6D Pose Estimation with Two-stream Net

Xiaolong Yang
KLMM, AMSS, CAS and Univ. of CAS
yangxiaolong17@mails.ucas.ac.cn

Xiaohong Jia*
KLMM, AMSS, CAS and Univ. of CAS
xhjia@amss.ac.cn

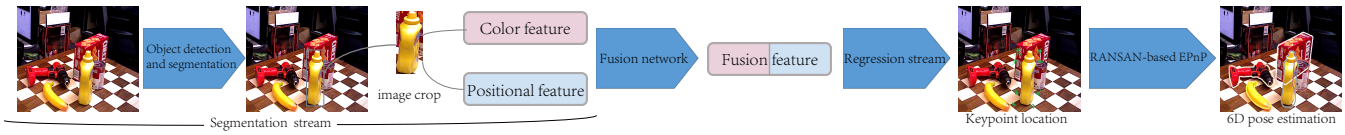

Figure 1: Overall workflow of our method. We design a two-stream architecture for segmentation and regression, a fusion network for couple texture and positional feature, and an end-to-end iterative pose refinement procedure for better result.

ABSTRACT

In this poster, we present a heterogeneous architecture for estimating 6D object pose from RGB images. First, we use a two-stream network to extract robust 3D-to-2D embedding feature correspondence. The segmentation stream processes the RGB information and spatial features individually. Then, we construct another fusion network to couple color and positional features, and predict the locations of keypoints in the regression stream. The pose can be obtained by an efficient RANSAC-based PnP algorithm. Moreover, we design an end-to-end iterative pose refinement procedure that further improves the reliable pose estimation. Our method outperforms state-of-the-art approaches in two public datasets.

KEYWORDS

pose estimation, two-stream network, fusion network

ACM Reference Format:
Xiaolong Yang and Xiaohong Jia. 2020. 6D Pose Estimation with Two-stream Net. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '20 Posters), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388770.3407423

1 INTRODUCTION

6D object pose estimation is crucial to many applications. Ideally, the solution should handle objects with deforming shapes and textures, be robust to severe occlusion, sensor noise, and changing lighting conditions, as well as achieve the real-time speed.

Traditional RGB methods may fail when the object is featureless or the scene is blocked by multiple objects. Recent deep-learning-based approaches estimate the 6D pose or detect key points directly in an end-to-end manner. However, in both cases, the object is still considered a global entity, which makes the algorithm vulnerable to greater occlusion.Recently, SegDriven [Hu et al. 2019] relies on a combination of local pose prediction , SilhoNet [Billings and Johnson-Roberson 2019] uses the contour to return the orientation of object. Pix2Pose [Park et al. 2019] uses GAN to estimate the 3D coordinates of each pixel to form a 2D-3D correspondence. However, these works still face the problem of low accuracy. ACC [Castro et al. 2019] uses a fully optimized DCNN model, which can reconstruct the mesh, but can only handle a single object, and cannot solve complex scenes. The goal of our work is to recognize 6D poses on RGB images even in the case of occlusion, with real-time performance.

2 PROPOSED METHOD

Inspired by [Hu et al. 2019] and [Wang et al. 2019], we design a two-stream architecture, a fusion network as shown in Fig. 1, and an additional end-to-end iterative pose refinement procedure.

The proposed two-stream net has segmentation stream and regression stream. For the segmentation stream, we use the Darknet-53 architecture of YOLOv3 and segment each detected object in the original input image. The role of the segmentation stream is to assign a label to each cell of the virtual $n \times n$ grid superposed on the image. More precisely, given m object classes, this outputs a feature vector P of dimension $m + 1$ with an additional dimension to account for the background.

Next, we design a fusion network to complement the above feature vector and corresponding color information. The key idea of our fusion network is to perform local per-grid fusion instead of global fusion so that we can make predictions based on each fused feature. We use known camera intrinsic parameters and ground truth with texture to associate the spatial position features of each grid with their corresponding pixel texture features based on the projection on the image. The obtained feature pairs are then connected and generate a fixed-size global feature vector.

The purpose of the regression stream is to predict the 2D projections of 3D keypoints from the global feature vector. Typically, we take these keypoints as be the 8 corners of the model bounding boxes. That is, let p be the 2D location of a grid cell center. For the i^{th} keypoint, we seek to predict an offset $f_i(p)$, such that the resulting location $p + f_i(p)$ is close to the ground-truth 2D location q_i. Similarly, for the color of the center c , offset $f_i(c)$ and ground-truth texture color t_i, we have

$$\Delta_i(p) = p + f_i(p) - q_i, \qquad \Delta_i(c) = c + f_i(c) - t_i \tag{1}$$

*Corresponding author

SIGGRAPH '20 Posters, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7973-1/20/08.
https://doi.org/10.1145/3388770.3407423

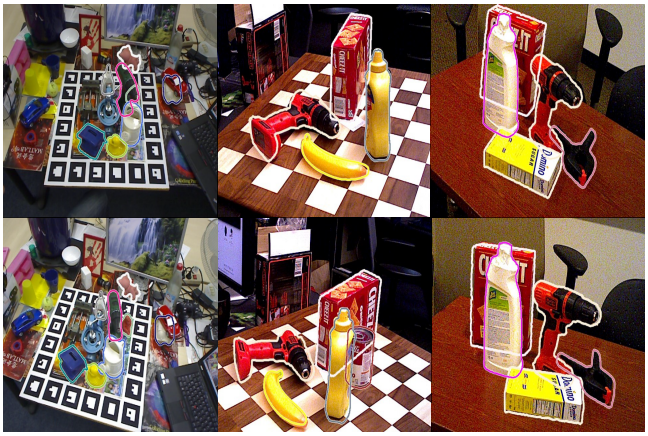

Figure 2: Visual experiment result. Each column is a pair of results with different perspectives and shows the accurate performance even when the object is blocked.

and by defining the loss function

$$E_{pos} = \sum_{Grid} \sum_{i=1}^{8} \|\Delta_i(p)\|_1 + \|\Delta_i(c)\|_1 , \tag{2}$$

where $\|\cdot\|_1$ denotes the L_1 loss function, which is less sensitive to outliers than the L_2 loss. Because both the position and color information can be considered as 3D information (XYZ and RGB), which have equal influence.

The regression stream also outputs a confidence value Con_i for each predicted keypoint, which is obtained via a sigmoid function on the network output. These confidence values should reflect the proximity of the predicted 2D projections to the ground truth. To encourage this, we define a second loss term

$$E_{pro} = \sum_{Grid} \sum_{i=1}^{8} \left\| con_i - exp(-\tau \, \|\delta_i(p) + \delta_i(c)\|_2 \right\|_1 , \tag{3}$$

where τ is a modulating factor. Then, the regression loss becomes

$$E = \alpha E_{pos} + \beta E_{pro}, \tag{4}$$

where α and β modulate the influence of the two terms.

For each object, we use the confidence score predicted by the network to establish a 2D to 3D correspondence between the image and the 3D model. Considering efficiency, we find that 10 most confident predictions is a good balance between speed and accuracy. We use the RANSAC-based EPnP algorithm to obtain the 6D pose.

The key idea of iterative refinement is to consider the previously predicted pose as a projection transformation on the original 3D model and get the corresponding 2D image. The obtained 2D image is regarded as the result of the segmentation stream and fed back to the network to get a more accurate iteration refinement.

3 EXPERIMENTAL RESULTS

We evaluate our method on two challenging 6D object pose estimation datasets: Occluded-LINEMOD [Krull et al. 2015] and YCB-Video [Xiang et al. 2018]. We compare with state-of-the-art methods: PoseCNN, BB8, Tekin, Heatmaps, Pix2Pose, SilhoNet and Seg-Driven. We use the most commonly used ADD-S as a metric and ADD-0.1d means that assume the predicted pose to be correct if the ADD below 10% of the model diameter.

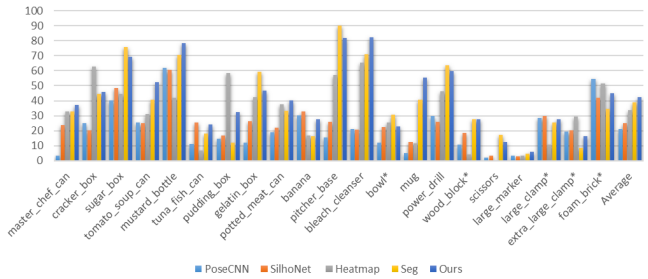

Figure 3: Quantitative evaluation of 6D pose (ADD-0.1d) on YCB. Objects with * are symmetric.

Our experiment results on two challenging datasets demonstrate that our approach outperforms the state of the arts on quantitative results and is able to achieve realtime performance. Some visual performances in the case of occlusion are shown in Fig. 2. However, results also indicate that there is still room for improvement on objects with symmetry, which will be the focus of our future work.

Table 1: Quantitative evaluation of 6D pose (ADD-0.1d) on Occluded-LINEMOD. Bold-name objects are symmetric.

Object	PoseCNN	Tekin	BB8	Pix2Pose	Heatmap	Seg	Ours
Ape	9.6	7.0	28.5	8.3	16.5	12.1	**29.1**
Can	45.2	1.2	11.2	12.1	42.5	39.9	**49.8**
Cat	0.9	3.6	**9.6**	9.3	2.8	8.2	6.9
Driller	41.4	1.4	0.2	10.9	47.1	45.2	**49.0**
Duck	19.6	5.1	6.8	6.3	11.0	17.2	**22.6**
Eggbox	22.0	9.6	4.0	13.8	**24.7**	22.1	11.9
Glue	38.5	6.5	4.7	11.3	**39.5**	35.8	16.5
Holepun	22.1	8.3	8.1	10.7	21.9	36	**53.1**
Average	24.9	5.3	9.1	10.3	25.8	27.0	**29.9**

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (61872354), Beijing Natural Science Foundation (Z190004), and Alibaba Group through Alibaba Innovative Research Program.

REFERENCES

Gideon Billings and Matthew Johnson-Roberson. 2019. SilhoNet: An RGB Method for 6D Object Pose Estimation. *IEEE Robotics and Automation Letters* 4, 4 (2019), 3727–3734.

Pedro Castro, Anil Armagan, and Tae-Kyun Kim. 2019. Accurate 6D Object Pose Estimation by Pose Conditioned Mesh Reconstruction. In *arXiv:1910.10653*.

Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. 2019. Segmentation-Driven 6D Object Pose Estimation. In *IEEE CVPR*. 3385–3394.

Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. 2015. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *IEEE ICCV*. 954–962.

Kiru Park, Timothy Patten, and Markus Vincze. 2019. Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *IEEE ICCV*. 7668–7677.

Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martin-Martin, Fei-Fei Li Cewu Lu, and Silvio Savarese. 2019. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *IEEE CVPR*. 3343–3352.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. 2018. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems*.