6D Object Pose Estimation with Pairwise Compatible Geometric Features

Muyuan Lin[1], Varun Murali[1] and Sertac Karaman[1]

***Abstract*— This work addresses the problem of 6-DoF pose estimation under heavy occlusion. While previous work demonstrates reasonable results in unoccluded situations, robust and efficient pose estimation is still challenging in heavily occluded and low-texture scenarios which are ubiquitous in many applications. To this end, we propose a novel end-to-end deep neural network model recovering object poses from depth measurements. The proposed model enforces pairwise consistency of 3D geometric features by applying spectral convolutions on a pairwise compatibility graph. We achieve comparable accuracy as the state-of-the-art graph matching solver while being much faster. Our approach outperforms state-of-the-art 6-DoF pose estimation methods on LineMOD and Occlusion LineMOD and runs in reasonable time (∼5.9 Hz). We additionally verify this method on a synthetic dataset with large affine changes.**

I. Introduction

Real time 6D (i.e. 3D translation and 3D rotation) object pose estimation in cluttered scenes is an important yet challenging task. The success of many applications such as bin picking [1], autonomous driving [2] and augmented reality [3] rely on accurate and robust pose estimation. For instance, consider the problem of navigating traffic on a busy street. To estimate the likelihood of collision it is required to estimate the pose of other vehicles and possibly pedestrians on the road with reference to our vehicle. Similarly, robots operating in a warehouse are required to estimate the pose of objects in a bin to plan a successful grasp. These applications have generated a large amount of interest in this problem but existing methods tend to fail in challenging scenarios such as environments which contain small, thin objects with limited texture and under heavy occlusion (see Fig. 1).

The availability of depth data has greatly grown with the access to inexpensive 3D sensors such as Kinect. This has fueled the direct use of depth information for computer vision tasks such as object recognition, segmentation and pose estimation. While color images contains texture rich information, they fail to capture the true geometry of the scene. Depth measurements on the other hand provide not only the geometry of an object or an environment, but also the robustness to different lighting conditions and low-texture objects. Vast amount of literature on color and RGB-D based approaches are available, yet few provides investigation into effective 6D pose learning on depth measurements.

In general, existing work offers their solutions in a two-step framework: 1) aggregate features to generate a pool of hypotheses, in the form of pose or indirect prediction of geometric quantities; 2) prune generated hypotheses using robust estimators such as RANSAC and Hough-voting. For example, before the recent wave of neural network research, 3D hand-crafted features such as Point Pair Features (PPF), FPFH [4], spin image [5] and SHOT [6] are often used to detect object in the scene [7], [8]. A Hough-voting scheme is then used to achieve a maximum consensus on a large number of hypotheses. Since they rely on hand-crafted features which are often sensitive to noise, brittle in the presence of outliers, traditional methods are often required to find solutions in high dimensionality of the search space. While these methods report competitive results, they are often too slow to apply in real-time applications.

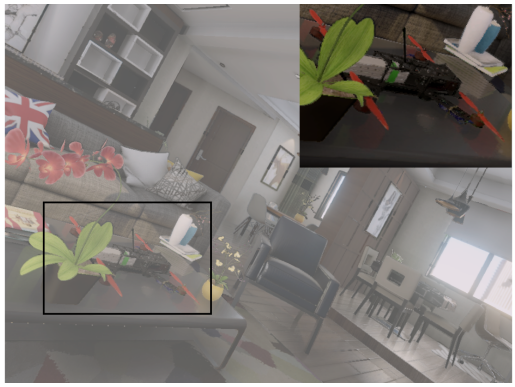


Figure 1. An example object used in this work from the FlightGoggles simulator. For typical robotic applications such as state estimation, grasping etc. it is required to estimate the pose of the object.

Of the existing methods, deep convolutional neural networks (ConvNets) achieve superior performance on RGB images. Deep ConvNets are able to extract more discriminative features from data therefore the chance to obtain high quality pose hypotheses increases. In comparison to the dominance of 2D ConvNets, feature learning research from 3D point cloud has only recently become popular [9], [10], [11]. Point-cloud-based learned features are mostly inferior to their RGB counterparts and thus it is often required to run excessive post-processing to obtain high-quality pose estimation.

Recent work such as [12], [13] predicts the object poses from RGB-D images by adding the depth data as an additional input channel. [11], [9], [10], [14] solve the 3D point cloud registration problem via deep metric learning. However, these methods relies on locating correspondences between two point cloud segments which easily fails in cases where two point clouds share a small overlap or have

[1] Authors are with Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge MA 02139, USA. {mylin, mvarun, sertac}@mit.edu

different point densities. These difficulties are unavoidable either due to heavy occlusion or varied viewpoints.

Learning-based image and point cloud features [15], [16], [17], [18], [19] have gained more attention recently. Methods in [11], [20], [21], [22] take as input a raw point cloud or a low-dimensional rotation invariant feature to train 3D local features. While achieving much better results than methods with handcrafted features, the receptive field of these models is often limited since only a small 3D patch or a set of points are related. [10] proposes a 3D fully-convolutional network to learn a compact geometric feature using a broader spatial context. However, estimating 6D pose by directly comparing features extracted from sensor data and the CAD model does not work well due to heavy occlusion. There is also related work training ConvNets end-to-end [23], [24], [25] and directly regresses 6D pose from data. [26] trains a deep ConvNet to map input images to object viewpoints. Recent work such as [27] and [25] predicts either a direction vector pointing to keypoints or offsets to the center of target objects for each point and achieves satisfactory accuracy.

In addition to feature learning and hypotheses generation, another direction of research focuses on developing robust estimators. Local reasoning via RANSAC and Hough-voting is frequently used [7], [8], [27]. Graphical models such as Conditional Random Fields (CRFs) could be used to achieve global reasoning [28]. ICP is arguably the most popular algorithm to refine initial pose estimation. DeepIM [29] iteratively refines pose estimation by estimating residual transformation between the rendered image against the observed image. DenseFusion [25] uses a refinement network to iteratively adjust pose estimation. Another perspective of research is to formulate the point cloud registration as an optimization problem which aims to recover global optimal solution from data containing outliers [30], [31].

Graph matching problem is also closely related to our problem. [32] formulates a maximum clique problem (MCP) to deal with point clouds with a large number of outliers in 3D registration problem. Their study has shown that an exact solver for MCP is robust to 98% outlier ratio. Comparing to RANSAC whose runtime grows exponentially as outlier ratio increases, graph-based formulation can be very efficient as graphs are sparse when a large number of outliers presents. However, solving MCP is NP-hard in general and the solver is expensive at runtime on dense graphs.

The method presented here aims to take advantage of the representative power of deep learning-based models and improve pairwise consistency of predictions. We show that the proposed method effectively regularizes output predictions and provide an efficient way to prune hypotheses which makes our model robust to occlusions. Comparing to exisitng RGB- or RGB-D-based methods, we only utilize depth images but still get better results. This work proposes an end-to-end trainable deep neural network model to address challenges of 6D pose estimation under heavy occlusion. The proposed algorithm is able to significantly improve the estimation accuracy on heavily occluded objects in comparison to RGB-based methods (see Section IV-A). In contrast to earlier work using point clouds, we propose to enforce pairwise compatibility constraint (*PCC*) of learned 3D geometric features by applying spectral convolutions on a pairwise compatibility graph. To show the advantages of our proposed method, we benchmark our algorithm against state-of-the-art RGB- and RGB-D-based approaches on widely used datasets and showcase it's applicability to large affine changes on a synthetic dataset created in FlightGoggles [33]. We also release our code to facilitate reproducibility and future research. In summary, the contributions of this work are: ***(i)*** We propose an end-to-end deep neural network model on point clouds. To overcome the fragility of localization and 3D deep learning network, we propose to enforce pairwise compatibility of 3D geometric features on a pairwise compatibility graph which greatly boosts the performance; ***(ii)*** The proposed learning module of outlier rejection on the compatibility graph is extremely efficient as compared to the state-of-the-art exact maximum clique solver; and ***(iii)*** Additionally, we create and release a synthetic pose prediction dataset in the FlightGoggles environment. We verify our method on this challenging dataset that features heavy occlusion and large viewpoint changes.

II. Method

We use bold lowercase characters for real vectors, and bold uppercase characters for real matrices. $[\mathbf{R} \mid \mathbf{t}]$ denotes a rigid body transformation, with $\mathbf{R} \in SO(3)$ a rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ a translation vector. We can now formally describe the problem under consideration. Given a set of depth measurements potentially containing one of multiple objects of interest, the objective is to detect each target object in the scene and estimate its 6D pose $[\mathbf{R} \mid \mathbf{t}]$. The pose transforms each point $\mathbf{x}_i \in \mathbf{X}$ in object coordinate frame into a point $\mathbf{y}_i \in \mathbf{Y}$ in the sensor frame.

We describe here our approach to the problem of 6D pose estimation in cluttered scenes where target objects are commonly partially occluded and poorly textured. Given a depth image, our approach (see Fig. 2) first localizes target objects using an off-the-shelf segmentation network [34]. We then extract pointwise 3D geometric features from the found subset of points using an adapted fully convolutional encoder-decoder network [36] (Section II-A). From which we regress 3D object coordinates via a fully connected layer. However, the extracted features are not regularized thus the corresponding predictions are very noisy. In Section II-B we first construct a compatibility graph and identify that the problem of hypothesis pruning is indeed a special case of graph matching. In Section II-C we show how to impose pairwise consistency of 3D geometric features using spectral convolutions on the defined compatibility graph and predict inliers/outliers without running computationally expensive graph matching solvers. We introduce loss functions in Section II-D.

A. Segmentation and Extraction of 3D Geometric Features

To localize multiple target objects in a clustered scene, we run an off-the-shelf segmentation network [34] on the input

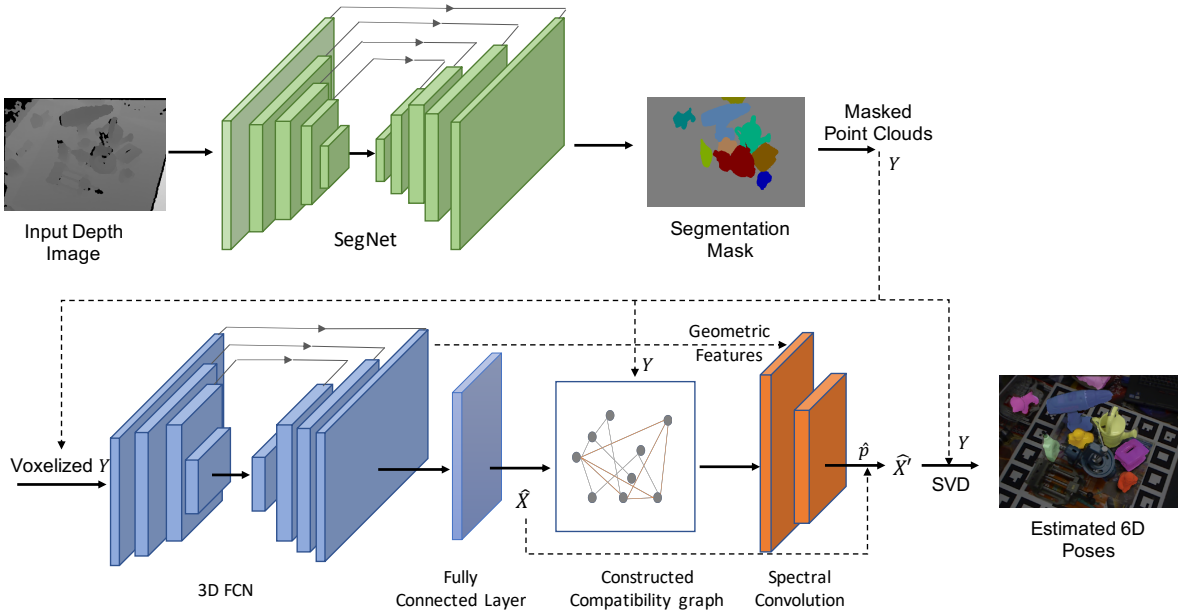


Figure 2. Overview of our method. A depth image is first processed by a SegNet [34] to obtain multiple point clouds Y containing target objects. We extract 3D geometric features via an FCN and predict pointwise 3D object coordinates $\hat{X}$ for each point cloud one at a time. To impose pairwise compatibility of 3D geometric features we apply spectral convolutions on the compatibility graph. Top k predictions $\hat{X}'$ selected by the inlier probability $\hat{p}$ are used to estimate 6D Poses of multiple objects by computing the singular value decomposition (SVD) of a derived matrix [35].

depth image. Segmentation on depth images produces worse results than its color counterpart, which makes downstream tasks more challenging. Nevertheless, we show that by effectively taking advantage of geometric information, we are actually able to compensate the loss of accuracy on segmentation. Our model then extracts 3D geometric features from point clouds using a fully convolutional encoder-decoder network [10]. The input point cloud is discretized into voxel grids with 3-dimensional spatial coordinates and 1-dimensional feature vector. In our case, we simply set the input feature as an all-ones vector (to indicate occupancy).

B. Construction of Compatibility Graph

The fragility of the segmentation and 3D feature extractor yields a large number of spurious outliers. Most previous work uses RANSAC [24], [27] to iteratively sample a least set of points and find a valid hypothesis with the maximum consensus. However, RANSAC is a non-deterministic algorithm and its computational time increases exponentially as the ratio of outliers increases. On the other hand, state-of-the-art graph matching methods are robust to a large amount of outliers with competitive inference time when graphs are sparse. We are motivated to find methods that can efficiently remove outliers from noisy coordinate predictions even when graphs are dense. For an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we denote $\mathcal{V}$ as the vertex set, and $\mathcal{E}$ as the edge set. Given an input point cloud $\mathbf{Y}$ and predicted 3D object coordinates $\hat{\mathbf{X}}$, we define a *compatibility graph* as the graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$ whose vertices are correspondence pairs between $\hat{\mathbf{X}}, \mathbf{Y}$, and there is an edge $(v_i, v_j) \in \mathcal{E}_c$ between $v_i = (\mathbf{x}_i, \mathbf{y}_i)$ and $v_j = (\mathbf{x}_j, \mathbf{y}_j)$ if they are compatible. Note our task of removing outliers from predictions is a special case of graph matching problem. Target objects considered in our application only undergo rigid body transformation thus the Euclidean distance between any two points on the object is rotation and translation invariant. We thus define two correspondence pairs as compatible if $d(v_i, v_j) = |\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \|\mathbf{y}_i - \mathbf{y}_j\|_2| < \varepsilon$, where ε is a problem specific threshold. Our goal is to find the largest set of correspondences that is pairwise compatible in the graph, which turns out to be a maximum clique problem (MCP).

C. Spectral Convolutions on the Graph

Solving MCP is in general NP-hard while state-of-the-art exact solvers are typically too computationally expensive in practice especially when the compatibility graphs are dense. To address this issue, we propose to integrate a graph convolutional network (GCN) [37] module performing inference on the defined compatibility graph. The intuition is that the graphs are embedded in an object-specific manifold and we should be able to efficiently find an approximated maximum clique by exploiting the power of GCNs in graph analysis. The GCN module classifies which correspondence pair belongs to the maximum clique in the compatibility graph. In addition, training the GCN module with the entire model end-to-end regularizes 3D geometric features through pairwise compatibility constraint (PCC). The feed forward propagation in GCNs is

$$H^{l+1} = \sigma(AH^l W^l), \tag{1}$$

where $A = \tilde{D}^{-1/2}(M+I)\tilde{D}^{-1/2}$ is the normalized adjacency matrix of the compatibility graph, M the adjacency matrix,

$\tilde{D}$ the degree matrix of $M+I$, H^l the output of the l^{th} layer, $\sigma(\cdot)$ a nonlinear activation function, W^l the filter matrix of the l^{th} layer. The input features of the GCN module are the learned 3D geometric features in Section II-A.

As (1) is the first order approximation of Laplacian operator [37], spectral convolutions on the compatibility graph can be interpreted as diffusion operators which help to regularize the 3D feature embeddings. Therefore our model enforces PCC on the compatibility graph resulting in more accurate coordinate predictions. In the experiments, we show that our design is highly efficient and our results provide a well approximated solution to MCP.

D. Loss Functions

There are two loss terms for training the proposed network (except segmentation): the L_1 norm of coordinate prediction and the cross entropy loss of inlier/outlier classification

$$L_{total} = L_{coords} + \alpha \cdot L_{cls}, \tag{2}$$

$$L_{coords} = \frac{1}{N}\sum_{i}^{N} \left\| \hat{\mathbf{x}}_i - \mathbf{x}_i \right\|_1, \tag{3}$$

$$L_{cls} = \frac{1}{N}\sum_{i=1}^{N} -(y_i \log(p_i) + (1-y_i)\log(1-p_i)), \tag{4}$$

where $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i$ are predicted and ground truth coordinates of 3D points sampled from object models, respectively. N is the number of sampled points, $y_i \in \{0,1\}$ a binary label, $p_i \in [0,1]$ the predicted probability for correspondence pair v_i, and $\alpha \in \mathbb{R}$ a weight parameter.

III. Experiments

A. Implementation Details

We implement our model in PyTorch framework [38] and use Minkowski Engine [36] as the auto-differentiation library for sparse tensors. All runtime measurements for our algorithm are performed on a desktop computer equipped with one Intel i9-7900X processor and one NVIDIA GTX 1080 Ti GPU of 11 GB memory. We use Adam [39] as the optimizer with an initial learning rate of 1×10^{-4} for the first 200 epochs of training. It then decays by half every 100 epochs. The classification loss term is only added with $\alpha = 0.01$ when the coordinate loss stop decaying. It takes 500 epochs in total to train the model. The value for selecting top k predictions is set as $k = 70$ in experiments but we found that a wide range of values can be used as shown in Fig. 3.

a) Detection of multiple objects: Our method is able to handle cases in which multiple objects are present as shown in Fig. 2. Note, in the multiple object setting each masked point cloud is fed into the 3D FCN sequentially and is processed as a single object.

b) Data preprocessing and augmentation: The network takes as input a depth image and localizes target objects by a SegNet. Masked depth pixels are then back projected to a point cloud in camera frame and converted to voxel representation with a fixed voxel size of 3 *mm* for LineMOD and Occlusion LineMOD, and 20 *mm* for FlightGoggles dataset. We augment training data on the fly by adding Gaussian noise with zero mean and standard deviation 5 *mm*, and applying a random rotation to the input point clouds.

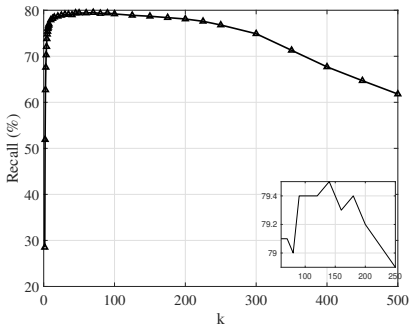


Figure 3. A comparison of selection of parameter k vs recall.

B. Datasets

We use the following datasets for evaluation: ***(i) LineMOD:*** [40]: This dataset is widely used as a benchmark for the problem of 6D pose estimation. Objects in clutter are captured using an RGB-D camera at varied viewpoints. ***(ii) Occlusion LineMOD:*** [41]: This dataset is a re-annotation of the bench vise segment of the original LineMOD dataset. All objects of each frame are annotated instead of just the bench vise and are much more heavily occluded in comparison to the original dataset. ***(iii) FlightGoggles Pose:*** We create a synthetic dataset using the FlightGoggles simulator. The simulator automatically generates a large corpus of data with perfect RGB-D images, segmentation and ground truth pose, which is impossible to achieve in real scenario.

C. Evaluation Metric

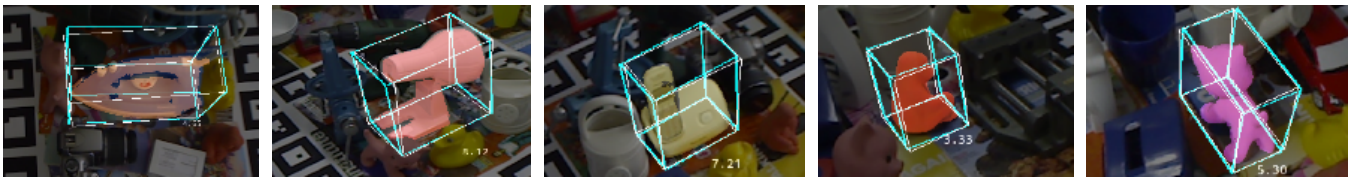
The ADD(-S) metric computes the average distance between the 3D model points transformed using the ground truth and predicted pose. For symmetric objects, it is defined as

$$m = \frac{1}{N}\sum_{i=0}^{N} \min_j \left\| (\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\hat{\mathbf{R}}\mathbf{x}_j + \hat{\mathbf{t}}) \right\|_2, \tag{5}$$

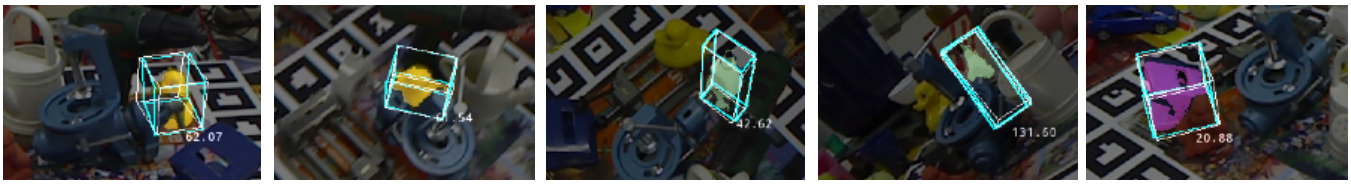
where $\min_j(\cdot)$ is removed from above equation for non-symmetric objects as a unique unambiguous match exists. In all evaluations, each prediction is correct if $m \leq \gamma d$, where d is the diameter of an object model, γ is a chosen coefficient. We use the same value of $\gamma = 0.1$ as previous work [40] and report the recall on all experiments except as otherwise indicated.

IV. Results

We first report results supporting our claim that our proposed method using only depth images achieves state-of-the-art performance on two widely used datasets. We then present our pose estimation dataset created in FlightGoggles simulator which provides a more controllable setting to verify pose estimation algorithms, and show that naively applying a state-of-the-art RGBD-based convolutional neural network model gives sub-optimal accuracy on depth images.



(a) LineMOD dataset



(b) Occlusion LineMOD dataset

Figure 4. Cropped and darkened test images overlaid with renderings of the object model in estimated 6D poses. White and green 3D bounding boxes represent ground truth and predicted poses, respectively. Texts next to bounding boxes indicate the mean difference between depth and rendered depth.

We provide our insights on contributions of *PCC* via ablation studies. Lastly, runtime of each component is reported.

A. Comparison with the State-of-the-Art on LineMOD and Occlusion LineMOD Datasets

a) Performance on LineMOD dataset: We evaluate performance of our method and compare pose estimation recall against state-of-the-art methods [15], [27], [24], [42], [25], [40], [8]. Among them, [40] and [8] are not deep learning-based approaches but report previous state-of-the-art recall with heavy post-processing (ICP). We present the quantitative results in Table I. The first observation we draw from the row of the mean recall is that, with a more consistent geometric representation, we can eliminate the gap with methods using RGB [15], [27] even without ICP refinement. Secondly, by comparing results on texture-less target objects, e.g. ape, we see that our method helps relieve the limitation of RGB-based methods. Finally, our proposed approach achieves highest recall even without ICP refinement. "MCS" in the table denotes an approximated maximum clique solver using k-core heuristic [43].

b) Performance on Occlusion LineMOD dataset: We verify the robustness of our method to heavy occlusion on the Occlusion LineMOD dataset (see Table II). We firstly conclude that methods that infer on depth measurements are more robust to occlusion in comparison to RGB-based methods. All state-of-the-art RGB-based methods perform poorly on the occlusion dataset. Secondly, our method outperforms all previous approaches on Occlusion LineMOD dataset and achieves highest recall even without ICP refinement.

c) Qualitative results: We present qualitative results on LineMOD and Occlusion LineMOD datasets in Fig. 4a and Fig. 4b. The predicted poses are satisfactory comparing to the ground truth even under heavy occlusion.

B. Results on FlightGoggles Pose Dataset

We additionally provide results on FlightGoggles Pose dataset, which is created mainly for quadrotor simulation. The selected target object drone has thin structure while most objects of existing datasets are solid. Our method achieves a higher recall (+7%) than state-of-the-art RGBD-based method as shown in Table III. The challenge due to heavy occlusion, large affine changes and inaccurate segmentation is well addressed. An example is shown in Fig. 5.

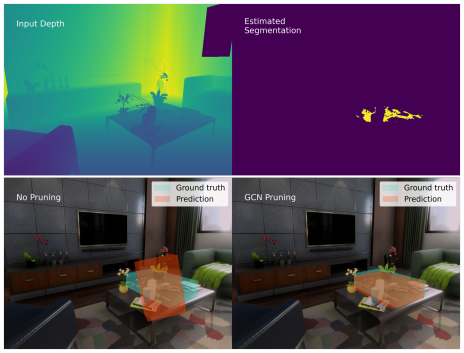


Figure 5. An example drawn from FlightGoggles Pose dataset in which SVD [35] fails due to heavy occlusion while our method succeeds.

C. Ablation Studies

We perform ablation studies to further evaluate the performance boost contributed by the proposed pairwise compatibility constraint. We also compare different hypothesis pruning approaches.

Pairwise Compatibility Constraint (PCC): We compare results with the model trained without PCC in Table IV. Adding PCC improves pose estimation recall with little added runtime mainly from graph construction.

RANSAC vs. MCS vs. Ours: Here we compare hypothesis pruning results of different approaches. We use off-the-shelf RANSAC implementation from Open3D [44] and reports numbers with 1K and 10K iterations. For MCS we use the state-of-the-art maximum clique solver originally reported in [43]. Note the exact solver does not finish the evaluation within allowable maximum runtime (30 minutes). Instead, we report numbers of approximated solutions using k-core heuristic from [43]. As shown in Table V, our approach provides comparable results to RANSAC and MCS while being faster.

Table I

POSE ESTIMATION RECALL ON LINEMOD DATASET [40]. * INDICATES SYMMETRIC OBJECTS. HIGHEST RECALL ARE IN BOLD.

	RGB			RGBD			Depth only		
Method	[15]	[27]	[24]	[42]	[25]	[40]	[8]	Ours	Ours+MCS
Refinement	-	-	DeepIM	ICP	[25]	ICP	ICP-variant	-	-
ape	40.4	43.6	77.0	20.6	92.3	95.8	**98.5**	94.3	95.3
bench	91.8	**99.9**	97.5	64.3	93.2	98.7	99.8	99.0	100.0
camera	55.7	86.9	93.5	63.2	94.4	97.5	**99.3**	99.0	98.8
can	64.1	95.5	96.5	76.1	93.1	95.4	**98.7**	97.0	96.8
cat	62.6	79.3	82.1	72.0	96.5	99.3	**99.9**	96.0	99.0
driller	74.4	96.4	95.0	41.6	87.0	93.6	93.4	97.0	**99.0**
duck	44.3	52.6	77.7	32.4	92.3	95.9	**98.2**	91.6	92.5
eggbox*	57.8	99.2	97.1	98.6	99.8	99.8	98.8	**100.0**	99.1
glue*	41.2	95.7	99.4	96.4	**100.0**	91.8	75.4	**100.0**	99.0
hole p.	67.2	81.9	52.8	49.9	92.1	95.9	**98.1**	97.1	**98.1**
iron	84.7	**98.9**	98.3	63.1	97.0	97.5	98.3	96.9	98.0
lamp	76.5	**99.3**	97.5	91.7	95.3	97.7	96.0	97.1	99.0
phone	54.0	92.4	87.7	71.0	92.8	93.3	98.6	**100.0**	**100.0**
MEAN	62.7	86.3	88.6	64.7	94.3	96.4	96.4	97.2	**97.8**

Table II

POSE ESTIMATION RECALL ON OCCLUSION LINEMOD DATASET [13]. * INDICATES SYMMETRIC OBJECTS. † INDICATES RESULTS ARE WITH ICP REFINEMENT. HIGHEST RECALL ARE IN BOLD.

	RGB			RGBD		Depth only		
Method	[16]	[27]	[24]	[40]†	[28]†	[8] †	Ours	Ours+MCS
ape	2.48	15.8	9.60	49.8	80.7	**81.4**	64.6	64.4
can	17.5	63.3	45.2	51.2	88.5	94.7	95.4	**97.7**
cat	0.67	16.7	0.93	34.9	57.8	55.2	60.5	**63.9**
driller	7.66	65.6	41.4	59.6	94.7	86.0	94.7	**95.7**
duck	1.14	25.2	19.6	65.1	74.4	**79.7**	61.9	62.6
eggbox*	-	50.2	22.0	39.6	47.6	65.6	**89.1**	88.9
glue*	10.1	49.6	38.5	23.3	73.8	52.1	77.6	**78.7**
hole p.	5.45	39.7	22.1	67.2	**96.3**	95.5	88.1	91.1
MEAN	6.42	40.8	24.9	48.8	76.7	76.3	79.2	**80.6**

Table III

RESULTS ON FLIGHTGOGGLES POSE DATASET.

Method	Input modality	Recall (%)
DenseFusion [25]	RGBD	86.5
Ours w/o PCC	Depth	76.7
Ours	Depth	93.8

Table IV

ABLATION STUDIES ON OCCLUSION LINEMOD. ADDITIONAL RUNTIME OF PRUNING METHOD IN BOLD. SF: SEGNET AND 3D FCN.

ID	Experiment settings	Recall (%)	Runtime (ms) (SF + **Pruning**)
a	w/o PCC	54.8	133
b	w/ PCC but w/o pruning	56.4	133+**37**
c	Our full pipeline	79.2	133+**37**

Table V

COMPARISON OF HYPOTHESIS PRUNING METHODS ON OCCLUSION LINEMOD. ADDITIONAL RUNTIME OF PRUNING METHOD IN BOLD. SF: SEGNET AND 3D FCN.

ID	Method	Recall (%)	Runtime (ms) (SF + **Pruning**)
a	w/o PCC	54.8	133
b	RANSAC 1K	77.7	133+ **45**
c	RNASAC 10K	77.9	133+**248**
d	MCS	77.9	133+ **63**
e	Ours	79.2	133+ **37**
f	Ours+MCS	80.6	133+ **96**

D. Runtime

Given a 480×640 depth image, it takes 1 ms to load the data, 12 ms for segmentation, 118 ms for forward propagation of a single object as we sample at most 1500 points, and 37 ms for constructing a compatibility graph and running GCN which results in a total runtime of around 170 ms for a single pass on one single object.

V. CONCLUSIONS

This work proposes an end-to-end trainable deep neural network model to address challenging 6D pose estimation problem under heavy occlusion. To overcome the fragility of segmentation and 3D learning module on depth measurements, we enforce pairwise compatibility of 3D geometric features by applying spectral convolutions on a pairwise compatibility graph. Through detailed experiments, we demonstrate that our proposed algorithm outperforms existing methods on LineMOD and Occlusion LineMOD datasets. We additionally verify this algorithm on a synthetic dataset with large affine changes. Future work includes improving the predictions from localization and 3D features and potentially extending to 6D pose estimation of multiple instances by replacing the SegNet with an instance segmentation network. Furthermore, the 3D geometric features can also be replaced with a graph neural network to further increase the efficiency.

REFERENCES

[1] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.

[2] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "Rt3D: Real-time 3-D vehicle detection in lidar point cloud for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3434–3440, 2018.

[3] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.

[4] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3212–3217.

[5] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[6] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[7] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *2010 IEEE computer society conference on computer vision and pattern recognition*. Ieee, 2010, pp. 998–1005.

[8] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *European conference on computer vision*. Springer, 2016, pp. 834–848.

[9] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 195–205.

[10] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8958–8966.

[11] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1802–1811.

[12] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1329–1335.

[13] A. Krull, E. Brachmann, F. Michel, M. Ying Yang, S. Gumhold, and C. Rother, "Learning analysis-by-synthesis for 6D pose estimation in RGB-D images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 954–962.

[14] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3D point cloud matching with smoothed densities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.

[15] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3828–3836.

[16] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[17] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3D object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[18] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2011–2018.

[19] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[20] P. Guerrero, Y. Kleiman, M. Ovsjanikov, and N. J. Mitra, "PCPNet learning local shape properties from raw point clouds," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 75–85.

[21] H. Deng, T. Birdal, and S. Ilic, "PPF-Foldnet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 602–618.

[22] Z. J. Yew and G. H. Lee, "3DFeat-Net: Weakly supervised local 3D features for point cloud registration," in *European Conference on Computer Vision*. Springer, 2018, pp. 630–646.

[23] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[24] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[25] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3343–3352.

[26] H. Su, C. R. Qi, Y. Li, and L. 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*, 2015, pp. 2686–2694.

[27] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[28] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother, "Global hypothesis generation for 6d object pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 462–471.

[29] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[30] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.

[31] H. Yang and L. Carlone, "A polynomial-time solution for robust registration with extreme outlier rates," *arXiv preprint arXiv:1903.08588*, 2019.

[32] A. P. Bustos, T.-J. Chin, F. Neumann, T. Friedrich, and M. Katzmann, "A practical maximum clique algorithm for matching with pairwise constraints," *arXiv preprint arXiv:1902.01534*, 2019.

[33] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," *arXiv preprint arXiv:1905.11377*, 2019.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[35] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[36] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," *arXiv preprint arXiv:1904.08755*, 2019.

[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 548–562.

[41] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *European Conference on Computer Vision*. Springer, 2014, pp. 536–551.

[42] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 699–715.

[43] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, M. M. A. Patwary, and M. Ali, "A fast parallel maximum clique algorithm for large sparse graphs and temporal strong components," *CoRR, abs/1302.6256*, 2013.

[44] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.