End-to-End Differentiable 6DoF Object Pose Estimation with Local and Global Constraints

Anshul Gupta, Joydeep Medhi, Aratrik Chattopadhyay, Vikram Gupta
Mercedes-Benz Research and Development India
firstname.lastname@daimler.com

Abstract

Inferring the 6DoF pose of an object from a single RGB image is an important but challenging task, especially under heavy occlusion. While recent approaches improve upon the two stage approaches by training an end-to-end pipeline, they do not leverage local and global constraints. In this paper, we propose pairwise feature extraction to integrate local constraints, and triplet regularization to integrate global constraints for improved 6DoF object pose estimation. Coupled with better augmentation, our approach achieves state of the art results on the challenging Occlusion Linemod dataset, with a 9% improvement over the previous state of the art, and achieves competitive results on the Linemod dataset.

1 Introduction

Estimating the 6DoF pose of an object is an important problem with applications in various domains like robotics [1], augmented reality [2] and autonomous driving [3]. With the pervasion of inexpensive RGB sensors, it is cost effective and highly beneficial to perform 6DoF pose estimation from a single RGB image without using additional depth sensors.

Some studies [4][5] attempted to regress the 6DoF pose directly from the image, however, these were not as competitive as recent two stage approaches. In the first stage of two stage approaches, a *correspondence estimator* detects the object and estimates the 2D image projections of the 3D object points (referred to as 2D keypoints). This establishes correspondences between the 2D and 3D points. [6][7][8] used a CNN based architecture to segment out regions containing the object and regress the 2D keypoints from those regions. A recent study regressed direction vectors to the 2D keypoints from the segmented regions of the object [9]. The 2D keypoints were then estimated from intersections of pairs of direction vectors. This approach was found to be more robust to occlusions of the object.

In the second stage, a RANSAC based Perspective-n-Point(PnP) algorithm serves as a *pose estimator* to predict the 6DoF object pose using the established 2D-3D correspondences. However, Hu et al. [10] showed that RANSAC is sensitive to the ordering of the 2D-3D correspondences and computationally costly when there are many of them. Further, the non-differentiable nature of the RANSAC based pose estimator does not allow for end-to-end training of the two stage approaches with respect to the final objective, namely the object pose. Hence, Hu et al. [10] proposed to replace the non-differentiable RANSAC based pose estimator with a trainable neural network to estimate the 6DoF object pose. Their end-to-end trainable model showed improved results compared to the two stage approach as validated with two state of the art correspondence estimators [8] [9]. We follow up on their model with [9] as the correspondence estimator as it shows superior performance and refer to it as SSPE.

While SSPE shows improved performance using end-to-end training, it does not utilize local and global geometric constraints. In this work, we propose pairwise features to utilize local information between direction vectors associated with the same 3D point, and triplet regularization to account for the global geometry between pairwise features associated with different 3D points. Coupled with increased masking augmentation, our model achieves state of the art results on the Occlusion

Workshop on Differentiable Vision, Graphics, and Physics in Machine Learning at NeurIPS 2020.

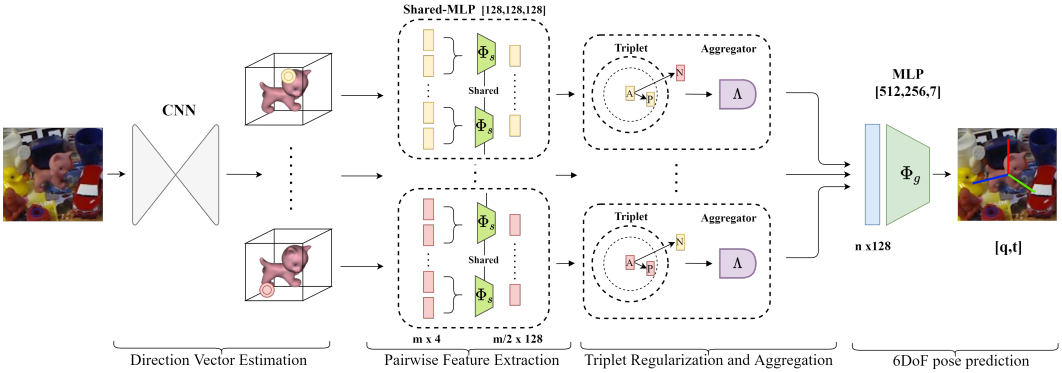


Figure 1: Illustration of our network architecture (SSPE-ours). The correspondence estimator predicts direction vectors to the 2D keypoints. Pairs of direction vectors are passed through a shared network Φ_s to give pairwise features which are aggregated using an aggregator Λ, and passed through a second network Φ_g to predict the pose. The color of the pairwise features indicates association to a 3D point.

Linemod [11] dataset and competitive results on the Linemod [12] dataset. In summary, our main contributions are:

- Pairwise feature extraction from direction vectors to better utilize local information
- Triplet regularization to account for the global geometry of the pairwise features
- State of the art results on Occlusion Linemod and competitive results on Linemod

2 Approach

We illustrate our approach in Figure 1. The correspondence estimator operates on an image and predicts a segmentation mask. It also predicts direction vectors to the 2D keypoints for each pixel in the mask. For each of the n 3D points p_i, the pose estimator selects m random direction vectors u_{ik} ($1 \leq i \leq n$, $1 \leq k \leq m$) from the segmented region of the object. It applies a shared MLP Φ_s to extract pairwise features, followed by aggregation using an aggregator Λ, and pose prediction from a second MLP Φ_g.

2.1 Local Constraint

A direction vector u_{ik} is represented as a 4D input $[x, y, dx, dy]$ where x, y is the pixel location, and dx, dy is the predicted vector from that pixel. In the first step for the SSPE pose estimator, a shared MLP is applied across all direction vectors to extract $n \times m$ local features. However, by operating on every direction vector independently the local features do not have information about the 2D keypoints. This is because a 2D keypoint is given by the intersection of a pair of direction vectors pointing to that keypoint [9]. Hence, we propose to concatenate pairs of direction vectors $[u_{ik}, u_{il}]$ and provide them as input to the shared MLP Φ_s. This gives us $n \times \frac{m}{2}$ D dimensional features f_{ih} ($1 \leq h \leq \frac{m}{2}$) termed as *pairwise features*.

$$f_{ih} = \Phi_s([u_{ik}, u_{il}]) \qquad 1 \leq i \leq n, \;\; 1 \leq h \leq \frac{m}{2}, \;\; k = 2h-1, \;\; l = 2h \tag{1}$$

While Φ_s can theoretically learn to approximate the intersection of direction vectors to give pairwise features with information about the 2D keypoints, we observe that adding global constraints can help learn better features for improved performance.

2.2 Global Constraint

We account for the global geometry of the pairwise features by considering their association to the 3D points. We want pairwise features associated with the same 3D point to be similar to each other, and pairwise features associated with different 3D points to be dissimilar to each other. To encourage this property we introduce a triplet regularization term. This also serves as a form of proxy supervision

Table 1: Results on Occlusion Linemod (Part I) and Linemod (Part II) using the ADD0.1d metric[1].

	Part I: Occlusion Linemod					**Part II: Linemod**			
	PVNet [9]	**DPVR** [13]	**SSPE** [10]	**SSPE-r**[2]	**SSPE-ours**	**PVNet** [9]	**DPVR** [13]	**SSPE-r**	**SSPE-ours**
Ape	15.8	19.2	19.2	**20.8**	18.8	43.6	**69.1**	66.7	52.5
Can	63.3	69.8	65.1	78.4	**79.3**	95.5	98.5	95.8	**99.2**
Cat	16.7	**21.1**	18.9	18.2	17.5	79.3	83.1	84.1	**88.5**
Driller	65.7	71.6	69.0	73.8	**76.4**	96.4	**99.0**	98.4	98.8
Duck	25.2	34.3	25.3	33.1	**34.4**	52.6	63.5	60.4	**68.7**
Eggbox*	50.2	47.3	**52.0**	46.0	44.6	99.2	**100.0**	99.7	**100.0**
Glue*	49.6	39.7	51.4	49.2	**53.2**	95.7	98.0	90.4	**98.5**
Holepuncher	39.7	45.3	45.6	53.5	**54.7**	81.9	**88.2**	85.3	88.1
Average	40.8	43.5	43.3	46.6	**47.4**	80.5	**87.4**	85.1	86.8

[1] We do not compare against models that perform refinement on predicted pose [14][15].
[2] We reimplement SSPE as authors have not open sourced the training code

to the shared MLP Φ_s as different pairs of direction vectors associated with the same 3D point give similar pairwise features. We mine triplets online and compute the triplet regularization term as:

$$\mathcal{L}_t = \frac{2}{nm} \sum_{i=1}^{n} \sum_{h=1}^{\frac{m}{2}} max(S_{ih,jd} - S_{ih,is} + \alpha, 0) \qquad 1 \leq j \leq n, \;\; i \neq j, \;\; 1 \leq d, s \leq \frac{m}{2} \tag{2}$$

where α is the margin and $S_{wx,yz}$ is the similarity between pairwise features f_{wx} and f_{yz}. We use the cosine similarity function given as:

$$S_{wx,yz} = \frac{f_{wx}^T f_{yz}}{||f_{wx}|| \; ||f_{yz}||} \qquad 1 \leq w, y \leq n, \;\; 1 \leq x, z \leq \frac{m}{2} \tag{3}$$

Similar to SSPE, we aggregate the pairwise features and apply a second MLP to compute the pose. The pairwise features associated with each 3D point are aggregated using an aggregator Λ to give n D dimensional group features g_i. We choose Λ as the mean pooling aggregator.

$$g_i = \Lambda(\{f_{i1}, f_{i2} ... f_{i\frac{m}{2}}\}) \qquad 1 \leq i \leq n \tag{4}$$

The group features are concatenated, and the nD dimensional vector is passed through a second MLP Φ_g to predict the pose as a quaternion $\hat{q}$ and translation $\hat{t}$.

$$[\hat{q}, \hat{t}] = \Phi_g([g_1, g_2 ... g_n]) \tag{5}$$

We recover the predicted rotation matrix $\hat{R}$ from $\hat{q}$ and compute the pose loss $\mathcal{L}_p$ as the 3D error:

$$\mathcal{L}_p = \frac{1}{n} \sum_{i=1}^{n} ||(\hat{R}p_i + \hat{t}) - (Rp_i + t)|| \tag{6}$$

where R and t are the ground truth rotation and translation.

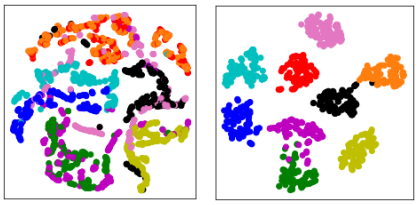
The final loss $\mathcal{L}$ to optimize is a linear combination of the cross entropy segmentation loss $\mathcal{L}_s$ and L1 vector regression loss $\mathcal{L}_k$ from the correspondence estimator [9], and the pose loss $\mathcal{L}_p$ and triplet regularization term $\mathcal{L}_t$ from the pose estimator.

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_k \mathcal{L}_k + \lambda_p \mathcal{L}_p + \lambda_t \mathcal{L}_t \tag{7}$$

3 Experiments

3.1 Training

We use $n = 9$ 3D key points for each object selected using the farthest point sampling algorithm. For the pose estimator, we randomly select $m = 200$ direction vectors for each of the 3D points. The triplet margin α is set to 0.1. The loss coefficients λ_s and λ_k are set to 1, λ_p is set to 0.01 and λ_t is set to 0.1. As per previous studies [9][10], we train separate models for each object. Training images are provided at an input resolution of 640×480 and augmented using scaling, translation, rotation, occlusion [16], gaussian blurring and colour jittering. We use the Adam optimizer and set the learning rate to $1e-3$ which is divided by 10 after processing 50%, 75%, and 90% of the data. All models are trained with a batch size of 32 for 300 epochs.



(a) (b)

Figure 2: t-SNE plot of the SSPE-r local features (a) and the SSPE-ours pairwise features (b) for the holepuncher object. Each colour represents the features of the 9 3D points.

Table 2: Ablation study on 4 non-symmetric and 1 symmetric object. Adding local constraints (SSPE-lc) improves performance over SSPE-r and SSPE-rp. Adding global constraints (SSPE-ours) further improves performance. Results reported on Occlusion Linemod using the ADD0.1d metric.

	SSPE-m	**SSPE-r**	**SSPE-rp**	**SSPE-lc**	**SSPE-ours**
Can	71.9	78.4	79.5	77.6	79.3
Driller	62.4	73.8	75.5	76.1	76.4
Duck	28.8	33.1	32.3	32.3	34.4
Glue*	51.3	49.2	53.6	55.9	53.2
Holepuncher	44.7	53.5	51.1	52.7	54.7
Average	51.8	57.6	58.4	58.9	59.6

3.2 Evaluation

We benchmark our approach on the Linemod [12] and Occlusion Linemod [11] datasets for 8 object classes. Similar to previous approaches [9][10], we augment the Linemod train data using synthetic data. We generate 10000 images containing multiple objects using the cut and paste [17] technique, and 8×10000 images of single objects using the rendering technique in [9].

For evaluation, we use the ADD0.1d metric [12] to measure accuracy in 3D space. The ADD0.1d metric measures the average distance between the 3D model points transformed using the predicted pose and the ground truth pose. A predicted pose is assumed correct if the average distance is less than 10% of the model diameter. We report the percentage of correctly predicted poses. We use the symmetric version of the metric [5] for symmetric objects, which are denoted by the * superscript.

3.3 Results

We report results on the Occlusion Linemod dataset in Part I of Table 1. SSPE-ours achieves state of the art results with a 9% improvement over the previous best method [13]. It has the highest scores for 5 of the 8 objects.

We perform ablation in Table 2 to demonstrate the strength of our approach. Average performance of SSPE with pairwise features (SSPE-lc) is better compared to standard SSPE with the aggregator as max pooling (SSPE-r) and SSPE with the aggregator as mean pooling (SSPE-rp). Adding triplet regularization (SSPE-ours) further improves performance. To support our hypothesis we do a t-SNE visualisation of the SSPE local features and our pairwise features as shown in Figure 2. We note much better clustering for our pairwise features. This suggests our approach successfully accounts for the local and global constraints to improve end-to-end pose estimation.

We also observe that increased masking augmentation [16] can help increase performance. We highlight its importance in Table 2 by initially setting the masking percentage to $10\% - 30\%$ (SSPE-m), and then tripling it to $30\% - 90\%$ (SSPE-r). We note an average increase of 5.8 points in ADD0.1d score. Hence, we use the increased masking in all our experiments.

We additionally show results on the Linemod dataset in Part II of Table 1. SSPE-ours achieves competitive results and has the highest scores for 5 of the 8 objects. It also shows improvement over SSPE-r.

4 Conclusion

We show that our approach (SSPE-ours) achieves state of the art results on the challenging Occlusion Linemod dataset. We also perform ablation to demonstrate the strength of our approach. This suggests the effectiveness of local and global constraints to improve end-to-end 6DoF object pose estimation. In the future, we hope to explore geometric properties to further improve end-to-end 6DoF object pose estimation.

References

[1] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2016.

[2] 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, 2016.

[3] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5452–5462.

[4] 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.

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

[6] 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.

[7] 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.

[8] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6d object pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3385–3394.

[9] 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.

[10] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2930–2939.

[11] 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.

[12] 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.

[13] X. Yu, Z. Zhuang, P. Koniusz, and H. Li, "6dof object pose estimation via differentiable proxy voting loss," *Proceedings of the British Machine Vision Conference*, 2020.

[14] 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.

[15] 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.

[16] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[17] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1301–1310.