OLF : RGB-D Adaptive Late Fusion for Robust 6D Pose Estimation

Petitjean Théo, Zongwei Wu, Cédric Demonceaux, Olivier Laligant

OLF : RGB-D Adaptive Late Fusion for Robust 6D Pose Estimation

Petitjean Théo[a,b], Zongwei Wu[a], Cédric Demonceaux[a], and Olivier Laligant[a]

[a]ImViA, Université de Bourgogne, Dijon, France
[b]Spie, Belfort, France

ABSTRACT

RGB-D 6D pose estimation has recently gained significant research attention due to the complementary information provided by depth data. However, in real-world scenarios, especially in industrial applications, the depth and color images are often more noisy[1,2] Existing methods typically employ fusion designs that equally average RGB and depth features, which may not be optimal. In this paper, we propose a novel fusion design that adaptively merges RGB-D cues. Our approach involves assigning two learnable weight α_1 and α_2 to adjust the RGB and depth contributions with respect to the network depth. This enables us to improve the robustness against low-quality depth input in a simple yet effective manner. We conducted extensive experiments on the 6D pose estimation benchmark and demonstrated the effectiveness of our method. We evaluated our network in conjunction with DenseFusion on two datasets (LineMod[3] and YCB[4]) using similar noise scenarios to verify the usefulness of reinforcing the fusion with the α_1 and α_2 parameters. Our experiments show that our method outperforms existing methods, particularly in low-quality depth input scenarios. We plan to make our source code publicly available for future research.

Keywords: Late fusion, Deep learning, Self Optimized parameter, PSNR and noise study,RGB-D

1. INTRODUCTION

6D pose estimation is crucial for implementing computer vision in real-world and industrial problems, such as robotic manipulation[6] and augmented reality.[2] Solutions must be robust to object texture and geometry, occlusion, and external constraints. However, expensive RGB-D sensors are currently used in the industry to limit noise, limiting access to cheaper sensors while demand for vision applications grows. Deep learning has led to more modern research in camera pose estimation, with direct regression through the network[5] being a simple and applicable approach. Some methods use DNNs and the Perspective-n-Point algorithm (PnP),[7] but suffer from problems transitioning from 2D to 3D space and loss of geometric information. Traditional methods[8,9] based on feature extraction and mapping to model clouds were effective but limited in performance due to changing scene brightness and occlusion.

The development of inexpensive and easily usable RGB-D sensors has enabled the creation of datasets carrying depth information, highlighting the importance of 3D and depth information in pose detection. This modality has improved processes used on 2D, whether in direct regression[5] or based on keypoints.[7] Methods using multiple modalities have highlighted the importance of RGB and Depth information, leading to input data fusion strategies. However, most fusion methods[5,10] simply concatenate the information, potentially propagating noise throughout the network. To address this, we propose a method using an external parameter to enhance the network's resistance to input noise. Our method's advantages are described by addressing two points:

- How to adapt a well-known fusion network (DenseFusion[5]) and to make it reliable even under extreme noise conditions.
- How much noise can impact classical networks and how much state-of-the-art metrics can be degraded by noise.

The smart grid will be driven according to the following scenario: the training data will be considered perfect and under controlled laboratory conditions. The test data will be noisy to get closer to potential complex conditions encountered in industrial environments.

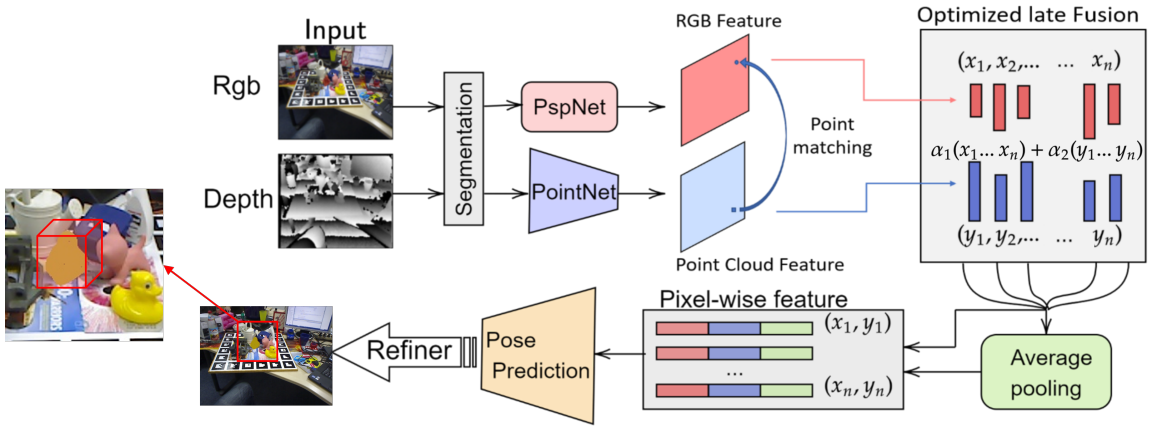

Figure 1. **Global overview of our Optimized late fusion** Our method is based on the DenseFusion[5] Architecture : after segmentation, Features are extracted from RGB and Depth and are then fused by our adaptive Fusion Module. The features generated by this fusion are then used by a pose predictor and a refiner. The pose Predictor predicts pose for each pixel and the predictions are voted to generate the final pose.

2. RELATED WORK

Given an RGBD image, the goal of 6D pose estimation is to predict the matrix that transforms the object coordinates to the camera coordinates.

Pose with RGB only. The traditional methods were essentially based on point of interest detection techniques.[1,11] These techniques already used in computer vision allowed to create correspondence between two images. Some methods sought to learn to predict 2D keypoint and determine the position using PnP algorithm.[12] The arrival of methods using deep learning allowed to outperform the classical methods particularly sensitive to the quality of the images and to the texture of the objects. These new methods based on CNN have allowed the emergence of networks that directly predict the pose from RGB images.[13] However, despite good results, 3D prediction remains a problem, especially because of the difficulty of back-projecting 2D points in 3D space. Some methods such as Xiang et al.[14] have tried to tackle this problem by taking advantage of the 3D models of the searched objects by producing 3D features from them. Mousavian et al.[15] have tried to predict 3D parameters using geometric constraints.

Pose with 3D point cloud. Some studies have also tried to solve the problem of pose estimation with methods based on voxel and point cloud studies. The traditional methods[16] were very time consuming. More recently, methods based on deep learning have allowed to directly predict the pose from networks adapted to 3D point clouds such as frustrumPointNet,[17] PointNetLike,[18] VoxelNet[19] etc ... These methods although very robust on urban driving application dataset, remain limited in the problem of 6D pose estimation. The sparsity and lack of texture in the point clouds limit the pose estimation accuracy. Moreover, objects with high reflectivity are an almost inescapable problem : active depth sensors are unable to create clouds on these particular surfaces.

Pose with RGB-D. Similar to the classical RGB methods, the classical RGB-D methods are based on two-step algorithms. The 3D features extracted from the input are therefore used by matching and verification techniques.[20,21] These methods are nevertheless sensitive to the change of brightness in the image.RGB-D methods can be categorized on three different approaches. Holistic methods based on Deep Neural Network (DNN) like DenseFusion[5] or PoseCNN,[22] seek to regress the position directly. To circumvent the problem induced by the non-linearity these methods use refining loops to estimate the position. Dense correspondence methods,[23]using a Hough voting scheme to vote for the final results with either random forest[20] and CNN.[24] KeyPoint based methods are also very present in RGB-D, works such as PVnet,[25] or PVN3D.[7]

Fusion. Many methods based on RGB-D images[17] choose to extract features from the point cloud and the RGB image and then merge them to improve the robustness and efficiency of their network. As also shown by DenseFusion,[5] PointFusion,[10] MoreFusion,[26] late fusion strategies are efficient and allow a better understanding of input data. However the concatenation strategy seems to propagate the noise inside the network. Some more modern methods have underlined the impact that a badly optimized fusion could have on a network : new strategies have been developed like in FFB6D.[27] With this in mind, and in order to improve the fusion while making it as insensitive to noise as possible, we propose in this work a new late fusion method balanced by two coefficient parameters(α_1 and α_2).

3. PROPOSED METHOD

With an input of type RGB-D, the 6D pose estimation seeks to predict a transformation matrix $[R|t]$ which transforms the object model from these coordinates to the camera space coordinate. The accuracy of pose estimation is highly dependent on the quality of the RGB and Depth data. Thus we would like to take into account the quality of the texture and geometry extracted from the RGB-D camera in order to improve the accuracy of the camera pose estimation.

3.1 Overview

We therefore propose, in order to limit the propagation of noise from the input to the whole network, a smart fusion allowing the network itself to give a weight to each modality. The model works as follows : first we extract the features for each object location. More concretely, two networks will extract features from each modality in parallel: on the one hand the RGB on which a CNN is applied and on the other hand a network similar to PointNet[18] on the depth. Then these features are merged in a so-called late fusion strategy. This fusion based on DenseFusion[5] assemble the features previously extracted from each network, and output a pixel-wise feature. This fusion is balanced by the network itself with the help of an additional parameter α, one parameter is linked to Rgb (α_1) and the other one to the depth (α_2). The purpose of this coefficient is to give confidence to information extracted from one of the two data sources according to their quality and/or relevance. Then the features are used by a network that predicts the position of the targeted object. Finally a refining network iteratively improves the results.

3.2 Global architecture

In this section we will see the different steps necessary to predict the final pose, from the initial segmentation, to the feature extraction.

Semantic Segmentation. In the first pre-processing step, the objects are segmented. The segmentation network generates $N+1$ segmentation maps. Each mask then generated is a binary mask whose class corresponds to the N objects of the dataset.

Feature extraction. Feature extraction is a primordial step in a pose estimation algorithm. We extract the features from the color and depth, in order to take advantage of the RGB-D data. As proposed in DenseFusion, we first extract a point cloud from the previously segmented depth map.

Then, with an architecture similar to PointNet, the geometric information is extracted from the generated point cloud. The features extracted from the depth map would best characterize the geometric information of the scene. In the same way a CNN encoder-decoder network will try to extract the information from the RGB image.

3.3 Adaptive Late Fusion

Once the features are extracted from each modality, we want to find a method limiting the impact of noise on the quality of the information transmitted in the network afterwards. Based on a late fusion technique, massively used in the state-of-the-art, we modify the commonly used strategy of information concatenation. This fusion strategy is massively used in several applications such as PVN3D[7] and DenseFusion (DF).[5] Although PVN3D outperforms DenseFusion, we chose to use DenseFusion because its direct regression prediction method in the network allows a better observation of the influence of the fusion on the final result.

As shown in Figure 1, the objective of the fusion is to create balancing α parameters : the network balances the weight of information for each modality on its own. This additional parameter in the PixelWiseFusion stage[5] allows, in addition, better occlusion robustness (induced by the method of Wang et al.), outlier filtering and reducing noise resistance (moreover, the concatenation strategy initially used in the DenseFusion method induces a propagation of the noise potentially present in each input modality). The geometrical and color features are then merged through a pooling strategy balanced by the α_1 and α_2 parameters. As proposed in prior work we keep, afterwards, an average-pooling strategy in order to add in addition to the geometrical and color features a parameter representing the global appearance of self balancing with α.

It is important to note that the merged features are injected into both the pose prediction network and the refinement network equally. The entire original network benefits from the noise resistance brought by our Optimized Late Fusion method.

3.4 Pose estimation and refinement

Once the fusion is done, we use a classical scheme of the state of art for position estimation. This estimation is done in two steps. First the network predicts for each point a rotation R and a translation t balanced by confidence C. The specificity of the method, initially proposed by dense fusion, is to minimize the Euclidean distance calculation for all the points and to choose the most efficient prediction, having the highest confidence.

$$L_i^p = \frac{1}{M} \sum_j \left\| (Rx_j + t) - (\hat{R}_i x_j + \hat{t}_i)) \right\| \tag{1}$$

Where x_j is the j^{st} randomly selected 3D point wrote as M from the object model. The two values R and t are the prediction p. $\hat{R}$ and $\hat{t}$ are the GT values known thanks to the dataset annotations. This distance calculation is well adapted for asymmetrical objects. For symmetrical objects, we minimize the distance to get the closest one with the model of points, removing a part of the problem of the rotation. In order to minimize the distance we replace $\sum_i$ from equation (1) by $\sum_i \min_{0<k<M}$

The Loss optimized by the network is the one proposed by DenseFusion:

$$L = \frac{1}{N} \sum_i (L_i^p c_i - \omega log(c_i)) \tag{2}$$

Where N is the randomly sampled fused dense-pixel from the P element of the initial segmentation and ω is the balancing hyper-parameter.

4. EXPERIMENTS

We simulate scenarios of more or less intense noise in order to verify how the fusion parameterized by α_1 and α_2 allows to minimize the impact of the noise latter on the network. Noise is alternatively (or both) added to RGB images or (and) to depth images. This will allow us to verify which noise has the most impact on the network and which modality has the most impact on the final results. For the RGB image we add a additive Gaussian white noise (GWN) whose strength will be fixed by its standard deviation σ_{noise}. The depth information is corrupted by a multiplicative GWN in order to avoid creating unwanted information on the 0 returned by the physical sensor. To quantify the noise level in our experiments, we use the SNR for each information channel (RGB ou depth). The closer the SNR value will be to 0 (or negative value) is, the stronger the noise is.

Our experiments are based on the DenseFusion algorithm. While this approach is no longer the best approach of the state of the art, it allows to easily demonstrate the impact on the proposed fusion stage. Nevertheless, our approach could be integrated in any other method using the two modalities (RGB and D). All the experiments presented here are obtained with 2 refinement loops.

	RGB+hN & D+hN		RGB & D+hN		RGB+hN & D		RGB+N & D+N		NoNoise	
Network Type	Df+R	OLF+R	Df+R	OLF+R	Df+R	OLF+R	Df+R	OLF+R	DF	OLF
ape	**00,9**	00,7	**01,0**	00,6	90,1	**93,6**	63,4	**78,6**	92,3	**94,0**
bench vi.	11,4	**22,9**	11,3	**21,7**	91,4	**92,3**	90,6	**91,0**	**93,2**	92,5
camera	02,6	**02,9**	02,3	**03,2**	**94,7**	94,0	88,9	**91,6**	94,4	94,4
can	04,6	**13,6**	04,2	**12,8**	92,4	**96,2**	90,7	**94,9**	93,1	**96,6**
cat	01,3	**03,8**	01,2	**04,9**	95,8	**96,6**	91,2	**94,7**	96,5	**97,0**
driller	01,3	**08,4**	01,9	**08,1**	85,0	**92,4**	83,0	**91,1**	87,0	**92,2**
duck	00,9	**01,0**	**00,5**	00,4	91,4	**95,9**	76,3	**85,4**	92,3	**96,2**
eggbox	35,3	**48,3**	36,4	**45,9**	99,8	99,6	99,8	99,8	99,8	**99,9**
glue	**41,4**	26,9	40,1	28,5	**99,8**	99,5	**99,6**	99,4	**100,0**	99,8
hole p.	01,2	**03,3**	01,9	**03,9**	**88,9**	88,3	78,8	**82,3**	**92,1**	88,2
iron	18,5	**19,2**	18,1	**18,6**	**97,5**	96,9	**96,9**	96,1	97,0	97,0
lamp	01,6	**08,4**	01,7	**09,5**	95,6	**96,2**	94,3	**95,5**	95,3	**96,0**
phone	**12,0**	11,7	11,4	**13,3**	92,5	**95,7**	90,8	**94,3**	92,8	**95,8**
AVERAGE	10,3	**13,3**	10,2	**13,2**	93,4	**95,2**	88,0	**92,0**	94,3	**95,3**

Table 1. Results of LineMOD under different noise scenario, with OLF(our)+(Refinement) and DF+(Refinement), hN mean high noise and N noise.

4.1 DataSet

LineMod dataset. The LineMOD dataset proposed by Hinterstoisser et al.[3] is the main benchmark used to compare 6D pose estimation methods. It consists of a series of videos and indoor images including 13 objects with a low texture and different lighting level.

YCB dataset. The YCB-Video dataset Xiang et al.[22] characterizes 21 objects from the YCB Calli et al.[4] database and being of different texture, size, and shape. The dataset contains a total of 92 videos with depth and RGB modality, in staging inside. These videos also include the annotations necessary for the 6D pose estimation work. We follow the previous work[22] regarding to the test and training set splitting.

4.2 Evaluation Metrics

The results of our network are evaluated by two metrics proposed in previous works.[22] We therefore use the ADD metrics for non-symmetric objects and ADD-S for symmetric objects. ADD is defined as follow :

$$ADD = \frac{1}{m}\sum_{i \in o}\left\|(R_i + t) - (\hat{R}_i + \hat{t}))\right\| \tag{3}$$

where i denotes a vertex in object o, and R, t the predicted pose and $\hat{R}, \hat{t}$ the ground truth. And for symmetric object we calculate ADD-S such as:

$$ADD\text{-}S = \frac{1}{m}\sum_{i \in o} \underset{j \in o}{min}\left\|(R_i + t) - (\hat{R}_j + \hat{t}))\right\| \tag{4}$$

According to the work of,[22] we evaluate the results of YCB DataSet by reporting the area under the ADD-S curve (AUC) and set the maximum threshold to be 0.1m. We also report the percentage of ADD smaller than 2cm.

4.3 LineMOD

LineMOD evaluation. The results are given in several parts. First, as shown in Table 1, we compare our network to classical DenseFusion approach re-trained from scratch for fair comparison. DF_{Ori} is the orginal DenseFusion method, DF_{rt} means that the fusion strategies has been handle by using a simple addition in the fusion bloc and OLF is our method where this fusion bloc takes into account the α parameters. Our method outperforms both DenseFusion strategies.

Robustness toward LineMOD+Noise. This is the core of our contribution. Introducing α_1 and α_2 parameter produces a method less sensitive to noises which could be present during the inference phase. This test is extremely important because in practice the input data are often noisy, especially depth modality in industrial environments. Thus, we consider the network is trained on exact data (LineMod) and the noise is added in the inference images. The results of the different noise scenarios can be seen in Table 1. The objective

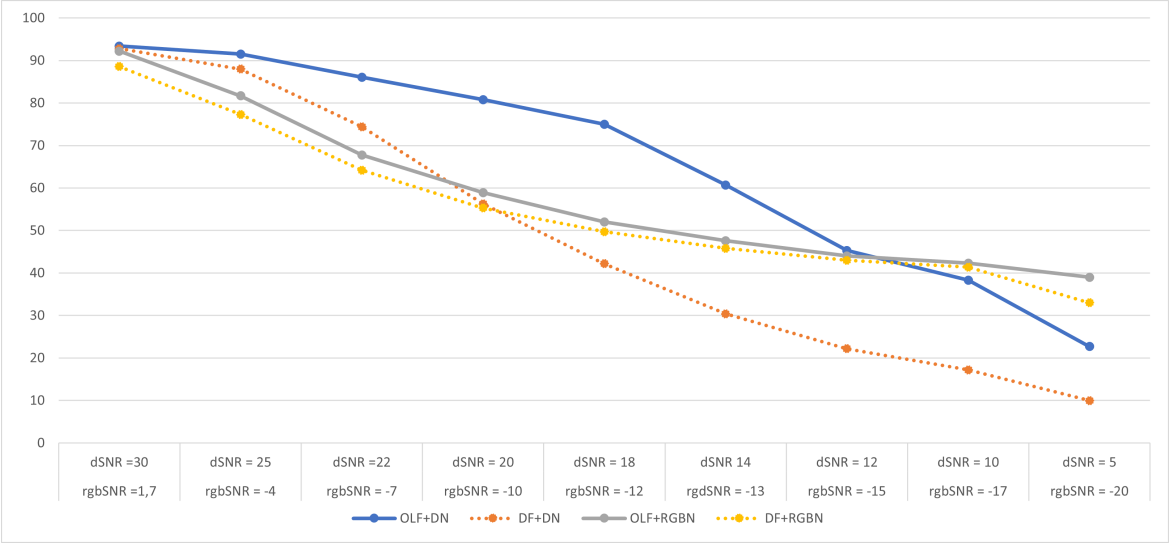

Figure 2. DenseFusion and OLF comparison related to noise evolution in RGB and depth

is both to identify the most problematic noise for the network and to verify that the Optimized Late Fusion is better than the classical fusion method. The noise strength is known through the SNR value estimated for each inputted σ.

- **RGB+hN** & **D+hN**: This case is the most noisy of all, here we apply a strong white noise on both modalities at the same time, that give us a SNR_d of 9 for Depth and -15 for SNR_{RGB}. We notice that in these conditions, the network is very poorly performing, on each type of objects. Nevertheless, we can observe that the OLF network performs better than the initial network by 3%
- **RGB** & **D+hN**: Here we only add a white noise on the depth modality with a equivalent SNR_d of 8. We can notice a strong degradation of the network performance. Nevertheless, OLF remains more resistant than DF in this noise condition.
- **RGB+hN** & **D**: This particular case is interesting because despite a strong noise applied on RGB with $SNR_{RGB} = 1.7$, the network keeps a very convincing performance or even almost unchanged. This result can be explained in part by the fact that the network gives more weight to the depth. To degrade the results in a similar way with noise only in RGB images, it would be necessary to apply an extremely strong noise to these images (see figure 2).
- **RGB+N** & **D+N**: Here a white noise of slightly reduced strength is applied on the two modalities ($SNR_d = 25$, $SNR_{RGB} = 7$) . This scenario is probably the closest to reality and the most interesting in our case. The Optimized Late Fusion performs much better than the original network. The resistance to noise brought by this new fusion method is felt on almost all objects, and globally the average success is improved by 4% on all test images.

Evolution of average on LineMod. Figure 2 shows the behavior of the average success of the DenseFusion network and the OLF network with growing noises. The noise is added with a growing value, allowing us to see the evolution of the percentage of success of the network as a function of the noise. The noise is, for this experiment, added either on RGB or on D, but never simultaneously.

The two curves ($OLF + DN$ and $DF + DN$) present the evolution of the average results obtained when the depth images are noisy. Our method dealing with noisy depth ($OLF + DN$) is more efficient than the original method whatever the intensity of the noise added. We can see that the methods is very relevant for Depth and

	DF_{ori}		OLF(ours)	
nonoise	AUC	$< 2cm$	AUC	$< 2cm$
	93.1	96.8	93.8	97.2
$SNR_D = 39$	AUC	$< 2cm$	AUC	$< 2cm$
$SNR_{rgb} = 6$	57.4	60.2	63.4	67.1
$SNR_D = 33$	AUC	$< 2cm$	AUC	$< 2cm$
$SNR_{rgb} = 3$	37.6	40.1	44.6	51.1

Table 2. Quantitative evaluation of 6D pose (AUC and ADD(2cm)) on the YCB data + Noise.

less for RGB, it's simply due to the methods itself, much more sensible to depth variation than to RGB variation. Inded when RGB is completly destroyed we can still perfom 6d pose estimation in 40 % of the cases.

$OLF + RGBN$ and $DF + RGBN$ show in an identical way, the evolution of the average precision according to the intensity of the noise applied on the RGB images. For higly noisy RGB images (SNR - 20) the curve stabilises around 38%, which means that once the RGB is fully unusable, depth achieves to recover 38% of the poses. The OLF fusion allows to increase the resistance to noise in all the tested scenarios. We can conclude from these series of tests on the LineMod, that our fusion method has well strengthened the original methods against the noise.

4.4 YCB dataset

In order to more illustrate the robustness of our method, we tested its performances on the YCB datest with high levels of noise. We choose two different white noise values. The SNR are different from LineMod because the image ranges are not the same. noise is added in YCB images jointly on both D and RGB.

Table 2 presents some results. The OLF method is slightly better than the DenseFusion in both ADD-s 2cm and AUC metrics. In presence of white noise, the OLF method strongly outperforms the original method. For $SNR = 39$ for example, a gap of 7% between the two methods is observed. These experiments confirm the robustness of our approach compared to DenseFusion.

4.5 Experiments under different noise Scenarios

We conducted experiments to evaluate the effectiveness of our fusion method in reducing the impact of noise on the network. We added noise to either RGB images, depth images, or both, and measured the impact of each type of noise on the final results. Gaussian white noise (GWN) was added to the RGB images, and multiplicative GWN was added to the depth information to avoid unwanted information on the 0 returned by the physical sensor. We used the signal-to-noise ratio (SNR) to quantify the noise level in our experiments, with a lower SNR indicating stronger noise.

The two curves ($OLF + DN$ and $DF + DN$) present the evolution of the average results obtained when the depth images are noisy. Our method dealing with noisy depth ($OLF + DN$) is more efficient than the original method whatever the intensity of the noise added.

$OLF + RGBN$ and $DF + RGBN$ show, in an identical way, the evolution of the average precision according to the intensity of the noise applied on the RGB images.

5. CONCLUSION

In this paper, we have proposed an optimized fusion method based on DenseFusion. Even if DenseFusion is working well, we have shown that this method is very sensitive to noise which makes our method more efficient for real application where data (especially depth data) are often noisy. We have shown here that a simple modification of the fusion module which takes into account the accuracy of the modalities improves the results and the robustness. These revisited DenseFusion-like methods are then more efficient and usable in real scenario. Many neural network methods of the state of the art as (PVN3D,[7] FFB6D,[27] ...) use a fusion module which concatenate RGB and Depth information and are good candidates for this new fusion methods. As our Optimized Late Fusion module is generic, it can be also added in these neural network methods to improve their robustness toward noise.

REFERENCES

[1] Collet, A., Martinez, M., and Srinivasa, S. S., "The moped framework: Object recognition and pose estimation for manipulation," *IJRR* **30**(10), 1284–1306 (2011).

[2] Marder-Eppstein, E., "Project tango," in [*ACM SIGGRAPH 2016 Real-Time Live!*], 25–25 (2016).

[3] Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., and Lepetit, V., "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in [*ICCV*], (2011).

[4] Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., and Dollar, A. M., "The ycb object and model set: Towards common benchmarks for manipulation research," in [*ICRA*], (2015).

[5] Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., and Savarese, S., "Densefusion: 6d object pose estimation by iterative dense fusion," in [*CVPR*], (2019).

[6] Zhu, M., Derpanis, K. G., Yang, Y., Brahmbhatt, S., Zhang, M., Phillips, C., Lecce, M., and Daniilidis, K., "Single image 3d object detection and pose estimation for grasping," in [*ICRA*], (2014).

[7] He, Y., Sun, W., Huang, H., Liu, J., Fan, H., and Sun, J., "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in [*CVPR*], (2020).

[8] Rios-Cabrera, R. and Tuytelaars, T., "Discriminatively trained templates for 3d object detection: A real time scalable approach," in [*ICCV*], (2013).

[9] Wohlhart, P. and Lepetit, V., "Learning descriptors for object recognition and 3d pose estimation," in [*CVPR*], (2015).

[10] Xu, D., Anguelov, D., and Jain, A., "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in [*CVPR*], (2018).

[11] Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J., "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *IJCV* **66**(3), 231–259 (2006).

[12] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**(6), 381–395 (1981).

[13] Tulsiani, S. and Malik, J., "Viewpoints and keypoints," in [*CVPR*], (2015).

[14] Xiang, Y., Choi, W., Lin, Y., and Savarese, S., "Data-driven 3d voxel patterns for object category recognition," in [*CVPR*], (2015).

[15] Mousavian, A., Anguelov, D., Flynn, J., and Kosecka, J., "3d bounding box estimation using deep learning and geometry," in [*CVPR*], (2017).

[16] Song, S. and Xiao, J., "Sliding shapes for 3d object detection in depth images," in [*ECCV*], (2014).

[17] Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J., "Frustum pointnets for 3d object detection from rgb-d data," in [*CVPR*], (2018).

[18] Qi, C. R., Su, H., Mo, K., and Guibas, L. J., "Pointnet: Deep learning on point sets for 3d classification and segmentation," in [*CVPR*], (2017).

[19] Zhou, Y. and Tuzel, O., "Voxelnet: End-to-end learning for point cloud based 3d object detection," in [*CVPR*], (2018).

[20] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., and Rother, C., "Learning 6d object pose estimation using 3d object coordinates," in [*ECCV*], (2014).

[21] Lowe, D. G., "Object recognition from local scale-invariant features," in [*ICCV*], (1999).

[22] Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D., "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in [*RSS*], (2018).

[23] Liebelt, J., Schmid, C., and Schertler, K., "independent object class detection using 3d feature maps," in [*CVPR*], (2008).

[24] Doumanoglou, A., Kouskouridas, R., Malassiotis, S., and Kim, T.-K., "Recovering 6d object pose and predicting next-best-view in the crowd," in [*CVPR*], (2016).

[25] Peng, S., Liu, Y., Huang, Q., Zhou, X., and Bao, H., "Pvnet: Pixel-wise voting network for 6dof pose estimation," in [*CVPR*], (2019).

[26] Wada, K., Sucar, E., James, S., Lenton, D., and Davison, A. J., "Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion," in [*CVPR*], (2020).

[27] He, Y., Huang, H., Fan, H., Chen, Q., and Sun, J., "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in [*CVPR*], (June 2021).