Object Pose Estimation from Monocular Image Using Multi-view Keypoint Correspondence

Jogendra Nath Kundu, M. V. Rahul(✉), Aditya Ganeshan, and R. Venkatesh Babu

Indian Institute of Science, Bengaluru, India
rahulmv.cs14@rvce.edu.in

Abstract. Understanding the geometry and pose of objects in 2D images is a fundamental necessity for a wide range of real world applications. Driven by deep neural networks, recent methods have brought significant improvements to object pose estimation. However, they suffer due to scarcity of keypoint/pose-annotated real images and hence can not exploit the object's 3D structural information effectively. In this work, we propose a data-efficient method which utilizes the geometric regularity of intraclass objects for pose estimation. First, we learn pose-invariant local descriptors of object parts from simple 2D RGB images. These descriptors, along with keypoints obtained from renders of a fixed 3D template model are then used to generate keypoint correspondence maps for a given monocular real image. Finally, a pose estimation network predicts 3D pose of the object using these correspondence maps. This pipeline is further extended to a multi-view approach, which assimilates keypoint information from correspondence sets generated from multiple views of the 3D template model. Fusion of multi-view information significantly improves geometric comprehension of the system which in turn enhances the pose estimation performance. Furthermore, use of correspondence framework responsible for the learning of pose invariant keypoint descriptor also allows us to effectively alleviate the data-scarcity problem. This enables our method to achieve *state-of-the-art* performance on multiple real-image viewpoint estimation datasets, such as Pascal3D+ and ObjectNet3D. To encourage reproducible research, we have released the codes for our proposed approach (Code: https://github.com/val-iisc/pose_estimation).

Keywords: Pose estimation · 3D structure · Keypoint estimation · Correspondence network · Convolutional neural network

J.N. Kundu, M.V. Rahul and A. Ganeshan—Equal contribution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-11015-4_23) contains supplementary material, which is available to authorized users.

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11131, pp. 298–313, 2019.
https://doi.org/10.1007/978-3-030-11015-4_23

1 Introduction

Estimating 3D pose of an object from a given RGB image is an important and challenging task in computer vision. Pose estimation can enable AI systems to gain 3D understanding of the world from simple monocular projections. While ample variation is observed in the design of objects of a certain type, say chairs, the intrinsic structure or skeleton is observed to be mostly similar. Moreover, in case of 3D objects, it is often possible to unite information from multiple 2D views, which in succession can enhance 3D perception of humans as well as artificial vision systems. In this work, we show how intraclass structural similarity of objects along with multi-view 3D interpretation can be utilized to solve the task of fine-grained 3D pose estimation.

By viewing instances of an object class from multiple viewpoints over time, humans gain the ability to recognize sub-parts of the object, independent of pose and intra-class variations. Such viewpoint and appearance invariant comprehension enables human brain to match semantic sub-parts between different instances of same object category, even from simple 2D perspective projections (RGB image). Inspired from human cognition, an artificial model with similar matching mechanism can be designed to improve final pose estimation results. In this work, we consider a single template model with known keypoint annotations as a 3D structural reference for the object category of interest. Subsequently, Key-point correspondence maps are obtained by matching keypoint-descriptors of synthetic RGB projections from multiple viewpoints, with respect to the spatial descriptors from a real RGB image. Such keypoint-correspondence maps can provide the geometric and structural cues useful for pose estimation.

The proposed pose estimation system consists of two major parts; (1) A Fully Convolutional Network which learns pose-invariant local descriptors to obtain keypoint-correspondence, and (2) A pose estimation network which fuses information from multiple correspondence maps to output the final pose estimation result. For each object class, we annotate a single template 3D model with sparse 3D keypoints. Given an image, in which the object's pose is to be estimated, first it is paired with multiple rendered images from different viewpoints of the template 3D model (see Fig. 1). Projections of the annotated 3D keypoints is tracked on the rendered synthetic images to provide ground-truth for learning of efficient key-point descriptor. Subsequently, keypoint-correspondence maps are generated for each image pair using correlation of individual keypoint descriptor (from rendered image) to the spatial descriptors obtained from the given image.

Recent works [10,12,21] show that deep neural networks can effectively merge information from multiple 2D views to deliver enhanced view estimation performance. These approaches require multi-view projections of the given input image to exploit the multi-view information. But in the proposed approach, we attempt to take advantages of multi-view cue by generating correspondence map from the single-view real RGB image by comparing it against multiview synthetic renders. This is achieved by feeding the multi-view keypoint correspondence maps through a carefully designed fusion network (convolutional neural network) to obtain the final pose estimation results. Moreover, by fusing information from multiple viewpoints, we show significant improvement in pose estimation, making our pose estimation approach *state-of-the-art* in competitive real-image datasets, such as Pascal3D+ [32] and ObjectNet3D [31]. In Fig. 1, a diagrammatic overview of our approach is presented.

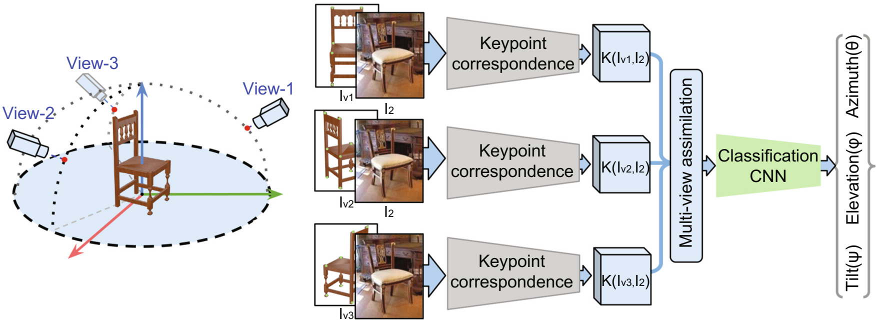

Fig. 1. Illustration of the proposed pipeline. Given a real image I_2, it is paired with multiple 2D views of a template 3D model with annotated keypoints. For each pair of images, keypoint correspondence maps are generated, represented by $K(I_{vk}, I_2)$. Finally, the pose estimator network assimilates information from all correspondence maps to predicts the pose parameters.

Many recent works [8,29,32], have utilized deep neural networks for 3D object understanding and pose estimation. However, these approaches have several drawbacks. Works such as [25,30] achieve improved pose estimation performance by utilizing a vast amount of synthetic data. This can be a severe bottleneck when an extensive repository of diverse 3D models for a specific category is unavailable (as in case of novel object-classes, such as mechanical parts, abstract 3D models etc.). Additionally, 3D-INN [30] require a complex keypoint-refinement module that, while being remarkable at keypoint estimation, shows sub-optimal performance for viewpoint estimation, when compared against current state-of-the-art models. We posit that it is essential to explore and exploit strong 3D-structural object priors to alleviate various general issues, such as data-bottleneck and partial-occlusion, which are observed in object viewpoint estimation. Moreover, our approach has two crucial advantages. Firstly, our keypoint correspondence map captures relation between the keypoint and the entire 2D spatial view of the object in a given image. That is, the correspondence map not only captures information regarding spatial location of keypoint in the given image, but also captures various relations between the keypoint and other semantic-parts of the object. In Fig. 2, we show the obtained correspondence map for varied keypoints, and provide evidence for this line of reasoning. Secondly, our network fuses the correspondence map of each keypoint from multiple views.

To summarize, our main contributions in this work include: (1) a method for learning pose-invariant local descriptors for various object classes, (2) a keypoint correspondence map formulation which captures various explicit and implicit relations between the keypoint, and a given image, (3) a pose estimation network which assimilates information from multiple viewpoints, and (4) state-of-the-art performance on real-image object pose estimation datasets for indoor object classes such as ‘Chair’, ‘Sofa’, ‘Table’ and ‘Bed’.

2 Related Work

Local Descriptors and Keypoint Correspondence: A multitude of work propose formulations for local discriptors of 3D objects, as well as 2D images. Early methods employed hand-engineered local descriptors like SIFT or HOG [2,3,15,27] to represent semantic part structures useful for object comprehension. With the advent of deep learning, works such as [5,9,23,33] have proposed effective learning methods to obtain local descriptor correspondence in 2D images. Recently, Huang *et al.* [11] propose to learn local descriptors for 3D objects following deep multi-view fusion approach. While this work is one of our inspirations, our method differs in many crucial aspects. We do not require extensive multi-view fusion of local descriptors as performed by Huang *et al.* for individual local points. Moreover we do not rely on a large repository of 3D models with surface segmentation information for generalization. For effective local descriptor correspondence, Universal Correspondence Network [5] formulate an optimization strategy for learning robust spatial correspondence, which is used in coherence with an active hard-mining strategy and a convolutional spatial transformer (STN). While [5] learn geometric and spatial correspondence for task such as semantic part matching, we focus on the learning procedure of their approach and adapt it for learning our pose-invariant local descriptors.

Multi-view Information Assimilation: Borotschnig *et al.* [4], and Paletta *et al.* [18] were one of the earliest works to show the utility of multi-view information for improving performance on tasks related to 3D object comprehension. In recent years, multiple innovative network architectures, such as [10,21] have been proposed for the same. One of the earliest works to combine deep learning with multi-view information assimilation, [24] showed that 2D image-based approaches are effective for general object recognition tasks, even for 3D models. They proposed an approach for 3D object recognition based on multiple 2D projections of the object, surpassing previous works which were based on other 3D object representations such as voxel and mesh format. In [20], Qi *et al.* give a comprehensive study on the voxel-based CNN and multi-view CNN for 3D object classification. Apart from object classification, multi-view approach is seen to be useful for a wide variety of other tasks, such as learning local features for 3D models [11], 3D object shape prediction [28] etc. In this work, we use multi-view information assimilation for object pose estimation in a given monocular RGB image using multiple views of a 3D template model.

Object Viewpoint Estimation: Many recent works [17,19] use deep convolutional networks for object viewpoint estimation. While works such as [29] attempt pose estimation along with keypoint estimation, an end-to-end approach solely for 3D pose estimation was first proposed by RenderForCNN [25]. Su *et al.* [25] proposed to utilize vast amount of synthetic rendered data from 3D CAD models with dataset specific cues for occlusion and clutter information, to combat the lack of pose annotated real data. In contrast, 3D Interpreter Network (3D-INN) [30] propose an interesting approach where 3D keypoints and view is approximated by minimizing a novel re-projection loss on the estimated 2D keypoints. However, the requirement of vast amount of synthetic data is a significant bottleneck for both the works. In comparison, our method relies on the presence of a single synthetic template model per object category, making our method significantly data efficient and far more scalable. This is an important pre-requisite to incorporate the proposed approach for novel object classes, where multiple 3D models may not exists. Recently, Grabner *et al.* [8] estimate object pose by predicting the vertices of a 3D bounding box and solving a perspective-n-point problem. While achieving state-of-the-art performance in multiple object categories, they could not surpass performance of [25] on the challenging indoor object classes such as 'chair','sofa', and 'table'. It is essential to provide stronger 3D structural priors to learn pose estimation under data scarcity scenario for such complex categories. The structural prior is effectively modeled in our case by keypoint correspondence and multi-view information assimilation.

3 Approach

This section consist of 3 main parts: in Sect. 3.1, we present our approach for learning pose invariant local descriptors, Sect. 3.2 explains how the keypoint correspondence maps are generated, and Sect. 3.3 explains our regression network, along with various related design choices. Finally, we briefly describe our data generation pipeline in Sect. 3.4.

3.1 Pose-Invariant Local Descriptors

To effectively compare given image descriptors with the keypoint descriptors from multi-view synthetic images, our method must identify various sub-parts of the given object, invariant to pose and intra-class variation. To achieve this we train a convolutional neural network (CNN), which takes an RGB image as input and gives a spatial map of local descriptors as output. That is, given an image I_1 of size $h \times w$, our network predicts a spatial local descriptor map L_{I_1} of size $h \times w \times d$, where the d-dimensional vector at each spatial location is treated as the corresponding local descriptor.

Following the approach of other established method [5,11], we use the CNN to form two branches of a Siamese architecture with shared convolutional parameters. Now, given a pair of images I_1 and I_2 with annotated keypoints, we pass them through the siamese network to get the spatial local descriptor maps L_{I_1} and L_{I_2} respectively. The annotated keypoints are then used to generate positive and negative correspondence pairs, where a positive correspondence pair refers to a pair of points $I_1(x_k, y_k), I_2(x'_k, y'_k)$ such that they represent a certain semantic keypoint. In [5], authors present the correspondence contrastive loss, which is used to reduce the distance between the local descriptors of positive correspondence pairs, and increase the distance for the negative pairs. Let $\mathbf{x_i} = (x_k, y_k)$ and $\mathbf{x'_i} = (x'_k, y'_k)$ represent spatial locations on I_1 and I_2 respectively. The correspondence contrastive loss can be defined as,

$$Loss = \frac{1}{2N} \sum_{i}^{N} \{ s_i \| L_{I_1}(\mathbf{x}) - L_{I_2}(\mathbf{x'}) \|^2 + \\ (1 - s_i) \max(0,\ m - \| L_{I_1}(\mathbf{x}) - L_{I_2}(\mathbf{x'}) \|^2) \} \quad (1)$$

where N is the total number of pairs, $s_i = 1$ for positive correspondence pairs, and $s_i = 0$ for negative correspondence pairs.

Chief benefit of using a correspondence network is its utility to combat data-scarcity. Given N samples with keypoint annotation, we can generate ${}^{N}C_2$ training samples for training the local descriptor representations. The learned local descriptors do most of the heavy lifting by providing useful structural cues for 3D pose estimation. This helps us avoid extensive usage of synthetic data and the common pitfalls associated with it, such as domain shift [14] while testing on real samples. Compared to state-of-the-art works [25,30], where millions of synthetic data samples were used for effecting training, we use only $8k$ renders of a single template 3D model per class (which is less than 1% of the data used by [25,30]). Another computational advantage we observe is in terms of run-time efficiency. Given a single image, we estimate the local descriptors for all the visible points on the object. This is in stark contrast to Huang *et al.* [11], where multiple images were used for generating local descriptors for each point of the object.

In most cases such as in [30], objects are represented by a sparse set of keypoints. Learning feature descriptors for only a few sparse semantic keypoints has many disadvantages. In such case, the models fails to learn efficient descriptors for spatial regions away from the defined semantic keypoint locations. However, information regarding parts away from these keypoints can also be useful for pose estimation. Hence, we propose to learn proxy-dense local descriptors to obtain more effective correspondence maps (see Fig. 3b and c). This also allows us to train the network more efficiently by generating enough amount of positive and negatives correspondence pairs. For achieving this objective, we generate dense keypoints for all images, details of which are presented in Sect. 3.3.

Correspondence Network Architecture: The siamese network contains two branches with shared weights. It is trained on the generated key-point annotations (details in Sect. 3.3) using the loss, Eq. 1 described above. For the Siamese network, we employ a standard Googlenet [26] architecture with imagenet pretrained weights. Further, to obtain spatially aligned local features L_I, we use a convolutional spatial transformation layer after *pool*4 layer of googlenet architecture, as proposed in UCN [5]. The use of convolutional spatial transformation layer is found to be very useful for semantic part correspondence in presence of reasonably high pose and intra-class variations.

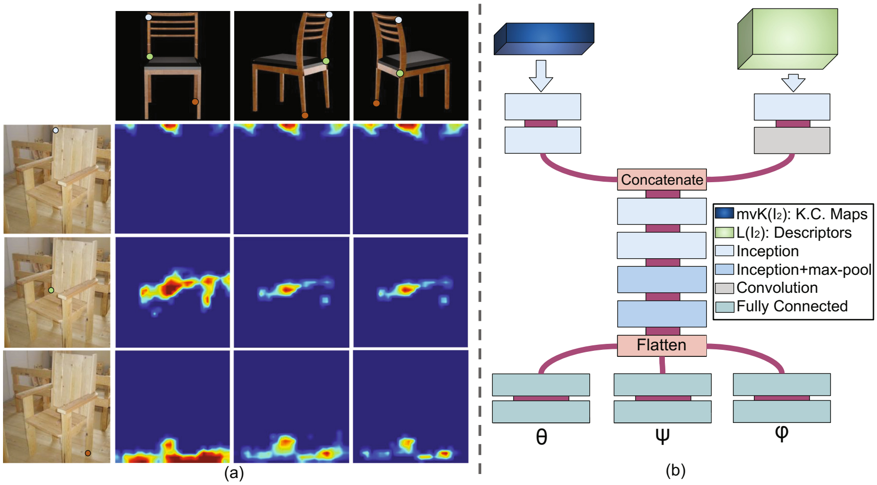

Fig. 2. (a) The Keypoint Correspondence map generated by our approach. The top row shows the template 3D model from 3 Views where 3 different keypoints are highlighted. First column shows the real image where pose has to be estimated. As we can see, Keypoints have lesser ambiguity when looked from views where they are clearly visible (For eg., back-leg keypoint, View 2 and 3). (b) The architecture of our pose estimator network.

3.2 Keypoint Correspondence Maps

The CNN introduced in the previous section provides a spatial local descriptor map L_{I_1} for a rendered synthetic image I_1. Now, using the keypoint annotations rendered from the 3D template model, we want to generate a spatial map, which can capture the location of corresponding keypoint in a given real image, I_2. To achieve this we propose to utilize pairwise descriptor correlation between both the images. Let, L_{I_1} is of size $h \times w \times d$, and x_k represents a keypoint in I_1. Now our goal is to estimate a correspondence map of keypoint x_k for the real image I_2. By taking correlation of the local descriptor at x_k, $L_{I_1}(x_k)$ with all locations (i', j') of the spatial local descriptor for image I_2, i.e. L_{I_2}, correspondence maps are obtained for each keypoint, x_k. Using max-out Hadamard product H, we compute the pairwise descriptor correlation for any (i', j') in I_2 and x_k in I_1 as follows:

$$H(x_k, (i, j)) = \max(0, L_{I_1}(x_k)^T L_{I_2}(i', j'))$$

$$C_{x_k, I_2}(L_{I_1}(x_k),\ L_{I_2}(i', j')) = \frac{\exp^{H(x_k, i, j)}}{\sum_{p,q} \exp^{H(x_k, p, q)}}$$

As the learned local descriptors are unit normalized, the max-out Hadamard product $H(x_k, (i, j))$ represents only positive correlation between local descriptor at x_k with local descriptors of all locations (i, j) in image I_2. By applying soft-max on the entire map of rectified Hadamard product, multiple high correlation values will be suppressed by making the highest correlation value more prominent in the final correspondence map. Such normalization step is in line with the traditionally used second nearest neighbor test proposed by Lowe *et al.* [16]. Using the above formulation, keypoint correspondence maps C_{x_k,I_2} is generated for a set of sparse structurally important keypoints $x_k, fork = 1, 2, ..., N$ in image I_1. The structurally important keypoints that we use for each object class are the same as the ones used by [30]. Finally, We use the structurally important keypoint set for individual object category as defined by Wu *et al.* [30]. Finally the stacked correspondence map for all structural keypoints of I_1 computed for image I_2 is represented by $K(I_1, I_2)$. Here $K(I_1, I_2)$ is of size $N \times h \times w$, where N is the number of keypoints.

As explained earlier, our keypoint correspondence map computes the relation between the keypoint x_k in I_1 and all the points (i, j) in I_2. In comparison to [30], where a location heatmap is predicted for each keypoint, our keypoint correspondence map captures the interplay between different keypoints as well. This in turn acts as an important cue for final pose estimation. Figure 1 shows keypoint correspondence maps generated by our approach, which clearly provide evidence of our claims.

3.3 Multi-view Pose Estimation Network

With the structural cues for object in image I_2 provided by the keypoint correspondence set $K(I_1, I_2)$, we can estimate pose of the object more effectively. In our setup, I_1 is a synthetically rendered image of the template 3D model with the tracked 2D keypoint annotations, and I_2 is the image of interest where the pose has to be estimated. It is important to note, that $K(I_1, I_2)$ contains information regarding relation between the keypoints $x_k, k = 1, 2, ..., N$ in I_1 with respect to the image I_2. However, as I_1 is a 2D projection of the 3D template object, it is possible that some keypoints are self occluded, or only partially visible. For such keypoints C_{x_k,I_2} would contain noisy and unclear correspondence. As mentioned earlier, the selected keypoints are structurally important and hence lack of information of any of them can hamper the final pose estimation performance.

To alleviate this issue, we propose to utilize a multi-view pose estimation approach. We first render the template 3D model from multiple viewpoints $I_{v1}, I_{v2}, ...I_{vm}$ considering m viewpoints. Then, the keypoint correspondence set is generated for each view by pairing I_{vk} with I_2 for all k. Finally, information from multiple views is combined together by concatenating all the correspondence sets to form a fused Multi-View Correspondence set, represented by $mvK(I_2)$. Here, $mvK(I_2)$ is of size $(m \times N, h, w)$; where m is the number of views, and N is the number of structurally important keypoints. subsequently, $mvK(I_2)$ is supplied as an input to our pose estimation network which effectively combines information from multiple-views of the template object to infer the required structural cues. For a given m, we render $I_{v1}, I_{v2}, ...I_{vm}$ from fixed viewpoints, $v_k = (360/m \times k, 10, 0)$ for $k = 1, 2, ...m$; where v_k represents a tuple of azimuth, elevation and tilt angles in degree.

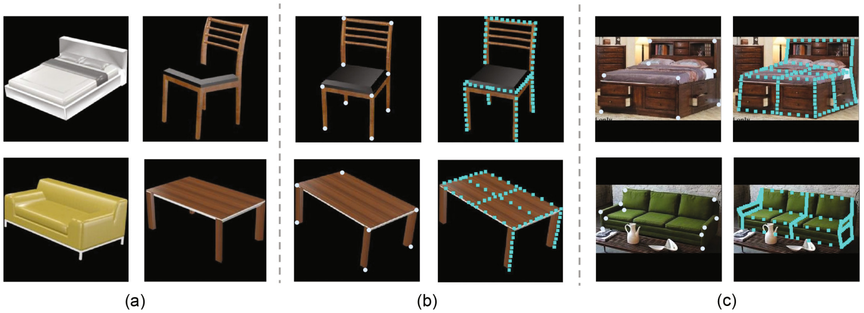

Fig. 3. (a) The single 3D template model selected for each class. (b) Template models are annotated with sparse 3D keypoints, which are projected to 2D keypoints in each rendered image. From these keypoints, dense keypoint annotation is generated by sampling along the skeleton. (c) Similar process is used on real image datasets where sparse 2D keypoint annotation has been provided.

In Fig. 2b, the architecture of our pose estimation network is outlined. Empirically, we found Inception Layer to be most efficient in terms of performance for memory footprint. We believe, multiple receptive fields in the inception layer help the network to learn structural relations at varied scales, which later improves pose estimation performance. For effective modeling, we consider deeper architecture with reduced number of filters per convolutional layer. Here, the pose estimation network classifies the three Euler angles, namely azimuth (θ), elevation (ϕ), and tilt (ψ). Following [25], we use the Geometric Structure Aware Classification Loss for effective estimation of all the three angles.

As a result of proxy-dense correspondence, Pose-Invariant local descriptor $L(I_2)$ has information about dense keypoints. But $mvK(I_2)$ leverages information only from the sparse set of structurally important keypoints. Therefore, we also explore whether $L(I_2)$ can also be utilized to improve the final pose estimation performance. To achieve this, we concatenate convolution-processed feature map of $L(I_2)$ with inception-processed features of $mvK(I_2)$ to form the input to our pose-estimation network. This brings us to our final state-of-the-art architecture. Various experiments are performed in Sect. 4.1, which outline the benefits of each of the design choices.

3.4 Data Generation for Local Descriptors

Learning an efficient pose-invariant keypoint descriptor requires presence of ground-truth positive correspondence pair in sufficient amount. For each real image, we generate an ordered set of dense keypoints by forming a skeletal frame of the object from the available sparse keypoint annotations provided in Keypoint-5 dataset [30]. To obtain dense positive keypoint pairs, we sample additional points along the structural skeleton lines obtained from the semantic sparse keypoints for both real and synthetic image. Various simple keypoint pruning methods based on seat presence, self-occlusion etc. are used to remove noisy keypoints (more detail in supplementary). Figure 3(c) shows some real images where dense keypoint annotation is generated from available sparse keypoint annotation as described above.

For our synthetic data, a single template 3D model (per category) is manually annotated with a sparse set of 3D keypoints. These models are shown in Fig. 3a. Using a modified version of the rendering pipeline presented by [25], we render the template 3D model and project sparse 2D keypoints from multiple views to generate synthetic data required for the pipeline. Similar skeletal point sampling mechanism as mentioned earlier is used to from dense keypoint annotation for each synthetic image as shown in Fig. 3b (more details in supplementary).

4 Experiments

In this section, we evaluate the proposed approach with other state-of-the-art models for multiple tasks related to viewpoint estimation. Additionally, multiple architectural choices are validated by performing various ablation on the proposed multi-view assimilation method.

Datasets: We empirically demonstrate *state-of-the-art* or competitive performance when compared to several other methods on two public datasets. *Pascal3D+* [32]: This dataset contains images from Pascal [6] and ImageNet [22] set labeled with both detection and continuous pose annotations for 12 rigid object categories. *ObjectNet3D* [31]: This dataset consists of 100 diverse categories, 90,127 images with 201,888 objects. Due to the requirement of keypoints, keypoint-based methods can be evaluated only on object-categories with available keypoint annotation. Hence, we evaluate our method on 4 categories from these dataset namely, Chair, Bed, Sofa and Dining-table (3 on Pascal3D+, as it does not contain Bed category). We evaluate our performance for the task of object viewpoint estimation, and joint detection and viewpoint estimation.

Metrics: Performance in object viewpoint estimation is measured using *Median Error* ($MedErr$) and *Accuracy at*θ (Acc_θ), which were introduced by Tulsiani *et al.* [29]. $MedErr$ measures the median geodesic distance between the predicted pose and the ground-truth pose (in degree) and Acc_θ measures the % of images where the geodesic distance between the predicted pose and the ground-truth pose is less than θ (in radian). While previous works evaluate Acc_θ with $\theta = \pi/6$ only, we evaluate Acc_θ with smaller θ as well (i.e. for $\theta = \pi/8$ and $\pi/12$) to highlights our models ability to deliver more accurate pose estimates. Finally, to evaluate performance on joint detection and viewpoint estimation, we use *Average Viewpoint Precision at 'n' views*(AVP-n) metric as introduced in [32].

Training details: We use ADAM optimizer [13] having a learning rate of 0.001 with minibatch-size 7. For each object class, we assign a *single* 3D model from Shapenet Repository as the object template. The local feature descriptor network is trained using 8,000 renders of the template 3D model (per class), along with real training images from Keypoint-5 and Pascal3D+. Dense correspondence annotations are generated for this segment of the training (refer Sect. 3.4).

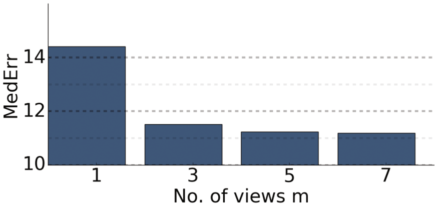

Fig. 4. $Acc_{\pi/6}$ vs number for views 'm' used for the multi-view information assimilation in our method.

Table 1. Ablation on our model for validating the utility of $L(I_2)$ in improving pose estimation.

Ours$_{\mathbf{N}}$	$MedErr$	$Acc_{\pi/6}$
w/o $L(I_2)$	11.51	0.74
with $L(I_2)$	**9.52**	**0.80**

Finally, the pose estimation network is trained using Pascal3D+ or ObjectNet3D datasets. This training regime provides us our normal model, labeled **Ours$_{\mathbf{N}}$**. Additionally, to compare against RenderForCNN [25] in the presence of synthetic data, we construct a separate training regime, where the synthetic data provided by RenderForCNN [25] is also utilized for training the pose estimation network. The model trained in this regime is labeled **Ours$_{\mathbf{D}}$**.

4.1 Ablative Analysis

In this section, we focus on evaluating the utility of various components of our method for object viewpoint estimation. Our ablative analysis focuses on the Chair category. The Chair category, having high intra-class variation, is considered one of the most challenging classes and provides minimally biased dataset for evaluating ablations of our architecture. For all the ablations, the network is trained on the train-subset of ObjectNet3D and Pascal-3D+ dataset. We report our ablation statistics on the easy-test-subset of Pascal3D+ for chair category, as introduced by [29].

First, we show the utility of the Multi-view information assimilation by performing ablations on the number of views 'm'. In Fig. 4, we evaluate the $MedErr$ for our method with 'm' varying from 1 to 7. Note that we do not utilize the local descriptors $L(I_2)$ in this setup and the pose estimator uses only the multi-view keypoint correspondence maps $mvK(I_2)$ as input. As the figure shows, additional information from multiple views is crucial. For having an computationally efficient yet effective system, we use $m = 3$ for all the following experiments. Next, it is essential to ascertain the utility of local descriptors $L(I_2)$ in improving our performance. In Table 1, we can clearly observe increment in performance due to usage of $L(I_2)$ along with $mvK(I_2)$. Hence, in our final pipeline, the pose estimator network is designed to include the $L(I_2)$ as an additional input.

4.2 Object Viewpoint Estimation

In this section, we evaluate our method against other *state-of-the-art* approaches for the task of viewpoint estimation. Similar to other keypoint-based pose estimation works, such as 3D-INN [30], we conduct our experiments on all object classes where 2D-keypoint information is available.

Pascal3D+: Table 2 compares our approach to other *state-of-the-art* methods, namely Grabner *et al.* [8] and RenderForCNN [25]. The table shows, our best performing method **Ours$_{\mathbf{D}}$** clearly outperform other established approaches on pose estimation task.

Table 2. Performance for object viewpoint estimation on PASCAL 3D+ [32] using ground truth bounding boxes. Note that *MedErr* is measured in degree.

Category	Su *et al.* [25]		Grabner *et al.* [8]		**Ours$_{\mathbf{D}}$**	
	$Acc_{\pi/6}$	*MedErr*	$Acc_{\pi/6}$	*MedErr*	$Acc_{\pi/6}$	*MedErr*
Chair	0.86	9.7	0.80	13.7	0.83	**8.84**
Sofa	0.90	**9.5**	0.87	13.5	**0.90**	10.74
Table	0.73	10.8	0.71	11.8	**0.87**	**6.00**
Average	0.83	10.0	0.79	13.0	**0.87**	**8.53**

ObjectNet3D: As none of the existing works have shown results on ObjectNet3D dataset, we trained RenderForCNN using the synthetic data and code provided by the authors Su *et al.* [25] for ObjectNet3D. Table 3 compares our method against RenderForCNN on various metrics for viewpoint estimation. RenderForCNN, which is trained using 500,000 more samples of synthetic images, still shows poor performance than the proposed method **Ours$_{\mathbf{N}}$**.

Table 3. Evaluation on viewpoint estimation based tasks on the ObjectNet3D dataset. Note that **Ours$_{\mathbf{N}}$** is trained with no synthetic data, where as Su *et al.* is trained with 500,000 synthetic images (for all 4 classes).

Method	Metric	Chair	Sofa	Table	Bed	Avg.
Object viewpoint estimation						
MedErr	Su *et al.* [25]	9.70	8.45	4.50	7.21	7.46
	Ours$_{\mathbf{N}}$	**7.94**	**3.55**	**3.33**	**7.10**	**5.48**
$Acc_{\pi/6}$	Su *et al.* [25]	0.75	0.90	0.77	0.77	0.80
	Ours$_{\mathbf{N}}$	**0.81**	**0.92**	**0.90**	**0.82**	**0.86**
$Acc_{\pi/8}$	Su *et al.* [25]	0.71	0.89	0.72	0.75	0.76
	Ours$_{\mathbf{N}}$	**0.78**	**0.90**	**0.88**	**0.79**	**0.83**
$Acc_{\pi/12}$	Su *et al.* [25]	0.64	0.80	0.68	0.72	0.71
	Ours$_{\mathbf{N}}$	**0.72**	**0.86**	**0.84**	**0.74**	**0.79**
Joint object detection and pose estimation						
AVP-4	Su *et al.* [25]	**23.9**	69.8	53.5	65.1	53.1
	Ours$_{\mathbf{N}}$	22.1	**71.9**	**65.7**	**71.6**	**57.8**

4.3 Joint Object Detection and Viewpoint Estimation

Now, for this task, our pipeline is used along with object detection proposal from R-CNN [7] using MCG [1] object proposals to estimate viewpoint of objects in each detected bounding box, as also followed by V&K [29]. Note that the performance of all models in this task is affected by the performance of the underlying Object Detection module, which varies significantly among classes.

Pascal3D+: In Table 4, we compare our approach against other *state-of-the-art* keypoint-based methods, namely, 3D-INN [30] and V&K [29]. The metric comparison shows superiority of our method, which in turn highlights ours' ability to predict pose even with noisy object localization.

Table 4. Comparison of $\mathbf{Ours_D}$ with other keypoint-based pose estimation approaches for the task of joint object detection and viewpoint estimation on Pascal3D+ dataset.

AVP−4	Chair	Sofa	Table	Avg.
V&K [29]	25.1	43.8	24.3	31.1
3D-INN [30]	23.1	**45.8**	–	–
$\mathbf{Ours_D}$	**26.0**	41.9	**26.5**	**31.5**

ObjectNet3D: Here, we trained RenderForCNN using the synthetic data and code provided by the authors Su *et al.* [25]. Table 3 compares our method against RenderForCNN on the AVP-n metric.

Table 3 clearly demonstrates sub-optimal performance of RenderForCNN on ObjectNet3D. This is due to the fact that, the synthetic data provided by the authors Su *et al.* [25] is overfitted to the distribution of Pascal3D+ dataset. This leads to a lack of generalizability in RenderForCNN, where a mismatch in the synthetic and real data distribution can significantly lower its performance. Moreover, Table 3 not only presents our superior performance, but also highlights the poor generalizability of RenderForCNN.

4.4 Analysis

Here, we present analysis of results on additional experiments to highlight the chief benefits of the proposed approach.

Effective Data Utilization: To highlight the effective utilization of data in our method, we compare $\mathbf{Ours_N}$ against other methods trained without utilizing any synthetic data. For this experiment, we trained RenderForCNN without utilizing synthetic data and compare it to $\mathbf{Ours_N}$ in Table 5. The Table not only provides evidence for high data dependency of RenderForCNN, it also highlights our superior performance against Grabner *et al.* [8] even in limited data scenario.

Table 5. Performance for object viewpoint estimation on PASCAL 3D+ [32] using ground truth bounding boxes.

Category	Su *et al.* [25]		Grabner *et al.* [8]		**Ours$_{\mathbf{N}}$**	
	$Acc_{\pi/6}$	*MedErr*	$Acc_{\pi/6}$	*MedErr*	$Acc_{\pi/6}$	*MedErr*
Chair	0.70	11.30	0.80	13.70	**0.80**	**9.52**
Sofa	0.65	14.45	**0.87**	13.50	0.80	**9.96**
Table	0.70	**5.80**	0.71	11.80	**0.83**	6.00
Average	0.68	10.51	0.79	13.0	**0.81**	**8.49**

Higher Precision of Our Approach: Table 6 compares **Ours$_{\mathbf{N}}$** to RenderForCNN [25] on stricter metrics, namely $Acc_{\pi/8}$ and $Acc_{\pi/12}$. Further, we show a plot of Acc_θ vs θ in Figs. 5 and 6 for multiple classes in both Pascal3D+ and ObjectNet3D dataset. Compared to the previous state-of-the-art model, we are able to substantially improve the performance with harsher θ bounds, indicating that our model is more precise on estimating the pose of objects on both 'Chair' and 'Table' category. This firmly establishing the superiority of our approach for the task of fine-grained viewpoint estimation.

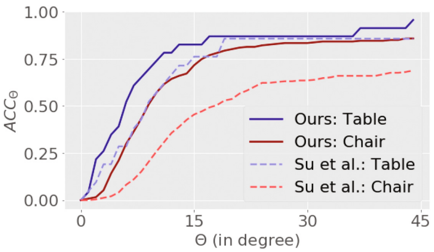

Fig. 5. Acc_θ vs θ in Pascal3D+.

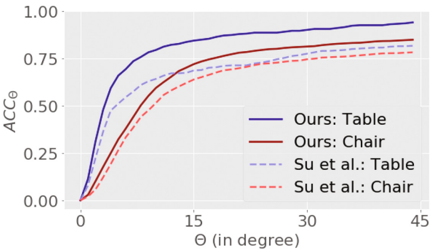

Fig. 6. Acc_θ vs θ in ObjectNet3D.

Table 6. Comparison of our approach to existing *state-of-the-art* methods for stricter metrics (On Pascal3D). For evaluating RenderForCNN on pascal3D+, the model provided by the authors Su *et al.* has been used. The best value has been highlighted in **bold**, and the second best has been colored red.

Metric	Method	Chair	Sofa	Table	Avg.
$Acc_{\pi/8}$	Su *et al.* [25]	0.59	0.79	0.68	0.68
	Ours$_{\mathbf{N}}$	0.78	0.77	0.83	0.79
	Ours$_{\mathbf{D}}$	**0.81**	**0.85**	**0.86**	**0.84**
$Acc_{\pi/12}$	Su *et al.* [25]	0.42	0.69	0.60	0.57
	Ours$_{\mathbf{N}}$	0.69	0.67	0.83	0.73
	Ours$_{\mathbf{D}}$	**0.72**	**0.75**	**0.83**	**0.76**

5 Conclusions

In this paper, we present a novel approach for object viewpoint estimation, which combines keypoint correspondence maps from multiple views, to achieve state-of-the-art results on standard pose estimation datasets. Being data-efficient, our method is suitable for large-scale or novel-object based real world applications. In future work, we would like to make the method weakly-supervised as obtaining keypoint annotations for novel object categories is non-trivial. Finally, the pose-invariant local descriptors show a promise of usability in other tasks, which will also be explored in the future.

References

1. Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
2. Aubry, M., Maturana, D., Efros, A.A., Russell, B.C., Sivic, J.: Seeing 3D chairs: exemplar part-based 2D–3D alignment using a large dataset of cad models. In: CVPR (2014)
3. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR (2005)
4. Borotschnig, H., Paletta, L., Prantl, M., Pinz, A.: Appearance-based active object recognition. Image Vis. Comput. **18**(9), 715–727 (2000)
5. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In: NIPS (2016)
6. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
8. Grabner, A., Roth, P.M., Lepetit, V.: 3D pose estimation and 3D model retrieval for objects in the wild. In: CVPR (2018)
9. Han, K., et al.: SCNet: learning semantic correspondence. In: ICCV (2017)
10. He, X., Zhou, Y., Zhou, Z., Bai, S., Bai, X.: Triplet-center loss for multi-view 3D object retrieval. In: CVPR (2018)
11. Huang, H., Kalogerakis, E., Chaudhuri, S., Ceylan, D., Kim, V.G., Yumer, E.: Learning local shape descriptors from part correspondences with multiview convolutional networks. ACM Trans. Graph. **37**(1), 6 (2017)
12. Kanezaki, A., Matsushita, Y., Nishida, Y.: RotationNet: joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: CVPR (2018)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
14. Kundu, J.N., Uppala, P.K., Pahuja, A., Babu, R.V.: Adadepth: unsupervised content congruent adaptation for depth estimation. In: CVPR (2018)
15. Liu, C., Yuen, J., Torralba, A.: SIFT flow: dense correspondence across scenes and its applications. In: Hassner, T., Liu, C. (eds.) Dense Image Correspondences for Computer Vision, pp. 15–49. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-23048-1_2
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)

17. Mahendran, S., Ali, H., Vidal, R.: 3D pose regression using convolutional neural networks. In: ICCV (2017)
18. Paletta, L., Pinz, A.: Active object recognition by view integration and reinforcement learning. Robot. Auton. Syst. **31**(1), 71–86 (2000)
19. Poirson, P., Ammirato, P., Fu, C.Y., Liu, W., Kosecka, J., Berg, A.C.: Fast single shot detection and pose estimation. In: 3DV (2016)
20. Qi, C.R., Su, H., Niessner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view CNNs for object classification on 3D data. In: CVPR (2016)
21. Rhodin, H., et al.: Learning monocular 3D human pose estimation from multi-view images. In: CVPR (2018)
22. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
23. Schmidt, T., Newcombe, R., Fox, D.: Self-supervised visual descriptor learning for dense correspondence. IEEE Robot. Autom. Lett. **2**, 420 (2017)
24. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3D shape recognition. In: ICCV (2015)
25. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. In: CVPR (2015)
26. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
27. Taniai, T., Sinha, S.N., Sato, Y.: Joint recovery of dense correspondence and cosegmentation in two images. In: CVPR (2016)
28. Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction. In: CVPR (2018)
29. Tulsiani, S., Malik, J.: Viewpoints and keypoints. In: CVPR (2015)
30. Wu, J., et al.: Single image 3D interpreter network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 365–382. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_22
31. Xiang, Y., et al.: ObjectNet3D: a large scale database for 3D object recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 160–176. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_10
32. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond PASCAL: a benchmark for 3D object detection in the wild. In: WACV (2014)
33. Yu, W., Sun, X., Yang, K., Rui, Y., Yao, H.: Hierarchical semantic image matching using CNN feature pyramid. Comput. Vis. Image Underst. **169**, 40–51 (2018)