LEARNING ROBUST FEATURES FOR 3D OBJECT POSE ESTIMATION

Christos Papaioannidis and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Greece

ABSTRACT

Object pose estimation remains an open and important task for autonomous systems, allowing them to perceive and interact with the surrounding environment. To this end, this paper proposes a 3D object pose estimation method that is suitable for execution on embedded systems. Specifically, a novel multi-task objective function is proposed, in order to train a Convolutional Neural Network (CNN) to extract pose-related features from RGB images, which are subsequently utilized in a Nearest-Neighbor (NN) search-based post-processing step to obtain the final 3D object poses. By utilizing a symmetry-aware term and unit quaternions in the proposed objective function, our method yielded more robust and discriminative features, thus, increasing 3D object pose estimation accuracy when compared to state-of-the-art. In addition, the employed feature extraction network utilizes a lightweight CNN architecture, allowing execution on hardware with limited computational capabilities. Finally, we demonstrate that the proposed method is also able to successfully generalize to previously unseen objects, without the need for extra training.

***Index Terms*—** 3D object pose estimation, Multi-task learning, Convolutional Neural Networks.

1. INTRODUCTION

Autonomous robots or systems are being increasingly employed in several industries (e.g., transportation, construction) to assist in simple or more complex tasks. However, their safe and successful operation in real-world scenarios requires advanced understanding of the surrounding environment. Object pose estimation is critical in this case, as it enables predicting the 3D poses of objects of interest in their surroundings, in order to autonomously take the correct actions according to a given objective. For example, in a human-Unmanned Aerial Vehicle (UAV or drone) collaboration scenario, the UAV should be able to estimate the 3D pose of a tool in order to grab it (e.g., by using a robotic arm) and pass it to a human worker.

Early deep learning-based methods addressed the 3D object pose estimation problem by training a Convolutional Neural Network (CNN) to directly regress [1, 2] or classify [3] an object image to its 3D pose. More recent 6D object pose estimation methods [4, 5, 6, 7, 8] utilized state-of-the-art object detection [9] and instance segmentation [10] methods to first localize the objects of interest in the 2D image (e.g., by regressing their 2D bounding boxes or a set of predefined keypoints) and then computed the final 6D object poses by using the 2D detections in a PnP algorithm. However, these approaches either lack increased 3D object pose estimation accuracy, or rely on very deep neural network architectures, which do not allow fast execution on embedded systems.

In an alternative approach, the final 3D object pose predictions can be obtained indirectly [11, 12, 13, 14, 15, 16]. That is, a lightweight CNN is first utilized to extract object image features, which are then matched with a set of precomputed database entries that represent orientation classes via a Nearest Neighbor (NN) search. While these methods managed to achieve increased 3D object pose estimation performance despite only using a lightweight CNN, non-trivial object symmetries [11, 12, 14, 15, 16] and poor 3D pose representation [13] can cause convergence issues during CNN training.

In this work, we propose a 3D object pose estimation method that aims to overcome all the aforementioned limitations of existing methods. More specifically, we improve the existing feature learning methods by introducing a novel multi-objective loss function to train a lightweight CNN to extract pose-related object image features. The proposed loss function utilizes unit quaternions to represent 3D poses and a symmetry-aware term to handle non-trivial symmetries of objects which facilitate the feature learning process, resulting in discriminative pose-related features from which the final 3D object poses can be more accurately obtained. Moreover, since the proposed loss function is carefully designed to encode 3D object poses in the extracted features, the proposed method is able to successfully generalize to previously unseen objects. Finally, the lightweight architecture of the feature extraction CNN allows execution on hardware with limited computational capabilities, rendering the proposed method suitable for autonomous systems.

In summary, this paper offers the following contributions:

- a novel multi-task objective function for 3D pose feature learning that combines unit quaternions and a symmetry-aware term,

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement numbers 731667 (MULTIDRONE) and 871479 (AERIAL-CORE).

- a 3D object pose estimation method with increased 3D object pose estimation performance and generalization ability that is suitable for embedded systems.

2. PREVIOUS FEATURE LEARNING 3D OBJECT POSE ESTIMATION METHODS

Feature learning pose estimation methods [11, 13, 14, 15, 16, 17] offer several advantages over pose regression [1, 4] or classification [3] approaches, as they only require a shallow CNN architecture, they are scalable to the number of objects [14] and can simultaneously perform object classification. In this case, a CNN is first trained to extract pose-related features from an object image. At test time, the extracted object image features are matched with a set of precalculated database image features via NN search, returning as final object class and 3D pose estimates the ones that correspond to the retrieved closest database image.

In order to learn such pose-related image features, a feature learning pose estimation framework was firstly introduced in [11], where a lightweight CNN was trained to calculate discriminative features using Siamese [18] and triplet [19] network architectures. The feature learning loss function which was used in [11] to train the CNN was of the following form:

$$\mathcal{L} = \mathcal{L}_d + \lambda_w \|\mathbf{w}\|_2^2, \tag{1}$$

where $\mathcal{L}_d$ is a feature learning term, λ_w is a regularization parameter and $\|\mathbf{w}\|_2$ is the L_2-norm of the network parameter vector. $\mathcal{L}_d$ consists of a pairwise and a triplet loss function, $\mathcal{L}_d = \mathcal{L}_{pairs} + \mathcal{L}_{triplets}$ [11], in order to learn object image features from which both the 3D object pose and the object class can be retrieved. $\mathcal{L}_{pairs}$ is responsible for keeping object images with similar poses close in the feature space, while $\mathcal{L}_{triplets}$ aims to distinguish between different object identities. Later, the total loss function (1) was extended in [14] by adding a dynamic margin in the triplet loss function $\mathcal{L}_{triplets}$ to improve the robustness of the resulting low-dimensional features. The dynamic margin, which utilized unit quaternions and quaternion distance, was defined as:

$$\varepsilon_d = \begin{cases} 2\arccos(|\mathbf{q}_i^T \mathbf{q}_j|) & if\ c_i = c_j, \\ n & \text{else, for } n > \pi, \end{cases} \tag{2}$$

where $\mathbf{q}_i$, $\mathbf{q}_j$ and c_i, c_j are the corresponding ground truth 3D pose and object class labels of training samples s_i, s_j, respectively, while n is a constant value for penalizing pairs from different object classes.

In order to learn more discriminative features for 3D object pose estimation and object recognition, a pose-guided pairwise loss and an extra 3D pose regression term was used in the total loss function in [13]:

$$\mathcal{L} = \mathcal{L}_{pose} + \mathcal{L}_{object} + \mathcal{L}_{reg} + \lambda \|\mathbf{w}\|_2^2. \tag{3}$$

In this case, the pairwise loss $\mathcal{L}_{pose}$ enforces a direct relationship between the learned features and the real pose label differences, while the role of triplet loss $\mathcal{L}_{object}$ remains the same as in [11, 14]. The trained model yielded pose-related features that greatly improved the 3D object pose estimation performance compared to [11]. The work of [14] was also extended in [15, 16], where a quaternion regression term was used in the total loss function, along with the feature learning term $\mathcal{L}_d$. By using quaternions and quaternion regression, the trained model not only demonstrated increased 3D object pose estimation performance, but also enabled direct 3D object pose regression by completely omitting the NN search step.

3. MULTI-TASK FEATURE LEARNING

The objective function used in the proposed 3D object pose estimation method is:

$$\mathcal{L} = \lambda_p \mathcal{L}_{pose} + \lambda_o \mathcal{L}_{obj} + \lambda_r \mathcal{L}_{qreg}, \tag{4}$$

where $\mathcal{L}_{pose}$, $\mathcal{L}_{obj}$, $\mathcal{L}_{qreg}$ are the pairwise, triplet and quaternion regression loss functions, respectively, and λ_p, λ_o, λ_r are hyper-parameters used to control the contribution of each term in the total loss function. The key difference between the proposed loss function and (3) is that the terms $\mathcal{L}_{pose}$, $\mathcal{L}_{qreg}$ in (4) utilize unit quaternions and quaternion distance. Unit quaternions $\mathbf{q} \in \mathbb{R}^4$, $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$, $\|\mathbf{q}\|_2 = 1$, offer a preferable alternative for rotations representation, as they are more compact compared to rotation matrices and also avoid the gimbal lock problem [20] of the Euler angle representation. Since unit quaternions double-cover the $\mathcal{SO}(3)$ ($\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation), in the proposed method we enforce $q_0 \geq 0$ in order to achieve a one-to-one correspondence between rotation matrices and quaternions. Moreover, the quaternion distance offer an accurate representation of real pose differences, which is required in the proposed method.

The pairwise ($\mathcal{L}_{pose}$) and the triplet ($\mathcal{L}_{obj}$) loss functions utilized in (4) are specifically designed in order to learn a discriminative feature space during CNN training. Let $s_i = \{\mathbf{x}_i, c_i, \mathbf{q}_i\}, i = 1, \ldots, N$ be a training set sample which contains an RGB-D image $\mathbf{x}_i$ of an object, its assigned object class label $c_i \in \mathcal{C} = \{c_1, \ldots, c_L\}$ and the corresponding 3D pose quaternion $\mathbf{q}_i \in \mathbb{R}^4$. Also, let $\mathcal{P} = \{s_i, s_j\}$, $\mathcal{T} = \{s_i, s_j, s_k\}$ be sets containing training sample pairs and triplets, respectively. The pairwise loss $\mathcal{L}_{pose}$ is computed on pairs $\{s_i, s_j\} \in \mathcal{P}$, where the samples s_i, s_j belong in the same object class $c_l, l = 1, \ldots, L$ and is used to enforce pose similarity within the same object class c_l. Note that some objects may seem very similar under different poses, therefore, we need our model to be able to handle these cases where objects are symmetric in a non-trivial way. To this end, we weight the contribution of each sample in $\mathcal{L}_{pose}$ with the corresponding symmetry-aware term $\phi(\mathbf{q}_i, \mathbf{q}_j)$ to impose infor-

mation about object symmetries to the model. Therefore, if $\mathbf{f}_i = f(\mathbf{x}_i) \in \mathcal{F} \subset \mathbb{R}^d$ are the features obtained from the last fully connected CNN layer having $\mathbf{x}_i$ as input, the pairwise loss is defined as:

$$\mathcal{L}_{pose} = \sum_{s_i, s_j} \phi(\mathbf{q}_i, \mathbf{q}_j) \cdot \{\|\mathbf{f}_i - \mathbf{f}_j\|_2^2 - 2\arccos(|\mathbf{q}_i^T \mathbf{q}_j|)\}^2, \tag{5}$$

where $\phi(\mathbf{q}_i, \mathbf{q}_j) = \|\mathbf{d}_{\mathbf{q}_i} - \mathbf{d}_{\mathbf{q}_j}\|_2^2$ and $\mathbf{d}_{\mathbf{q}_i}$, $\mathbf{d}_{\mathbf{q}_j}$ are the rendered depth object images under the poses $\mathbf{q}_i$, $\mathbf{q}_j$, respectively. Essentially, $\mathcal{L}_{pose}$ forces the Euclidean feature distance between two samples from the same object class to be equal to the quaternion distance between the corresponding 3D poses $\mathbf{q}_i$, $\mathbf{q}_j$. However, as in some cases an object may appear the same when seen from different viewpoints, convergence problems may occur during training. By using $\phi(\mathbf{q}_i, \mathbf{q}_j)$ in (5), the learned features of symmetric samples with very similar rendered depth images $\mathbf{d}_{\mathbf{q}_i}$, $\mathbf{d}_{\mathbf{q}_j}$ are not directly affected by the magnitude of the symmetry-agnostic $2\arccos(|\mathbf{q}_i^T \mathbf{q}_j|)$ term. Therefore, minima closer to the global minimum can be located during the network optimization process.

The triplet loss term $\mathcal{L}_{obj}$ in (4) enforces features coming from same-object class samples to have smaller distances in the feature space, when compared to the distances of features calculated from different object class samples. For this purpose, the sample triplets $\{s_i, s_j, s_k\} \in \mathcal{T}$, consist of samples s_i, s_j coming from the same object class $c_l, l = 1, \ldots, L$, while s_k is a sample coming from any different object class. $\mathcal{L}_{obj}$ is of the following form:

$$\mathcal{L}_{obj} = \sum_{s_i, s_j, s_k} \frac{\|\mathbf{f}_i - \mathbf{f}_j\|_2}{\|\mathbf{f}_i - \mathbf{f}_k\|_2 + \varepsilon}, \tag{6}$$

so that the distance in the feature space between the same object class is forced to be smaller than the distance between object features coming from different classes. ε is a small regularizing constant, that also prevents having a zero denominator in (6).

The quaternion regression term $\mathcal{L}_{qreg}$ is defined as:

$$\mathcal{L}_{qreg} = 2arccos(|\mathbf{q}^T \hat{\mathbf{q}}|), \tag{7}$$

where $\mathbf{q}$, $\hat{\mathbf{q}}$ are the ground truth and the predicted 3D object pose quaternion, respectively. $\mathcal{L}_{qreg}$ not only enables the CNN to directly regress the 3D object pose in the form of a unit quaternion, but also assists the feature learning process by imposing extra information about 3D object poses to the model. However, quaternion regression requires special attention, as the four quaternion entries q_0, q_1, q_2, q_3 are not independent. The term $\sin\frac{\theta}{2}$ is found in all three entries q_1, q_2, q_3, while $\cos\frac{\theta}{2}$ contributes to q_0. Therefore, trying to directly regress $\mathbf{q}$ leads to inferior performance [21]. In contrast, in the proposed method, unit quaternions are obtained implicitly. That is, the independent axis-angle rotation representation entries $\mathbf{r} = [\theta', u_1, u_2, u_3]^T$, $\theta' = \frac{\theta}{2}$ are regressed, where $\mathbf{u} \in \mathbb{R}^3$, $\mathbf{u} = [u_x, u_y, u_z]^T$ is the unit rotation axis and $\theta \in \mathbb{R}$ its rotation angle. Ultimately, the predicted 3D object pose quaternion $\hat{\mathbf{q}} = [\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3]$ used in (7) is obtained as follows:

$$\begin{cases} \hat{q}_0 = \cos(\theta') \\ \hat{q}_1 = u_1 \sin(\theta') \\ \hat{q}_2 = u_2 \sin(\theta') \\ \hat{q}_3 = u_3 \sin(\theta'). \end{cases} \tag{8}$$

4. EXPERIMENTAL EVALUATION

The CNN used in the proposed method has the same architecture as the ones used in [11, 13, 14, 15]. The first two network layers are convolutional ones with rectified linear (ReLU) activation function, followed by two fully connected layers. The final fully connected layer produces the 3D pose feature vector $\mathbf{f} \in \mathcal{F} \subset \mathbb{R}^d$. In all the experiments, the feature dimensionality was set to $d = 32$. For the quaternion regression, an extra fully connected layer is added after the feature layer. This layer is followed by the quaternion activation layer, which outputs $\hat{\mathbf{q}}$, according to (8). We empirically set $\lambda_p = 10$, $\lambda_o = 1$ and $\lambda_r = 0.5$ in all our experiments to benefit 3D pose feature learning. The overall CNN is trained for 400 epochs using the stochastic gradient decent method, with momentum 0.9 and initial learning rate of 0.01, which is reduced in each epoch.

The proposed method is compared to the baseline methods of [11, 13, 14, 15]. In all experiments all models were trained using the Cropped LineMOD dataset [11]. It has to be noted that in all cases, the estimated 3D object pose is the ground truth pose assigned to the closest database sample retrieved by the nearest neighbor search. For the quantitative evaluation of all trained models, the angular error metric is used [14]:

$$err(\mathbf{q}, \hat{\mathbf{q}}) = 2\arccos(|\mathbf{q}^T \hat{\mathbf{q}}|), \tag{9}$$

where $\mathbf{q}$, $\hat{\mathbf{q}}$ are the the ground truth and the estimated object pose, respectively. The 3D pose estimation accuracy at threshold t is then defined as the percentage of test samples, for which the angular error between the estimated and the ground truth pose is below a threshold angle t, $err(\mathbf{q}, \hat{\mathbf{q}}) < t$. Note that, the pose estimation accuracy is calculated only for the test samples that were correctly matched to their corresponding object class.

The comparison between the performance of the proposed and the baseline CNN models $3DPOD$ [11], $PEDM$ [14], $PGFL$ [13] and QL [15] for threshold angle values $t \in [5°, 10°, 15°, 20°, 30°, 40°, 45°]$ is presented in Table 1, where the object classification accuracy is also reported. It should be noted that, since the code of [14] could not be made available, the results reported in [14] are directly cited in Table 1 only for threshold angle values $t \in [10°, 20°, 40°]$. As can be seen in Table 1, the proposed method outperforms

Table 1. 3D object pose estimation and object classification accuracy.

	Angular threshold t								
	5°	**10°**	**15°**	**20°**	**30°**	**40°**	**45°**	**Mean (Median) ± Std**	**Object classification**
$3DPOD$ [11]	40.15%	72.72%	86.02%	91.76%	95.42%	96.90%	97.34%	12.75°(7.06°) ± 24.61°	98.94%
$PEDM$ [14] ⋆	-	60.00%	-	93.20%	-	98.00%	-	-	99.30%
$PGFL$ [13]	41.28%	83.07%	93.98%	97.43%	99.11%	99.52%	99.60%	6.89°(5.79°) ± 6.29°	99.64%
QL [15]	41.37%	82.02%	95.32%	98.49%	99.72%	99.92%	99.94%	6.64°(5.78°) ± 5.14°	99.50%
$ours$	**44.13%**	**84.25%**	**95.76%**	**98.77%**	**99.84%**	**99.93%**	**99.94%**	**6.31°(5.53°) ± 4.58°**	**99.68%**

⋆ The results of $PEDM$ are directly cited from [14].

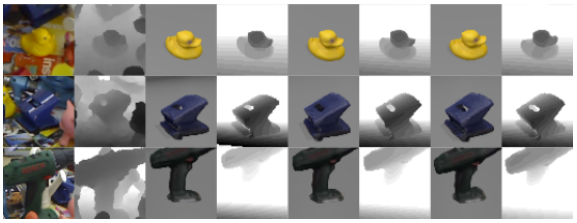

Fig. 1. Retrieved top 3 nearest neighbors for 3 query images from Cropped LineMOD dataset [11]. First two columns show the query RGB-D images and the rest columns depict the retrieved closest nearest neighbors from left to right.

all competing methods, yielding increased 3D object pose estimation accuracy for all angle threshold values. Particularly in the high 3D object pose estimation accuracy area ($t = 5°$) the proposed method increased the 3D object pose estimation accuracy up to 4%. In addition, the proposed method outperforms all competing methods in the object classification task. As also reported in Table 1, the proposed method has lower mean and standard deviation values of the angular error compared to all competing models. The results show that by incorporating the symmetry-aware term ϕ in the quaternion-based feature learning process using (4), the proposed method is able to extract more discriminative pose-related features that enable increased 3D object pose estimation performance.

Apart from the comparison reported in Table 1, we also performed a qualitative evaluation of the proposed method. More specifically, the images of the closest 3 database samples retrieved by the proposed method for random query test images from the Cropped LineMOD dataset are presented in Fig. 1. It can be seen that all query images are successfully matched to database samples that have very similar 3D pose, with the 3D pose difference between them being imperceptible in most cases. Finally, the generalization ability of the proposed method on a previously unseen object (car in random scenes, simulating an objective of a self-driving car scenario) is evaluated by utilizing images from WCVP [22] dataset. Thus, random images from the first split of WCVP dataset were used as query images, while all images from the second split of WCVP dataset were utilized as database samples. Note that, as a pre-processing step, depth images were extracted using the depth estimation method of [23] (since depth images are not provided by WCVP) and then both RGB and depth images were cropped using the ground truth 2D bounding boxes provided by WCVP. The results presented in Fig. 2 show that the proposed method was able to match query images with database samples that have almost identical 3D poses, without the need for an extra training step and despite the fact that cars between query and database images may have different shape or color.

Fig. 2. Generalization ability of the proposed method on previously unseen object. First two columns show the query RGB-D images and the rest columns depict the retrieved closest nearest neighbors from left to right.

5. CONCLUSION

In this work, a 3D object pose estimation method for embedded execution was presented. By utilizing a lightweight CNN and a specifically designed 3D pose feature learning objective function that considers non-trivial object symmetries, the proposed method yielded more discriminating 3D pose features, hence, outperforming state-of-the-art feature learning methods. Experiments in the Cropped LineMOD dataset showed that the proposed method increased the 3D object pose estimation accuracy for all angle threshold values as well as the object classification accuracy. Finally, the proposed method demonstrated increased generalization ability to unseen objects, without the need for extra training.

6. REFERENCES

[1] 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.

[2] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[3] 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 (ICCV)*, 2015.

[4] T.-T. Do, M. Cai, T. Pham, and I. Reid, "Deep-6Dpose: Recovering 6D object pose from a single rgb image," *arXiv preprint arXiv:1802.10367*, 2018.

[5] 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 (ICCV)*, 2017.

[6] 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 (ICCV)*, 2017.

[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 (CVPR)*, 2018.

[8] 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 (CVPR)*, 2019.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3D object detection and 6D pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[13] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim, "Pose guided rgbd feature learning for 3D object pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] S. Zakharov, W. Kehl, B. Planche, A. Hutter, and S. Ilic, "3D object instance recognition and pose estimation using triplet loss with dynamic margin," in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[15] C. Papaioannidis and I. Pitas, "3D object pose estimation using multi-objective quaternion learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2683–2693, 2019.

[16] M. Bui, S. Zakharov, S. Albarqouni, S. Ilic, and N. Navab, "When regression meets manifold learning for object recognition and pose estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[17] C. Papaioannidis, V. Mygdalis, and I. Pitas, "Domain-translated 3D object pose estimation," *IEEE Transactions on Image Processing*, vol. 29, pp. 9279–9291, 2020.

[18] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," *Advances in Neural Information Processing Systems*, 1994.

[19] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*, 2015.

[20] S. L. Altmann, *Rotations, quaternions, and double groups*, Courier Corporation, 2005.

[21] H.-W. Hsu, T.-Y. Wu, S. Wan, W. H. Wong, and C.-Y. Lee, "Quatnet: Quaternion-based head pose estimation with multi-regression loss," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1035–1046, 2018.

[22] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and continuous pose estimation," *Image and Vision Computing*, vol. 30, no. 12, pp. 923–933, 2012.

[23] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," *arXiv preprint arXiv:2011.14141*, 2020.