KOSNet: A Unified Keypoint, Orientation and Scale Network for Probabilistic 6D Pose Estimation

Kunimatsu Hashimoto*, Duy-Nguyen Ta*, Eric Cousineau and Russ Tedrake
*These authors contributed equally to this work.

***Abstract*— We propose a novel method using a Convolutional Neural Network (CNN) for probabilistic 6D object pose estimation from color images. Unlike other methods that compute only one data point as the output, our network returns the information necessary to estimate the full probability distributions of 6D object poses. This not only captures the ambiguity of object appearance in the image in a principled manner, but also enables the results to be fused with other sensing modalities using well-established probabilistic inference techniques. One of the main challenges is to provide probabilistic ground truth labels for training the network. To this end, we introduce a way to approximate uncertainties of object poses related to rotational symmetry, occlusion, and how distinct an object is from the background. We demonstrate the unique capability of our network on both fully and partially rotationally symmetric objects while achieving comparable performance with a state-of-the-art method on publicly available datasets.**

I. Introduction

Recognizing objects and estimating their 6D poses from color images are often critical steps in robotics applications to enable manipulation of particular objects of interest in the scene. However, despite significant progress on 6D pose estimation methods using deep neural networks [1], [2], [3], [4], [5], there are two remaining challenges that have not been sufficiently addressed by the state-of-the-art: (1) how to fuse 6D pose outputs from a neural network with results from other sensing modalities and (2) how to handle the ambiguity of object appearance due to occlusion, camouflage and/or rotational symmetry.

This paper presents KOSNet, a unified **K**eypoint, **O**rientation and **S**cale **Net**work, for probabilistic 6D pose estimation that can address both problems at the same time. Our network achieves that goal by outputting probability distributions of the object's 6D pose instead of just point-wise estimates as is usually done by other methods. The benefits of outputting probabilistic distributions are tremendous. First, it can capture the uncertainties of pose estimates due to ambiguities in object geometry and/or image information in a principled manner. For example, rotation estimates of a rotationally symmetric object should have large uncertainties because it looks the same at different angles around the axis of symmetry. Similarly, the ambiguity due to occlusion or camouflage can also be captured in probability densities. More importantly, it enables the neural network's outputs to be fused with other sensing modalities using well-established probabilistic methods [6], [7].

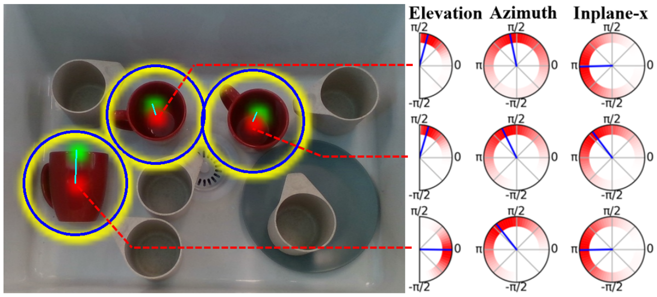

Fig. 1: KOSNet's output: Beside belief maps of keypoints (red and green), it learns to output distributions of orientations (right figure) and scales (yellow). Ground truth in blue.

Our network extends a CNN-based keypoint detection network to output probabilistic belief maps of object keypoint locations, orientations and scales – all possible geometric cues that can be extracted from object appearance in an input image. Recent works in 6D pose estimation using belief maps only output heatmaps of 2D keypoint locations [5], [8] and obtain the pose from 3D-2D correspondences using PnP [9]. The main disadvantage of these approaches is that keypoints themselves are insufficient for objects with rotational symmetry, because the position of certain keypoints, e.g. the bounding box corners [5], cannot be uniquely defined. Our network fixes this problem by learning distributions of orientations and scales directly.

However, there are two key challenges towards our goal: (1) how to architect a CNN to learn distributions of orientations and scales and (2) how to generate ground-truth probability distributions for training the network. Learning distributions of orientations and scales is challenging with CNN-based architectures as they do not correspond directly to image pixels as keypoint locations do. We fix this problem by learning a discretized joint belief space of keypoints and object orientations, and estimating scales indirectly via belief maps of the object's 3D bounding sphere projection on the image. Regarding ground-truth distributions for training labels, a constant standard deviation is sufficient for keypoint belief maps [10], [5], but it is not enough to reflect the true amount of rotation uncertainties due to different kinds of ambiguities. Our method approximates the true uncertainty with a local Gaussian whose standard deviation is computed numerically using finite differencing on synthetic images.

The main contributions of our work are:

- a 6D pose estimation network that can output probability distributions which are ready to be fused with other

All the authors are with Toyota Research Institute, Cambridge, MA, United States {firstname.lastname}@tri.global

sources of probabilistic information and can represent estimation uncertainties due to the ambiguity of object appearance in input images,

- an extension of a CNN-based keypoint detection network to learn belief maps of rotations and scales whose spaces, unlike keypoints, are not isometric to the image space, and
- a method to approximate ground-truth belief maps capturing the ambiguities of object appearance in the image for training our network.

We plan to publish our code and dataset and include them in the final version of the paper.

II. Related work

Estimating the 6D pose of an object in a color image is a long-standing problem in computer vision [11], [12]. Hodaň et al. [13] presents benchmarking results of non-deep-learning methods on standard datasets. A summary of state-of-the-art methods using deep neural networks as of last year, 2018, can be found in [2]. Since then, the current trend seems to converge on the idea of detecting 2D keypoints of an object in the image, then using a PnP algorithm to compute the 6D object pose from 2D-3D point correspondences. This idea was pioneered by the BB8 [14] and Semantic Keypoints [15] networks. State-of-the-art methods significantly improve upon those results by exploiting recent advances in single-shot CNN architectures [16] or keypoint detection networks such as [5], [8]. These networks, however, give poor results when the object of interest is heavily occluded. More recent works focus on fixing this problem by using local patches to reduce the effect of occlusion [17] or adding a segmentation head to aggregate information only from pixels in the object regions [18], [19].

Despite fast and significant progress on the 6d pose estimation problem using deep neural networks, probabilistic fusion of network outputs with other sources of information is still a big challenge. This is because most networks only output single point estimates of the poses [4], [20], lacking the uncertainty information needed for fusion [6], [7]. For example, in multi-view pose estimation, it is challenging to infer the correct pose from conflicting results estimated from different views without knowledge about the uncertainty of the estimates. In [21], a voting scheme is used to choose the pose that best agrees with all other network outputs. The accuracy of this heuristic depends largely on the number of networks and the consistency of their outputs. Sensor fusion with neural networks has also been done by training all sensor inputs jointly [22], but this approach faces challenges in heterogeneous network design beside scalability issues, such as requiring retraining with new data when a new sensor is added in addition to the increase in network size. Several other works [15], [19] realize the benefits of heat maps of keypoints in enabling probabilistic fusion. However, keypoint-based methods cannot deal with rotationally symmetric objects [5]. Our network overcomes this challenge by learning the full heat maps of rotations, not just keypoints.

Handling ambiguities of object appearance is another big challenge for 6D pose estimation networks. If not handled carefully, ambiguities can cause network confusion during training due to vastly different pose outputs of similar-looking input images. The ambiguity caused by rotational symmetry is commonly addressed in pose-estimation networks, typically by treating symmetric objects differently [20], limiting the range of their poses in the training set [14], or using a carefully designed loss function to avoid the ambiguity [4], [23]. However, other types of ambiguities, e.g. due to occlusion or camouflage, have not been addressed sufficiently. For example, although a mug with a handle is not rotationally symmetric, its image appearance where the handle is completely occluded by itself or by other objects does not carry enough information to determine the exact amount of yaw rotation. Similarly, image appearance of a red mug on a red background is more ambiguous than its appearance on a green background. By outputting probability distributions of poses, our network is capable of capturing all these kinds of ambiguities.

Finally, we note that the goal of the latest work, PoseRBPF in [24], built on top of [1], is closest to ours. By forcing an augmented auto-encoder (AAE) to reconstruct a canonical output image from a training set of domain-randomized input images of the same viewpoint but vastly different in other dimensions, e.g. lighting direction, object color, image contrast, cluttered background, foreground occlusion, etc., the latent space of the AAE in [1] successfully encodes the generic rotation space and does not suffer from the rotational symmetry problem. PoseRBPF defines its likelihood function for probabilistic tracking as distances between the latent vector of the input image and those of canonical images. This metric, however, largely depends on the reconstruction quality of the decoder, which is sensitive to small shifts or scale changes. In contrast, our network learns to output the probability distributions directly.

III. Methodology

A. Pose representation

Fig. 2 shows our chosen representation of the camera pose in the object frame, which is convenient to learn with a belief map-based keypoint detection network, especially for objects with rotational symmetry. Our representation is based on the viewing ray, connecting the camera center C and the object center O, since object appearance strongly depends on the direction of this vector [25], [26], [27]. The object center O is chosen to be the centroid of the main rotationally symmetric part of the object's body, e.g. the centroid of the mug's body excluding its handle. The object's z-axis corresponds to the axis of rotational symmetry.

The camera's translation in the object frame is determined by the direction of vector $\overline{OC}$ in the object frame O together with its length. The camera's orientation in the object frame is determined by (1) the 2D coordinate of object's center keypoint on the image plane and (2) an in-plane rotation angle quantifying how much the object rotates around the

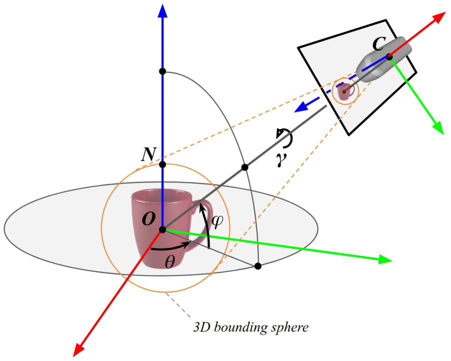

Fig. 2: KOSNet pose representation. The azimuth θ and elevation φ angles of the viewing ray and the size of the bounding circle on the image capture the camera position in the object frame. The rotation is captured by the image projection of the object center and the in-plane rotation γ.

viewing ray as detailed in [27]. These quantities provide us with full coverage of SE(3).

We represent $\overline{OC}$'s direction using its azimuth θ and elevation φ angles in the object frame. For objects with rotational symmetry around the z-axis, the azimuth distribution is uniform and easy to specify. We use the object's 2D bounding circle, the projection of the object's 3D bounding sphere around its center with a known radius, in the image to capture $\overline{OC}$'s length, given that the camera intrinsic parameters and object's 3D model are known. Similar to belief maps of keypoints, belief maps of 2D bounding circles are easy to specify and learn using the same network architecture. Moreover, unlike the popular 2D bounding box representation, the projection of a sphere is view-point independent as it is always a circle under all view angles.

We choose the center keypoint and in-plane rotation over other popular representations, e.g. Euler angles or $SO(3)$, to represent the camera rotation in the object frame, because the keypoint is ready to be learned using a belief map-based keypoint detection network. However, unlike the center keypoint, the in-plane rotation is not trivial to define due to a subtle singularity problem that is often ignored by most previous works [20], [27]. The amount of in-plane rotation around the viewing ray can be defined as the angle between the image projection of the object's z-axis and the image's x-axis. However, when the object's z-axis coincides with the viewing ray, its projection on the image becomes a point, and the angle is ill-defined. We overcome this singularity issue by using one more angle, measuring between the projection of the object's x-axis and the image's x-axis. The object's two axes compensate for each other: they cannot be both in the singularity condition at the same time, so at least one is always well-defined in every case. The projection of the object's z-axis is easy to represent for learning with a belief map by using a keypoint N, named the "north point", located at a known distance from the object center O along its z-axis. Unfortunately, the projection of the object's x-axis cannot be defined by a keypoint in the same way because it is ambiguous for rotationally symmetrical objects. Hence, we choose to represent the angle between the projection of the object's x-axis and the image's x-axis explicitly and refer to this as "in-plane x" for brevity.

B. Network architecture

Fig. 3 shows the network architecture of KOSNet. The basic architecture of KOSNet is structured on top of a belief map-based keypoint detection network, which we call it KPD for brevity, by taking the above mentioned representation into consideration. The base KPD outputs two 2D belief maps which correspond to the object center O and the north point N. In addition, KOSNet also outputs (1) a 2D belief map for object bounding circles and (2) three 3D belief maps for the joint distributions between the center keypoint and each of the elevation, azimuth, and in-plane x angles. The first two dimensions of the 3D belief maps correspond to the keypoint's dimensions and are the same as those of the feature map F, which represents the output of a backbone network, whereas the third dimension corresponds to one of the aforementioned angles.

As shown in Fig. 3, KOSNet has four main streams: scale, elevation, azimuth, and in-plane x in addition to the keypoint stream from the base KPD. Each of the four streams is given a dedicated branch in order to compute the feature. The building blocks for each of the branches are all identical except the input and output channel numbers. The all four branches take two stages. Each first stage, represented as blue blocks, consists of five convolutional blocks where each block takes a convolution layer, batch norm and ReLU, except the last block which only has a convolution layer. Similarly, each following stage, represented as pink blocks, includes seven convolutional blocks, each of which is similar to the convolutional blocks in the first stage, including the last one. Each stage outputs 2D or 3D belief maps and those are fed to the loss function and jointly minimized with the keypoint belief map using ground truth belief maps.

As the base KPD and backbone network, Convolutional Pose Machines (CPMs) [10], [28] and the first ten convolutional block of the VGG-19BN network [29] are adopted throughout this work.

C. Uncertainty approximation

Belief map uncertainty has not gained enough attention in previous works. The original CPMs and the subsequent related work only use a Gaussian with a fixed standard deviation as ground-truth belief maps for training data.

Our work requires more accurate uncertainty values to capture the ambiguity of object appearance in the image. One way for the network to output the correct uncertainty is to train it with a large amount of data uniformly sampled in the regions of ambiguity, making the network confused, and hope that it will generate belief maps with approximately correct uncertainties due to the confusion. However, that might need many training samples to correctly approximate the distributions [30].

To be more sample-efficient, we choose to approximate the ground-truth uncertainty with a local Gaussian around

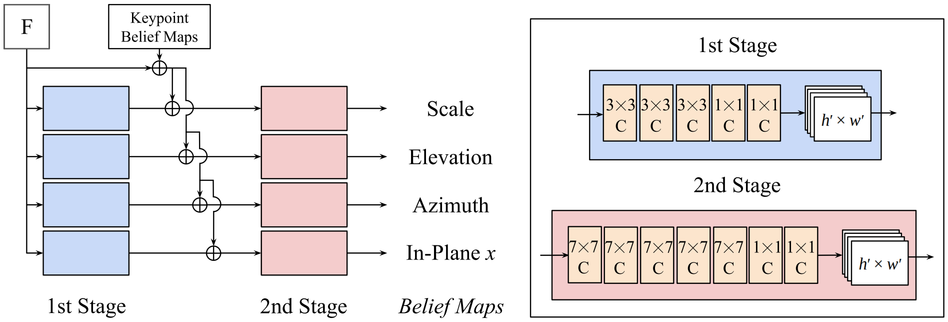

Fig. 3: Architecture of KOSNet. See text for details.

the ground-truth pose using finite differences on synthetic images to detect local ambiguities. More specifically, we consider a generative model where an image I of an object at pose X is generated by a function $f(X)$ with Gaussian pixel noise of standard deviation σ. The posterior belief of the pose X given the training image I is approximated as follows, using the first-order Taylor expansion of $f(X)$:

$$\begin{aligned} p(X|I) &\propto p(I|X)p(X) \\ &\propto \exp(\frac{1}{\sigma^2}||f(X) - I||^2) \\ &\approx \exp(\frac{1}{\sigma^2}||J_0(X - X_0) + f(X_0) - I||^2) \end{aligned} \tag{1}$$

where X_0 is the ground-truth pose of the object in the image I, $J_0 = \frac{\partial f}{\partial X}|_{X=X_0}$ is the derivative of the image generative function f at X_0, and $p(X)$ is a constant as we assume a uniform prior on X. Under this formulation, the posterior belief $p(X|I)$ is locally approximated as a Gaussian with mean $f(X_0)$ and the information (inverse covariance) matrix $\Sigma^{-1} = \sigma^{-2} J_0^T J_0$.

In practice, we use a graphics renderer as the generative function $f(X)$ to generate a predictive image of an object at a specific pose. We approximate the Jacobian J_0 using finite differences $J_0 \approx \frac{f(X_0+\delta)-f(X_0-\delta)}{2\delta}$ by computing pixel differences between two rendered images of the object at poses $X_0 + \delta$ and $X_0 - \delta$. We note that since $X \in SE(3)$, the operation involving them should be interpreted in Lie Group settings [31].

This finite-differencing method on rendered images is generic enough to approximate the uncertainty due to factors such as rotational symmetry, occlusion, and camouflage, and this large uncertainty should be captured by the small difference between the two rendered images $f(X_0 + \delta)$ and $f(X_0 - \delta)$. In the rotational symmetry case, the differences should be exactly the same wrt. small changes δ along the azimuth dimension; the information matrix should be zero and the covariance matrix should be infinite, equivalent to a uniform distribution. However, due to discretization errors of the object mesh and numerical errors of the graphics renderer, the two images are not exactly the same, but their difference is small enough to produce a large uncertainty approximating the uniform density. In the occlusion case, if the handle of a uniformly colored mug is occluded by another object or even by itself, a small pose perturbation around its z-axis will reveal only a small portion of its handle, leading to small pixel differences between the two rendered images resulting in small information matrices. Similarly, in a camouflage situation when a red mug is in front of a red color background, even if its handle is not occluded the differences between two rendered images will still be small due to the similarity of the mug's and the background's color.

IV. Experiments

A. Implementation

Our network is implemented using PyTorch v1.0 [32], [33]. The first ten convolutional blocks, derived from VGG-19BN, were initialized using the weights pretrained on ImageNet [34]. The weights in the subsequent convolutional and batch normalization layers are initialized with Xavier [35] and uniform distributions respectively, and all the biases are initialized with zero. We used 7 as the number of stages for the keypoint belief maps and link vector fields inference. We adopted 36 as the number of belief map's third channel for the elevation, and 72 for azimuth and inplane-x so the resulting discretized step is 5 degrees.

The networks were trained for 60 epochs using synthetic data, and fine-tuned for additional 20 epochs using real data. During the first 60 epochs, additional random augmentations were added to each input image whose values range from 0 to 255: with a probability of 0.7, a Gaussian blur was applied using a 5x5 kernel with the strength sampled uniformly from [0.1, 2.0]; uniform per pixel noise within the range [-20, 20] were added; and with a probability of 0.3, the channels were randomly swapped. The Adam optimizer [36] was used with a base learning rate 0.0016 and weight decay of 0.9. In addition, these learning rates are decayed by 0.3 every 20 epochs. The L2 norm was used as the loss function. The networks were trained using 32 NVIDIA V100 GPUs with batch size 256.

B. Datasets

We evaluated KOSNet's performance and compared its results with our own implementation of DOPE [5] on two datasets: the publicly available YCB-Video dataset [4] and our own custom dataset, the TRI Kitchen v1 dataset.

The TRI Kitchen v1 dataset came from our robotics research efforts at Toyota Research Institute. Unlike the YCB-Video dataset, it includes multiple instances per object category in the scene. The objects are put randomly in the sink, mimicking scenarios with highly cluttered kitchen sinks. It is more challenging than the YCB-Video dataset due to many ambiguities from partially occluded and rotationally symmetric objects. We used three types of *foreground* objects: `corelle_livingware_11oz_mug_red`, `plastic_mug`, and `ikea_dinera_plate_8in`, referred to as the plastic mug, red mug and plate respectively for brevity. We also added *background* objects such as silverware, plastic fruits, napkins, tissues and sponges to the scene as distractors. Multiple configurations of the dishes were captured using RGB and depth from three Intel D415 RealSense cameras. The poses of the *foreground* objects were labeled using a process similar to LabelFusion [37], where the point clouds were concatenated from each camera, and the object labels were estimated by humans using both the 3D point clouds and back projections on the camera images. Each scene was first captured without distractors under three different levels of lighting. Afterwards, distractors were added to the scene (being careful to not disturb the objects) and captured again with the same three levels lighting. For reproducibility, we will make this dataset publicly available, including the high-quality 3D mesh models of *foreground* objects, their Physically-Based Rendering (PBR) materials for generating photo-realistic training images, and the commercially available links to purchase the real physical objects.

C. Training and Evaluation

Following the procedure in [5], we first trained both networks on domain-randomized datasets of synthetic images. We used 60k images per object, four *foreground* instances of the object and up to ten distractors per scene. The PBR graphics engine in Godot [38] was used to render the synthetic images, randomizing the following attributes: poses of all of the objects in the scene, albedo color, metallic, specular and roughness factors for the *foreground* objects, textures, shapes and the number of instances per scene for the distractors, and ambient light energy, directional light's orientation and color, as well as background images for the scenes. For the random background images and the textures on distractors, we used Open Images V5 [39]. For YCB objects, we also included the FallingThings3D dataset [40] to mitigate the domain gap [5].

In addition, we used a small set of real images to fine tune both networks. Although the original DOPE is trained with synthetic data only and has shown its generalization to the data from different domains, we found that adding real images significantly improves its precision on the test datasets, approximately by 20% at a threshold of 2cm for ADD. For objects in the TRI Kitchen v1 dataset, we use one portion of the dataset consisting of 648 real images for fine tuning, leaving the remaining images for evaluation. For the YCB objects, we used a subset of the YCB-Video training dataset for fine-tuning, which consists of 13927 frames, sampled every other five frames from the original training video streams ID 0000-0059.

D. Results

We evaluate the performance of both networks on the YCB-Video test dataset and on the remaining images of the TRI Kitchen v1 dataset that were not used for fine-tuning. For the YCB-Video dataset, we used five out of the 21 YCB objects in our experiments as in [5]: `003_cracker_box`, `004_sugar_box`, `005_tomato_soup_can`, `006_mustard_bottle` and `010_potted_meat_can`.

As shown in Fig. 4, KOSNet achieves comparable results with DOPE on the YCB-Video dataset, but outperforms DOPE on the more challenging TRI Kitchen v1 dataset by a wide margin. Fig. 4 shows the precision of KOSNet and DOPE over varying average distance thresholds on the YCB-Video and TRI Kitchen v1 datasets with area under the curve (AUC). We use the ADD metric as the average distance for all objects except plates, which were evaluated using the ADI metric due to its rotational symmetry [41].

Beside the original network presented in section III-B, named KOSNet-KP2, we also experimented with adding more keypoints to the network, hoping that they can help capture relevant features to improve the network's performance under heavy occlusion. The extended version, named KOSNet-KP7, has five additional channels in the output keypoint belief maps, corresponding to the five additional crossing points between object's 3D bounding box surfaces and the x, y and z axes of the object frame, in addition to the crossing point from the positive z axis already included as the north point N. As shown in Fig. 4, KOSNet-KP7 improves its average precision by approx. 10 to 15 % at the thresholds of 2cm and 4cm for ADD. This improvement especially becomes obvious when the objects are heavily occluded. However, it is not effective in relatively easy scenes like those used for the metrics of `004_sugar_box` and `006_mustard_bottle` in Fig. 4

The ambiguities in the TRI Kitchen v1 dataset due to heavy occlusion and rotational symmetry confuse DOPE whereas KOSNet can still capture the information in its estimated rotation distributions. Fig. 5 and 1 visualize KOSNet outputs on red mugs in the TRI Kitchen v1 dataset, showing the output belief maps of keypoints, links, bounding circles, and rotation angles at the peak locations of the center keypoint heatmap.

Lastly, we conducted experiments to understand our Gaussian uncertainty approximation for angle distributions using the finite-differencing method in section III-C. We compare its results with the results when using a constant standard deviation of 3 degrees, which we call ”spike mode”. Fig. 6 shows KOSNet's estimates of angle distributions on a sequence of synthetic images of one red mug viewed from different angles. Notice that in the ambiguous cases where mug handles are occluded, the heatmaps of azimuth have a

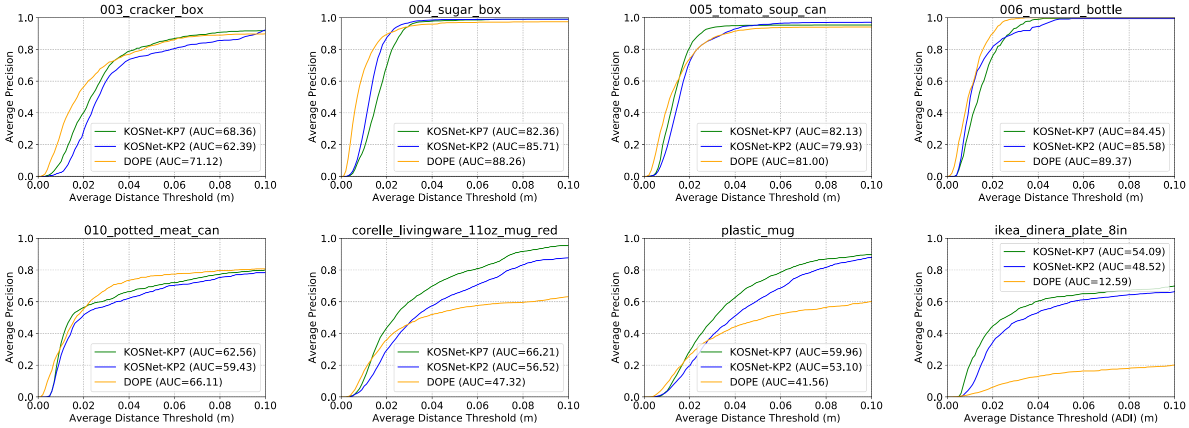

Fig. 4: Precision vs. average distance threshold curves for KOSNet and DOPE on YCB-Video and TRI Kichen v1 datasets.

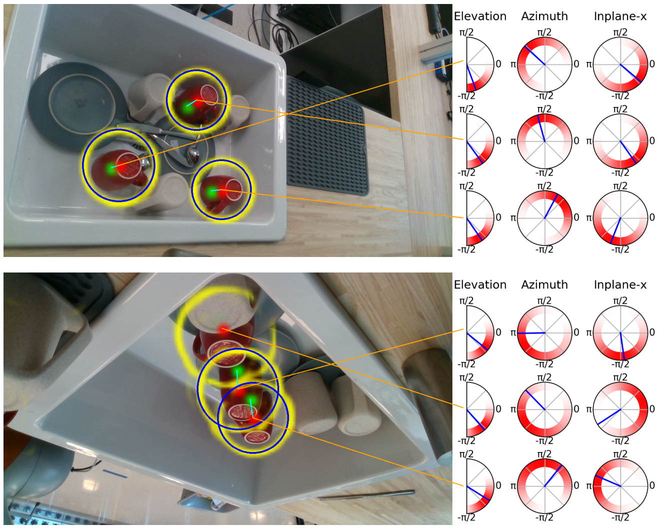

Fig. 5: KOSNet results on TRI Kitchen v1 dataset. Belief maps of center keypoints, north keypoints and bounding circles are overlaid on the input images in red, green and yellow respectively. Belief maps of elevation, azimuth and inplane-x angles at the center keypoints are on the right. Ground truth circles and angles are shown in blue.

wider breadth than those in cases with no occlusion. Interestingly, our method tends to over estimate the uncertainty, whereas the spike mode, while being noisier especially in inplane-x estimates, correctly approximates the distributions in the true intervals. The mean estimates of the spike test mode, however, are biased in some cases where KOSNet's Gaussians are better.

V. Conclusion

The two major paradigms of estimation methods, model-based probabilistic inference and data-driven neural networks, both have their own weaknesses and strengths. By teaching a network to estimate probability distributions, we can combine the strengths of these two vastly different paradigms together. Our KOSNet framework is one step toward that direction. Its probabilistic outputs not only capture the inherent uncertainties due to ambiguities in input information, but also are ready to be fused with other sources of information in any probabilistic framework. We demonstrated its capabilities in handling uncertainties due to heavy occlusion, outperforming a state-of-the-art method. While not demonstrated here, the method can be easily extended to handle objects with discrete rotational symmetry.

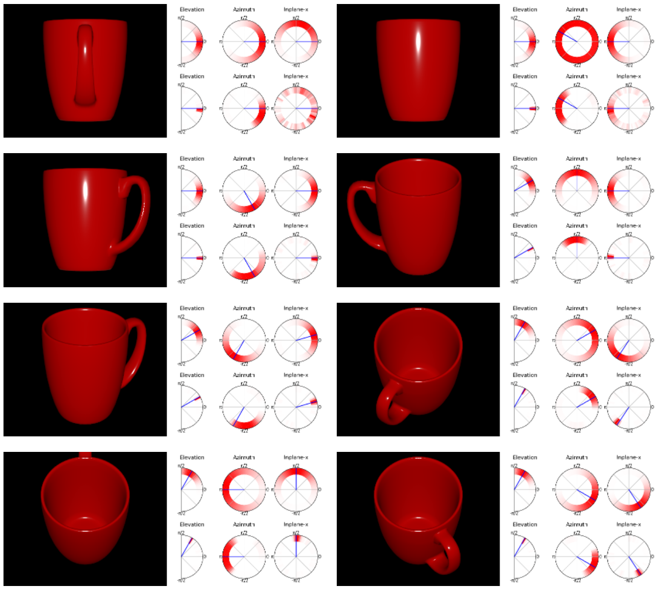

Fig. 6: Comparing KOSNet's estimates of angle distributions in two modes: with approximate standard deviations using finite-differencing (first row heatmaps in each image) and with a constant standard deviation of 3 degrees (second row).

For future work, we aim to apply KOSNet in various vision-based robotics applications involving multisensor fusion and/or fusion of estimates over time. We also aim to understand more deeply the effectiveness and accuracy of its uncertainty estimates, especially compared to related methods, and improve its results by experimenting with different backbone networks and uncertainty representations. In addition, we plan to apply KOSNet to the category-level object pose estimation problem.

References

[1] 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.

[2] T. Hodan, R. Kouskouridas, T.-K. Kim, F. Tombari, K. Bekris, B. Drost, T. Groueix, K. Walas, V. Lepetit, and A. Leonardis, "A Summary of the 4th International Workshop on∼ Recovering 6D Object Pose," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[3] 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.

[4] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," in *Proceedings of Robotics: Science and Systems*, vol. 14, June 2018.

[5] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects," in *Conference on Robot Learning*, 2018, pp. 306–316.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[7] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[8] Z. Zhao, G. Peng, H. Wang, H.-S. Fang, C. Li, and C. Lu, "Estimating 6D Pose From Localizing Designated Surface Keypoints," *arXiv:1812.01387 [cs]*, Dec. 2018.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[10] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional Pose Machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[11] 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, Dec. 2016.

[12] V. Lepetit and P. Fua, "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[13] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D Object Pose Estimation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11214, pp. 19–35.

[14] 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 *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 3848–3856.

[15] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-Dof Object Pose from Semantic Keypoints," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 2011–2018.

[16] B. Tekin, S. N. Sinha, and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, June 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] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D Object Pose Estimation," *arXiv preprint arXiv:1812.02541*, 2018.

[19] 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.

[20] 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.

[21] C. Li, J. Bai, and G. D. Hager, "A Unified Framework for Multi-View Multi-Class Object Pose Estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 254–269.

[22] 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," *arXiv preprint arXiv:1901.04780*, 2019.

[23] E. Corona, K. Kundu, and S. Fidler, "Pose Estimation for Objects with Rotational Symmetry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7215–7222.

[24] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking," *arXiv:1905.09304 [cs]*, May 2019.

[25] S. Tulsiani and J. Malik, "Viewpoints and Keypoints," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, June 2015, pp. 1510–1519.

[26] J. J. Koenderink and A. J. van Doorn, "The internal representation of solid shape with respect to vision," *Biological cybernetics*, vol. 32, no. 4, pp. 211–216, 1979.

[27] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3559–3568.

[28] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[30] N. M. Z. Hashim, Y. Kawanishi, D. Deguchi, I. Ide, H. Murase, A. Amma, and N. Kobori, "Next viewpoint recommendation by pose ambiguity minimization for accurate object pose estimation," in *VISIGRAPP*, 2019.

[31] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[32] B. Steiner, Z. DeVito, S. Chintala, S. Gross, A. Paszke, F. Massa, A. Lerer, G. Chanan, Z. Lin, E. Yang, A. Desmaison, A. Tejani, A. Kopf, J. Bradbury, L. Antiga, M. Raison, N. Gimelshein, S. Chilamkurthy, T. Killeen, L. Fang, and J. Bai, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019.

[33] "Pytorch," https://github.com/pytorch/pytorch.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, "Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–8.

[38] "Godot engine - free and open source 2d and 3d game engine," https://godotengine.org/, (Accessed on 2019/07/01).

[39] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy, "Openimages: A public dataset for large-scale multi-label and multi-class image classification." *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[40] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," *CoRR*, vol. abs/1804.06534, 2018. [Online]. Available: http://arxiv.org/abs/1804.06534

[41] T. Hodaň, J. Matas, and Š. Obdržálek, "On Evaluation of 6D Object Pose Estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 606–619.