2p}), \tag{2}$$

where $\oplus$ is the concatenate operation.

One thing to mention is that in the flow of the appearance feature encoding, the height and width of the RGB feature maps get smaller when the network goes deeper. Therefore, we need to maintain a corresponding XYZ map so that each pixel of feature can find its 3D coordinate. Since the decreased size of the feature map is generated by convolution kernel scanning through the original feature map with stride, the centers of kernels become new coordinates of the feature maps. One simple way is to apply the same size

of kernels to calculate the mean of XYZ within it to generate a new XYZ coordinate of a pixel. The corresponding XYZ map is then obtained by scanning through the XYZ map with the mean kernel in the same stride as CNN. However, noise points are produced by the mean operation as the depth changes remarkably on the boundary between the foreground objects and the background. Instead, a better solution is to resize the XYZ map to the same size as the feature map within the nearest interpolation algorithm.

Point-to-pixel fusion from point cloud features to image features. These modules build bridges to transfer the geometric information obtained from the PCN to the CNN. The procedure is shown on the right side of Figure 2(b). Same as the pixel-to-point fusion modules, we fuse the feature densely rather than naively concatenating the global point feature to each pixel. Specifically, for each pixel of feature with its XYZ coordinate, we find its K_{p2r} nearest points from the point cloud and gather the corresponding point features. We squeeze the point features to the same channel size as the RGB feature and then use max pooling to integrate them. The integrated point feature is then concatenated to the corresponding color feature and mapped by a shared MLP to generate the fused one:

$$F_{p2r} = \max_{j=1}^{K_{p2r}} (MLP(F_{p_j})), \tag{3}$$

$$F_{fused_r} = MLP(F_{rgb} \oplus F_{p2r}), \tag{4}$$

where F_{p_j} denotes the j_{th} nearest point features, F_{p2r} the integrated point features and $\oplus$ the concatenate operation.

Dense RGBD feature embedding With the proposed full flow fusion network, we obtain dense appearance embeddings from the CNN branch and dense geometry features from the PCN branch. We then find the correspondence between them by projecting each point to the image plane with the camera intrinsic matrix. According to the correspondence, we obtain pairs of appearance and geometry features and concatenate them together to form the extracted dense RGBD feature. These features are then fed into an instance semantic segmentation module and a 3D keypoint detection module for object pose estimation in the next step.

3.3. 3D Keypoint-based 6D Pose Estimation

Recently, the PVN3D [17] work by He *et al.* opens up new opportunities for using 3D keypoints to estimate object pose. In this work, we follow their 3D keypoint formulation but further improve the 3D keypoint selection algorithm to fully leverage the texture and geometry information of objects. Specifically, we first detect the per-object selected 3D keypoints in the scene and then utilize a least-squares fitting algorithm to recover the pose parameters.

Per-object 3D keypoint detection So far we have obtained the dense RGBD embeddings. We then follow

PVN3D [17] and obtain the per-object 3D keypoints by adding an instance semantic segmentation module to distinguish different object instances and a keypoint voting module to recover 3D keypoints. The instance semantic segmentation module consists of a semantic segmentation module and a center point voting module, where the former one predicts per-point semantic labels, and the latter one learns the per-point offset to object centers for distinguishment of different instances. For each object instance, the keypoint voting module learns the point-wise offsets to the selected keypoints that vote for 3D keypoint within a MeanShift [10] clustering manners.

Keypoint selection Previous works [46, 17] select keypoints from the target object surface using the Farthest Point Sampling (FPS) algorithm. Specifically, they maintain a keypoint set initialized by a random point on the object surface and iteratively add other points that are farthest to those within the set until N points are obtained. In this way, the selected keypoints spread on the object surface and stabilize the following pose estimation procedure [46, 17]. However, since the algorithm only takes the Euclidean distance into account, the selected points may appear in non-salient regions, such as flat planes without distinctive texture. These points are hard to detect and the accuracy of the estimated pose decrease. To fully leverage the texture and geometry information of objects, we propose a simple but effective 3D keypoint selection algorithm, named SIFT-FPS. Specifically, we use the SIFT [39] algorithm to detect 2D keypoints that are distinctive in texture images and then lift them to 3D. The FPS algorithm is then applied for the selection of top N keypoints among them. In this way, the selected keypoints not only distribute evenly on the object surface but are also distinctive in texture and easy to detect.

Least-Squares Fitting Given the selected 3D keypoints in the object coordinates system $\{p_i\}_{i=1}^N$, and the corresponding 3D keypoints in the camera coordinated system $\{p_i^*\}_{i=1}^N$. The Least-Squares Fitting [1] algorithm calculate the pose parameters R and T by minimizing the squared loss:

$$L_{\text{lsf}} = \sum_{i=1}^{N} ||p_i^* - (R \cdot p_i + T)||^2. \quad (5)$$

4. Experiments

4.1. Benchmark Datasets

We evaluate our method on three benchmark datasets.

YCB-Video [5] contains 92 RGBD videos that capture scenes of 21 selected YCB objects. We followed previous works [68, 65, 17] to split the training and testing set. Synthetic images were also taken for training as in [68] and the hole completion algorithm [30] is applied for hole filling for depth images as in [17].
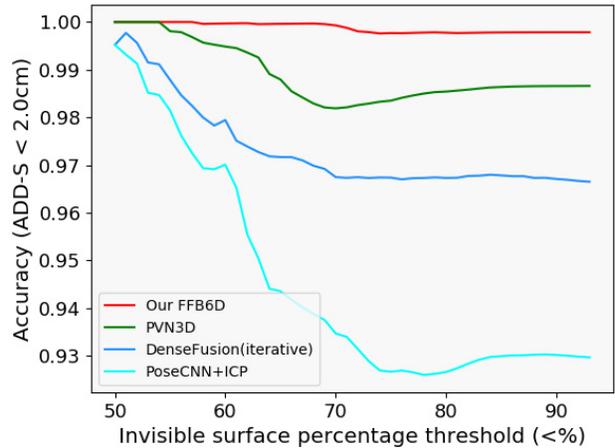

Figure 3: Performance of different approaches under increasing levels of occlusion on the YCB-Video dataset.

LineMOD [19] is a dataset with 13 videos of 13 low-textured objects. The texture-less objects, cluttered scenes, and varying lighting make this dataset challenge. We split the training and testing set following previous works [68, 46] and generate synthesis images for training following [46, 17].

Occlusion LINEMOD [3] was selected and annotated from the LineMOD datasets. Each scene in this dataset consists of multi annotated objects, which are heavily occluded. The heavily occluded objects make this dataset challenge.

4.2. Evaluation Metrics

We use the average distance metrics ADD and ADD-S for evaluation. For asymmetric objects, the ADD metric calculate the point-pair average distance between objects vertexes transformed by the predicted and the ground truth pose, defined as follows:

$$\text{ADD} = \frac{1}{m} \sum_{v \in \mathcal{O}} ||(Rv + T) - (R^*v + T^*)||. \quad (6)$$

where v denotes a vertex in object $\mathcal{O}$, R, T the predicted pose and R^*, T^* the ground truth. For symmetric objects, the ADD-S based on the closest point distance is applied:

$$\text{ADD-S} = \frac{1}{m} \sum_{v_1 \in \mathcal{O}} \min_{v_2 \in \mathcal{O}} ||(Rv_1 + T) - (R^*v_2 + T^*)||. \quad (7)$$

In the YCB-Video dataset, we follows previous methods [68, 65, 17] and report the area under the accuracy-threshold curve obtained by varying the distance threshold (ADD-S and ADD(S) AUC). In the LineMOD and Occlusion LineMOD datasets, we report the accuracy of distance less than 10% of the objects diameter (ADD-0.1d) as in [20, 46].

	PoseCNN [68]		PointFusion [69]		DCF [35]		DF (per-pixel) [65]		PVN3D [17]		Our FFB6D	
Object	ADDS	ADD(S)	ADDS	ADD(S)	ADDs	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)	ADDS	ADD(S)
002 master chef can	83.9	50.2	90.9	-	90.9	74.6	95.3	70.7	96.0	80.5	**96.3**	**80.6**
003 cracker box	76.9	53.1	80.5	-	87.1	79.3	92.5	86.9	96.1	**94.8**	**96.3**	94.6
004 sugar box	84.2	68.4	90.4	-	94.3	84.2	95.1	90.8	97.4	96.3	**97.6**	**96.6**
005 tomato soup can	81.0	66.2	91.9	-	90.5	79.8	93.8	84.7	**96.2**	88.5	95.6	**89.6**
006 mustard bottle	90.4	81.0	88.5	-	90.6	83.5	95.8	90.9	97.5	96.2	**97.8**	**97.0**
007 tuna fish can	88.0	70.7	93.8	-	91.7	73.8	95.7	79.6	96.0	**89.3**	**96.8**	88.9
008 pudding box	79.1	62.7	87.5	-	89.3	84.1	94.3	89.3	**97.1**	**95.7**	97.1	94.6
009 gelatin box	87.2	75.2	95.0	-	92.9	89.5	97.2	95.8	97.7	96.1	**98.1**	**96.9**
010 potted meat can	78.5	59.5	86.4	-	83.2	74.6	89.3	79.6	93.3	**88.6**	**94.7**	88.1
011 banana	86.0	72.3	84.7	-	84.8	71.0	90.0	76.7	96.6	93.7	**97.2**	**94.9**
019 pitcher base	77.0	53.3	85.5	-	89.5	80.3	93.6	87.1	97.4	96.5	**97.6**	**96.9**
021 bleach cleanser	71.6	50.3	81.0	-	88.4	79.8	94.4	87.5	96.0	93.2	**96.8**	**94.8**
024 bowl	69.6	69.6	75.7	75.7	80.3	80.3	86.0	86.0	90.2	90.2	**96.3**	**96.3**
025 mug	78.2	58.5	94.2	-	90.7	76.6	95.3	83.8	**97.6**	**95.4**	97.3	94.2
035 power drill	72.7	55.3	71.5	-	87.4	78.4	92.1	83.7	96.7	95.1	**97.2**	**95.9**
036 wood block	64.3	64.3	68.1	68.1	84.2	84.2	89.5	89.5	90.4	90.4	**92.6**	**92.6**
037 scissors	56.9	35.8	76.7	-	84.2	70.3	90.1	77.4	96.7	92.7	**97.7**	**95.7**
040 large marker	71.7	58.3	87.9	-	89.5	81.0	95.1	89.1	**96.7**	**91.8**	96.6	89.1
051 large clamp	50.2	50.2	65.9	65.9	63.6	63.6	71.5	71.5	93.6	93.6	**96.8**	**96.8**
052 extra large clamp	44.1	44.1	60.4	60.4	64.4	64.4	70.2	70.2	88.4	88.4	**96.0**	**96.0**
061 foam brick	88.0	88.0	91.8	91.8	83.1	83.1	92.2	92.2	96.8	96.8	**97.3**	**97.3**
ALL	75.8	59.9	83.9	-	85.7	77.9	91.2	82.9	95.5	91.8	**96.6**	**92.7**

Table 1: Quantitative evaluation of 6D Pose without iterative refinement on the YCB-Video Dataset. The ADD-S [68] and ADD(S) [20] AUC are reported. Symmetric objects are in bold. DF (per-pixel) means DenseFusion (per-pixel).

	PCNN+ICP	DF(iter.)	MoreFusion	PVN3D+ICP	FFB6D+ICP
ADD-S	93.0	93.2	95.7	96.1	**97.0**
ADD(S)	85.4	86.1	91.0	92.3	**93.1**

Table 2: Quantitative evaluation of 6D Pose with iterative refinement on the YCB-Video Dataset (ADD-S [68] and ADD(S) AUC [20]). Baselines: PoseCNN+ICP [68], DF(iter.) [65], MoreFusion [64], PVN3D+ICP [17].

4.3. Training and Implementation

Network architecture. We apply ImageNet [11] pre-trained ResNet34 [16] as encoder of RGB images, followed by a PSPNet [71] as decoder. For point cloud feature extraction, we randomly sample 12288 points from depth images following [17] and applied RandLA-Net [24] for representation learning. In each encoding and decoding layers of the two networks, max pooling and shared MLPs are applied to build bidirectional fusion modules. After the process of the full flow bidirectional fusion network, each point has a feature $f_i \in \mathbb{R}^C$ of C dimension. These dense RGBD features are then fed into the instance semantic segmentation and the keypoint offset learning modules consist of shared MLPs.

Optimization regularization. The semantic segmentation branch is supervised by Focal Loss [37]. The center point voting and 3D keypoints voting modules are opti-mized by L1 loss as in [17]. To jointly optimize the three tasks, a multi-task loss with the weighted sum of them is applied following [17].

SIFT-FPS keypoint selection algorithm. We put the target object at the center of a sphere and sample viewpoints of the camera on the sphere equidistantly. RGBD images with camera poses are obtained by render engines. We then detect 2D keypoints from RGB images with SIFT. These 2D keypoints are lifted to 3D and transformed back to the object coordinates system. Finally, an FPS algorithm is applied to select N target keypoints out of them.

4.4. Evaluation on Three Benchmark Datasets.

We evaluate the proposed models on the YCB-Video, the LineMOD, and the Occlusion LineMOD datasets.

Evaluation on the YCB-Video dataset. Table 1 shows the quantitative evaluation results of the proposed FFB6D on the YCB-Video dataset. We compare it with other single view methods without iterative refinement. FFB6D advances state-of-the-art results by 1.1% on the ADD-S metric and 1.0% on the ADD(S) metric. Equipped with extra iterative refinement, our approach also achieves the best performance, demonstrated in Table 2. Note that the proposed FFB6D without any iterative refinement even outperforms state-of-the-arts that require time-consuming post-refinement procedures. Qualitative results are reported in

	RGB				RGB-D				
	PoseCNN DeepIM [68, 33]	PVNet[46]	CDPN[34]	DPOD[70]	Point-Fusion[69]	Dense-Fusion[65]	G2L-Net[7]	PVN3D[17]	Our FFB6D
MEAN	88.6	86.3	89.9	95.2	73.7	94.3	98.7	99.4	**99.7**

Table 3: Quantitative evaluation of 6D pose on the LineMOD dataset (ADD-0.1d [20] metrics).

Method	PoseCNN [68]	Oberweger [43]	Hu et al. [26]	Pix2Pose [45]	PVNet [46]
ADD-0.1d	24.9	27.0	27.0	32.0	40.8
Method	DPOD [70]	Hu et al.[25]	HybridPose [55]	PVN3D [17]	Our FFB6D
ADD-0.1d	47.3	43.3	47.5	63.2	**66.2**

Table 4: Quantitative evaluation of 6D pose (ADD-0.1d) on the Occlusion-LineMOD dataset.

Fusion Stage			Pose Result	
FE	FD	DF	ADD-S	ADD(S)
			91.9	87.5
✓			96.2	92.2
	✓		93.0	89.2
		✓	94.0	90.8
✓	✓		**96.6**	**92.7**
✓	✓	✓	96.4	92.6

Table 5: Effect of fusion stages on the YCB-Video dataset. FE: fusion during encoding; FD: fusion during decoding; DF: Dense Fusion on the two final feature maps.

Fusion Direction		Pose Result	
P2R	R2P	ADD-S	ADD(S)
		91.9	87.5
✓		95.8	91.1
	✓	94.5	90.6
✓	✓	**96.6**	**92.7**

Table 6: Effect of fusion direction on the YCB-Video dataset. P2R means fusion from point cloud embeddings to RGB embeddings, and R2P means fusion from RGB embeddings to point cloud embeddings.

the supplementary material.

Robustness towards occlusion. We follow [65, 17] to report the ADD-S less than 2cm accuracy under the growth of occlusion level on the YCB-Video dataset. As is shown in Figure 3, previous methods degrade as the occlusion increase. In contrast, FFB6D didn't suffer from a drop in performance. We think our full flow bidirectional fusion mechanism makes full use of the texture and geometry information in the captured data and enables our approach to locate 3D keypoints even in highly occluded scenes.

Evaluation on the LineMOD dataset & Occlusion LineMOD dataset. The proposed FFB6D outperforms the state-of-the-art on the LineMOD dataset, presented in Table 3. We also evaluate FFB6D on the Occlusion LineMOD dataset, shown in Table 4. In the table, our FFB6D without iterative refinement advances state-of-the-art by 4.7%, further confirming its robustness towards occlusion.

4.5. Ablation Study

In this subsection, we present extensive ablation studies on our design choices and discuss their effect.

Effect of full flow bidirectional fusion. To validate that building fusion modules between the two modality networks in full flow help, we ablate fusion stages in Table 5. Compared to the mechanism without fusion, adding fusion modules either on the encoding stage, decoding stage,

or on the final feature maps, can all boost the performance. Among the three stages, fusion on the encoding stages obtained the highest improvement. We think that's because the extracted local texture and geometric information are shared through the fusion bridge on the early encoding stage, and more global features are shared when the network goes deeper. Also, adding fusion modules in full flow of the network, saying that on both the encoding and decoding stages, obtains the highest performance. While adding extra DenseFusion behind full flow fusion obtains no performance gain as the two embeddings have been fully fused.

We also ablate the fusion direction in Table 6 to validate the help of bidirectional fusion. Compared with no fusion, both the fusion from RGB to point cloud and the inverse way facilitate better representation learning. Combining the two obtains the best results. On the one hand, from the view of PCN, we think the rich textures information obtained from high-resolution RGB images helps semantic recognition. In addition, the high-resolution RGB features provide rich information for blind regions of depth sensors caused by reflective surfaces. It serves as a completion to point cloud and improve pose accuracy, as shown in Figure 4(a). On the one hand, geometry information extracted from point cloud helps the RGB branch by distinguishing foreground objects from the background that are in similar colors. Moreover, the shape size information extracted from point clouds helps divide objects with a similar appearance but in a different size, as is shown in Figure 4(b).

Effect of representation learning frameworks. We explore the effect of different representation learning frameworks for the two modalities of data in this part. The result is presented in Table 7. We find that neither concatenating the XYZ map as extra information to CNN (CNN-R⊕D) nor adding RGB values as extra inputs to the PCN (PCN-R⊕D) achieves satisfactory performance. Using two CNNs
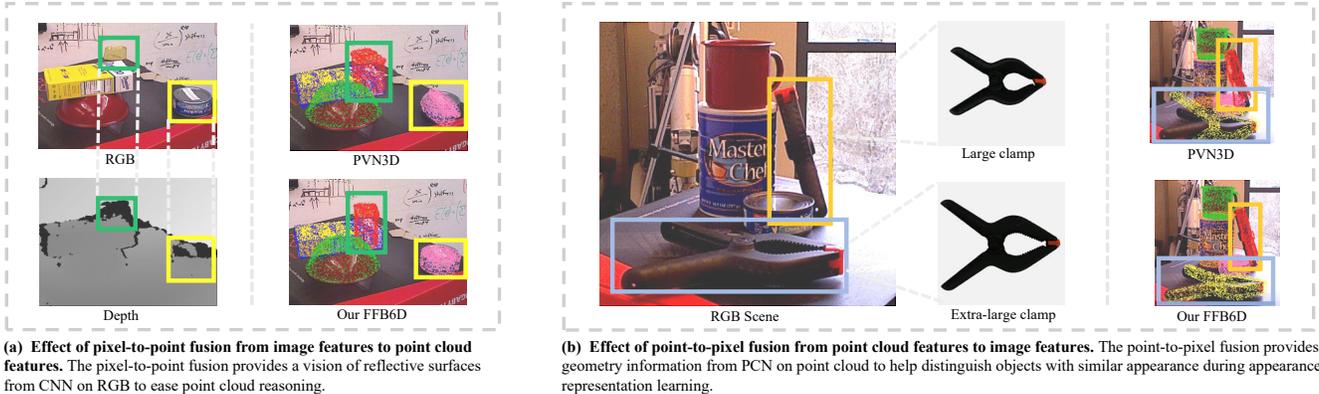
(a) **Effect of pixel-to-point fusion from image features to point cloud features.** The pixel-to-point fusion provides a vision of reflective surfaces from CNN on RGB to ease point cloud reasoning.

(b) **Effect of point-to-pixel fusion from point cloud features to image features.** The point-to-pixel fusion provides geometry information from PCN on point cloud to help distinguish objects with similar appearance during appearance representation learning.

Figure 4: Effect of full flow bidirectional fusion, compared to PVN3D [17] with DenseFusion architecture.

	CNN-R⊕D	PCN-R⊕D	CNN-R+CNN-D
ADD-S	91.0	90.9	92.1
ADD(S)	85.5	81.2	87.2
	PCN-R+PCN-D	CNN-R+3DC-D	CNN-R+PCN-D
ADD-S	91.8	94.9	**96.6**
ADD(S)	84.3	90.1	**92.7**

Table 7: Effect of representation learning framework on the two modalities of data. CNN: 2D Convolution Neural Network; PCN: point cloud network; 3DC: 3D ConvNet; R: RGB images; D: XYZ maps for CNN, point clouds for PCN and voxelized point clouds for 3D ConvNet.

	FPS4	S-F4	FPS8	S-F8	FPS12	S-F12
KP err. (cm)	1.3	1.2	1.4	**1.2**	1.6	1.3
ADD-S	95.9	96.4	96.1	**96.6**	96.0	96.5
ADD(S)	92.0	92.4	92.3	**92.7**	92.0	92.6

Table 8: Effect of keypoint selection algorithm. S-F means the proposed SIFT-FPS algorithm.

(CNN-R+CNN-D) or two PCNs (PCN-R+PCN-D) with full flow bidirectional fusion modules get better but are still far from satisfactory. In contrast, applying CNN on the RGB image and PCN on the point cloud (CNN-R+PCN-D) gets the best performance. We think that the grid-like image data is discrete, on which the regular convolution kernel fits better than continuous PCN. While the geometric information residing in the depth map is defined in a continuous vector space, and thus PCNs can learn better representation.

Effect of 3D keypoints selection algorithm. In Table 8, we study the effect of different keypoint selection algorithms. Compared with FPS that only considers the mutual distance between keypoints, our SIFT-FPS algorithm taking both object texture and geometry information into account is easier to locate. Therefore, the predicted keypoint error is smaller and the estimated poses are more accurate.

Effect of the downsample strategy of the assisting XYZ map. The size of RGB feature maps are shrunk by

	Parameters	Run-time (ms/frame)		
		NF	PE	All
PVN3D[17]	39.2M	170	20	190
Our FFB6D	33.8M	57	18	75

Table 9: Model parameters and run-time breakdown on the LineMOD dataset. NF: Network Forward; PE: Pose Estimation. Our FFB6D with fewer parameters is 2.5x faster.

stridden convolution kernels. To maintain the corresponding XYZ maps, we first scale it down with the same size of mean kernels and got 96.3 ADD-S AUC. However, simply resize the XYZ map with the nearest interpolation got 96.6. We find the average operation produces noise points on the boundary and decrease the performance.

Model parameters and time efficiency. In Table 9, we report the parameters and run-time breakdown of FFB6D. Compared to PVN3D [17], which obtained the fused RGBD feature by dense fusion modules [65] in the final layers, our full flow bidirectional fusion network achieve better performance with fewer parameters and is 2.5 times faster.

5. Conclusion

We propose a novel full flow bidirectional fusion network for representation learning from a single RGBD image, which extract rich appearance and geometry information in the scene for pose estimation. Besides, we introduce a simple but effective SIFT-FPS keypoint selection algorithms that leverage texture and geometry information of objects to simplify keypoint localization for precise pose estimation. Our approach outperforms all previous approaches in several benchmark datasets by remarkable margins. Moreover, we believe the proposed full flow bidirectional fusion network can generalize to more applications built on RGBD images, such as 3D object detection, 3D instance semantic segmentation and salient object detection etc. and expect to see more future research along this line.

References

[1] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987. 5

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014. 2

[3] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014. 5

[4] Ming Cai and Ian Reid. Reconstruct locally, localize globally: A model free method for object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3153–3163, 2020. 2

[5] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 1, 5

[6] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11973–11982, 2020. 2

[7] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, and Ales Leonardis. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4233–4242, 2020. 7

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 1

[9] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011. 1

[10] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):603–619, 2002. 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[12] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3583–3592, 2016. 2

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1

[14] Daniel Glasner, Meirav Galun, Sharon Alpert, Ronen Basri, and Gregory Shakhnarovich. aware object detection and pose estimation. In *2011 International Conference on Computer Vision*, pages 1275–1282. IEEE, 2011. 2

[15] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *European Conference on Computer Vision*, pages 408–421. Springer, 2010. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11641, 2020. 1, 2, 4, 5, 6, 7, 8

[18] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, 2011. 2

[19] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011. 1, 2, 5

[20] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012. 2, 5, 6, 7

[21] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11703–11712, 2020. 2

[22] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 1

[23] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 1

[24] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 11108–11117, 2020. 2, 6

[25] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2020. 7

[26] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019. 7

[27] Daniel P Huttenlocher, Gregory A Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993. 2

[28] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *European Conference on Computer Vision*, pages 205–220. Springer, 2016. 2

[29] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2

[30] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22. IEEE, 2018. 5

[31] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[32] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018. 1, 2

[33] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 2, 7

[34] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7678–7687, 2019. 1, 2, 7

[35] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 2, 6

[36] Joerg Liebelt, Cordelia Schmid, and Klaus Schertler. independent object class detection using 3d feature maps. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[38] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11602–11610, 2020. 2

[39] David G Lowe et al. Object recognition from local scale-invariant features. In *iccv*, volume 99, pages 1150–1157, 1999. 5

[40] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015. 1

[41] Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother. Global hypothesis generation for 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2017. 2

[42] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2

[43] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2, 7

[44] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10710–10719, 2020. 2

[45] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7668–7677, 2019. 7

[46] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 1, 2, 5, 7

[47] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 2

[48] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 1, 2

[49] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[50] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 4

[51] Reyes Rios-Cabrera and Tinne Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proceedings of the IEEE international conference on computer vision*, pages 2048–2055, 2013. 2

[52] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006. 2

[53] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 2

[54] Jianzhun Shao, Yuhang Jiang, Gu Wang, Zhigang Li, and Xiangyang Ji. Pfrl: Pose-free reinforcement learning for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11454–11463, 2020. 2

[55] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 431–440, 2020. 7

[56] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014. 2

[57] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. 2

[58] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 2

[59] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *European Conference on Computer Vision*, pages 658–671. Springer, 2010. 2

[60] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 2

[61] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In *European Conference on Computer Vision*, pages 462–477. Springer, 2014. 2

[62] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. 1

[63] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 2

[64] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14540–14549, 2020. 2, 6

[65] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 1, 2, 5, 6, 7, 8

[66] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2

[67] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3109–3118, 2015. 2

[68] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 2, 5, 6, 7

[69] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 1, 2, 6, 7

[70] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1941–1950, 2019. 1, 7

[71] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 6

[72] Wanqing Zhao, Shaobo Zhang, Ziyu Guan, Wei Zhao, Jinye Peng, and Jianping Fan. Learning deep network for detecting 3d object keypoints and 6d poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14134–14142, 2020. 2

[73] Guangliang Zhou, Yi Yan, Deming Wang, and Qijun Chen. A novel depth and color feature fusion framework for 6d object pose estimation. *IEEE Transactions on Multimedia*, 2020. 2

[74] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2