Supplementary: iSPA-Net: Iterative Semantic Pose Alignment Network

In this supplementary we outline the various secondary details which provide interesting insight into this work, while also elaborating on various intricacies of our approach.

1 DATA GENERATION

Overview

We select a single 3D template model M_c for each class of object $c \in C$ from ShapeNet dataset [1]. Using a modified version of the rendering pipeline presented by [4], we render the selected template model M_c at various viewpoints to create samples of image $I_2(\theta_2, \phi_2, \psi_2)$. Note that, our reliance on synthetic data is minimal. We use only 8K renders of a single 3D template model. This is in sharp contrast to works such as [3, 4], which use millions of synthetic images in their training pipeline.

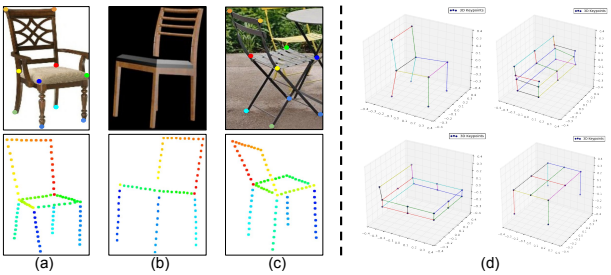

Figure 1: Data Generation.

To train our pose-invariant local descriptors, $\mathcal{L}$, we use the contrastive corresondence loss function, introduced in[2], which requires dense correspondence annotations between image pair (I_1, I_2). For generating such correspondence information, we use automated processing of annotations provided in the Keypoint-5 dataset, released in 3D-INN [5]. We use the 2D skeletal representation, which is based on annotation of sparse 2D skeletal keypoints of real images. As shown in Figure 1, the sparse 2D keypoints are annotated on real images at important joint locations such as leg ends, seat joint etc. For each image sample I_1, we join these sparse keypoints in a fixed order, to obtain the corresponding 2D skeletal frame S_a (second row of Figure 1. a, c). To generate a similar skeletal frame for our rendered 3D object templates in I_2, we manually annotate sparse 3D keypoints for our template objects, as shown in Figure 1 d. Using the projection of these 3D keypoints, based on the viewpoint parameters, we generate 2D keypoints for any rendered image I_2 and also the corresponding 2D skeletal frame S_2 in a similar fashion. Now, by pairing points along this generated skeletal frames (S_1, S_2) of any image pair (I_1, I_2), 2D keypoints on I_1 can be matched to the corresponding keypoints on image I_2. Following this, we generate dense correspondence set for any image pair (I_1, I_2) to improve performance of $\mathcal{L}$, the local descriptors.

.

Figure 2: The template 3D model we use for each class.

Figure 1 shows some qualitative examples of generated annotations for our training. We employ various methods to prune and reform our generated annotations, such as depth based pruning (for I_1), and other methods, explained further in the next section. Although, sometimes the generated annotations are not accurate (Figure 1.c), the correspondence model is able to learn improved view invariant local descriptors $\mathcal{L}$, due to the presence on ample amount of correct noise-free annotations.

Finally, Figure 2 shows the single 3D template model used for each object category.

Keypoint Pruning Mechanism

For Keypoint pruning, we use three main approaches:

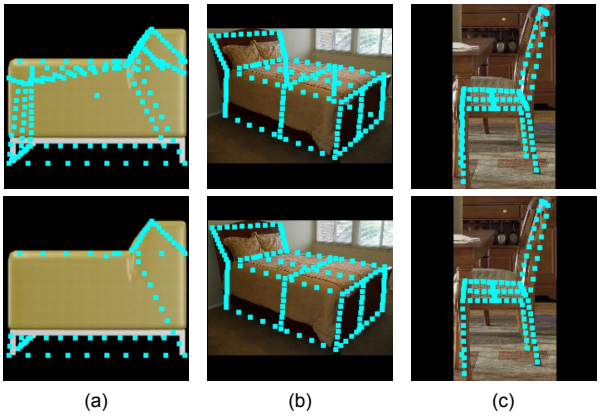

Figure 3: The utility of the three pruning mechanism presented in section 1.

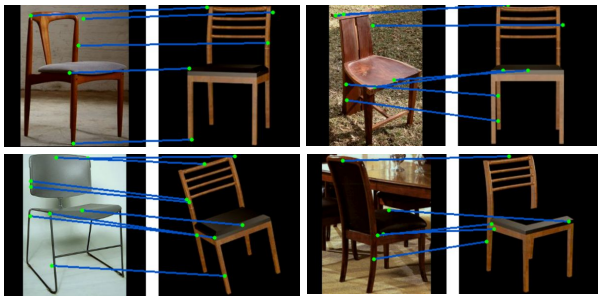

Figure 4: Keypoint Estimation results.

- **Visibility Based Pruning in image** I_2: As the image I_2 is rendered using a template 3D model, a visibility map of the entire object can also be formed easily. Using this visibility map, we prune our points which are not visible from the rendered viewpoint. In figure 3 (a), an example is presented.
- **Seat Presence in Image** I_1: As we know visibility information of all parts of the real image is not available. Hence, we instead use some approximations. We assume that all images of the object are from positive elevation angles. If this assumption holds true, all the leg skeletal keypoints which occur inside the the 2D region covered by the seat are not visible and hence can be pruned out. In figure 3 (b), examples of this pruning mechanism is presented.
- **Self-Occlusion of Legs in Image** I_2 Self occlusion of object legs can be a very frequent occurrence, and almost in all angles, some legs of an object may occlude other legs. We further prune out keypoints on occluded leg, by applying a heuristic approach. First, We approximate the pose quadrant of the object by joining a 2D vector from the back of the seat to the front. Now, based on the pose of the object, which leg can occlude the other is known. This information is then used to prune out self-occluded leg keytpoints. In figure 3 (c), an example is presented.

2 KEYPOINT CORRESPONDENCE

For our proposed approach, the optimality of the learnt local descriptors for giving correspondence map is crucial. In this section, we show some qualitative results to demonstrate keypoint estimation ability of our pose-invariant local descriptors. For each keypoint in the synthetic render, we find the closest matching location in a given real image. In figure 4, we show some qualitative results of keypoint matching between real images and multiple renders of our template 3D model.As we can see, the learnt local descriptors are indeed pose-invariant as they are correctly corresponding to right locations even after considerable change in pose (for example, the bottom right pair).

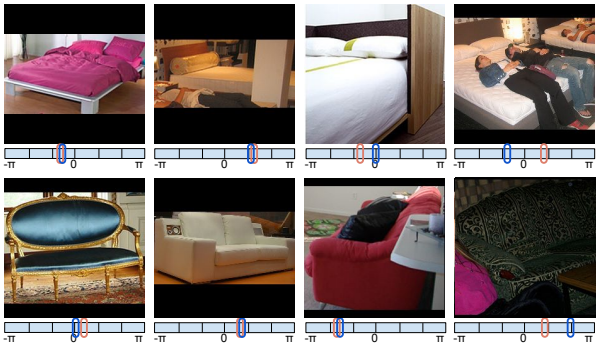

Figure 5: Pose Estimation results. The bar below each image represent the azimuth angle values. The green ellipse represents the Ground Truth pose, and the blue ellipse represents the predicted pose.

3 QUALITATIVE SAMPLES FOR POSE ESTIMATION

In this section, we show some of the results achieved by our network. In Figure 5, we show examples of images from Pascal 3D+ easy test-dataset, along with predicted and annotated azimuth pose angle. The images are arranged in ascending order of angular error in azimuth estimation. As we can see, many times, high error in pose estimation occurs due to extremely poor image quality, due to factors such as lack of illumination, clutter etc.

REFERENCES

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository.* Technical Report arXiv:1512.03012 [cs.GR].

[2] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. 2016. Universal correspondence network. In *NIPS.*

[3] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. 2017. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *CVPR.*

[4] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *CVPR.*

[5] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. 2016. Single image 3-D interpreter network. In *ECCV.*