Learning Local RGB-to-CAD Correspondences for Object Pose Estimation

Georgios Georgakis[1], Srikrishna Karanam[2], Ziyan Wu[2], and Jana Košecká[1]
[1]Department of Computer Science, George Mason University, Fairfax VA
[2]Siemens Corporate Technology, Princeton NJ
ggeorgak@gmu.edu,{first.last}@siemens.com,kosecka@cs.gmu.edu

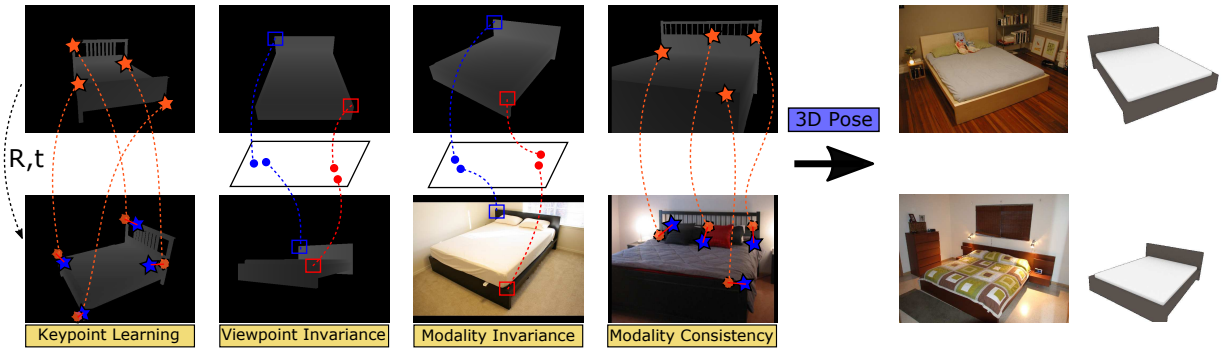


Figure 1: We present a new method that matches RGB images to depth renderings of CAD models for object pose estimation. It does not require either textured CAD models or 3D pose annotations for RGB images during training. This is achieved by enforcing viewpoint and modality invariance for local features, and learning consistent keypoint selection across modalities.

Abstract

We consider the problem of 3D object pose estimation. While much recent work has focused on the RGB domain, the reliance on accurately annotated images limits generalizability and scalability. On the other hand, the easily available object CAD models are rich sources of data, providing a large number of synthetically rendered images. In this paper, we solve this key problem of existing methods requiring expensive 3D pose annotations by proposing a new method that matches RGB images to CAD models for object pose estimation. Our key innovations compared to existing work include removing the need for either real-world textures for CAD models or explicit 3D pose annotations for RGB images. We achieve this through a series of objectives that learn how to select keypoints and enforce viewpoint and modality invariance across RGB images and CAD model renderings. Our experiments demonstrate that the proposed method can reliably estimate object pose in RGB images and generalize to object instances not seen during training.

1. Introduction

Estimating the 3D pose of objects is an important capability for enabling robots' interaction with real environments and objects as well as augmented reality applications. While several approaches to this problem assume RGB-D data [17, 31], most mobile and wearable cameras are not paired with a depth sensor, prompting recent research focus on the RGB domain. Furthermore, even though several methods have shown promising results on 3D object pose estimation with real RGB images, they either require accurate 3D annotations [23, 26, 44, 30, 16] or 3D object models with realistic textures [5, 37, 6, 17] in the training stage. Currently available datasets [20, 43] are not large enough to capture real world diversity, limiting the potential of these methods in generalizing to a variety of applications. In addition, capturing real RGB data and manual pose annotation is an arduous procedure.

The problem of object pose estimation is an inherently 3D problem; it is the shape of the object which gives away its pose regardless of its appearance. Instead of attempting to learn an intrinsic decomposition of images [14], we focus on finding the association of parts of objects depicted in RGB images with their counterparts in 3D depth images. Ideally, we would like to learn this association in order to establish correspondences between a query RGB image and a rendered depth image from a CAD model, without requiring any existing 3D annotations. This, however, requires us to address the problem of the large appearance gap between

these two modalities.

In this paper, we propose a new framework for estimating the 3D pose of objects in RGB images, using only 3D textureless CAD models of objects instances. The easily available CAD models can generate a large number of synthetically rendered depth images from multiple viewpoints. In order to address the aforementioned problems, we define a *quadruplet* convolutional neural network to jointly learn keypoints and their associated descriptors for robust matching between different modalities and changes in viewpoint. The general idea is to learn the keypoint locations using a pair of rendered depth images from a CAD model from two different poses, followed by learning how to match keypoints across modalities using an aligned RGB-D image pair. Figure 1 outlines our training constraints. At test time, given a query RGB image, we extract keypoints and their representations and match them with a database of keypoints and their associated descriptors extracted from rendered depth images. These are used to establish 2D-3D correspondences, followed by a RANSAC and PnP algorithm for pose estimation.

To summarize, our key contributions include: **1)** A new framework for 3D object pose estimation using only textureless CAD models and aligned RGB-D frames in the training stage, without explicitly requiring 3D pose annotations for the RGB images. **2)** An end-to-end learning approach for keypoint selection optimized for the relative pose estimation objective, and transfer of keypoint predictions and their representations from rendered depth to RGB images. **3)** Demonstration of the generalization capability of our method to new (unseen during training) instances of the same object category.

2. Related Work

There is a large body of work on 3D object pose estimation. Here, we review existing methods based on the type and the amount of used training data and its modalities.

Using 3D textured instance models. Notable effort was devoted to the problem of pose estimation for object instances from images, where 3D textured instance models were available during the training stage [9, 5, 37]. Early isolated approaches led to the development of more recent benchmarks for this problem [11]. Traditional approaches of this type included template matching [9, 48], where the target pose is retrieved from the best matched model in a database, and local descriptor matching [5, 37], where hand-engineered descriptors such as SIFT [22] are used to establish 2D-3D correspondences with a 3D object model followed by the PnP algorithm for 6-DoF pose. Additionally, some works employed a patch-based dense voting scheme [4, 38, 6, 17], where a function is learned to map local representations to 3D coordinates or to pose space. However, these approaches assume that the 3D object models were created from real images and contain realistic textures. In contrast, our work uses only textureless CAD models of object instances.

2D-to-3D alignment with CAD models. Other work has sought to solve 3D object pose estimation as a 2D-to-3D alignment problem by utilizing object CAD models [1, 24, 20, 13, 2, 31]. For example, Aubry *et al*. [1] learned part-based exemplar classifiers from textured CAD models and applied them on real images to establish 2D-3D correspondences. In a similar fashion, Lim *et al*. [20] trained a patch detector from edge maps for each interest point. The work of Massa *et al*. [24] learned how to match view-dependent exemplar features by adapting the representations extracted from real images to their CAD model counterparts. The closest work to ours in this area is Rad *et al*. [31], which attempts to bridge the domain gap between real and synthetic depth images, by learning to map color features to real depth features and subsequently to synthetic depth features. In their attempt to bridge the gap between the two modalities, these approaches were required to either learn a huge number of exemplar classifiers, or learn how to adapt features for each specific category and viewpoint. We avoid this problem by simply adapting keypoint predictions and descriptors between the two modalities.

Pose estimation paired with object detection. With the recent success of deep convolutional neural networks (CNN) on object recognition and detection, many works extended 3D object instance pose estimation to object categories, from an input RGB image [23, 25, 26, 44, 30, 16, 18, 41, 15]. In Mahendran *et al*. [23] a 3D pose regressor was learned for each object category. In Mousavian *et al*. [26], a discrete-continuous formulation for the pose prediction was introduced, which first classified the orientation to a discrete set of bins and then regressed the exact angle within the bin. Poirson *et al*. [30] and Kehl *et al*. [16] both extended the SSD [21] object detector to predict azimuth and elevation or the 6-DoF pose respectively. In Kundu *et al*. [18], an analysis-by-synthesis approach was introduced, in which, given predicted pose and shape, the object was rendered and compared to 2D instance segmentation annotations. All of these approaches require 3D pose annotations for the RGB images during training, as opposed to our work, which only needs the CAD models of the objects.

Keypoint-based methods. Another popular direction in the pose estimation literature is learning how to estimate keypoints, which can be used to infer the pose. These methods are usually motivated by the presence of occlusions [27, 12] and require keypoint annotations. For example, Wu *et al*. [42] trained a model for 2D keypoint prediction on real images and estimated the 3D wireframes of objects using a model trained on synthetic shapes. The 3D wireframe is then projected to real images labeled with 2D keypoints to enforce consistency. In Li *et al*. [19], the authors manually

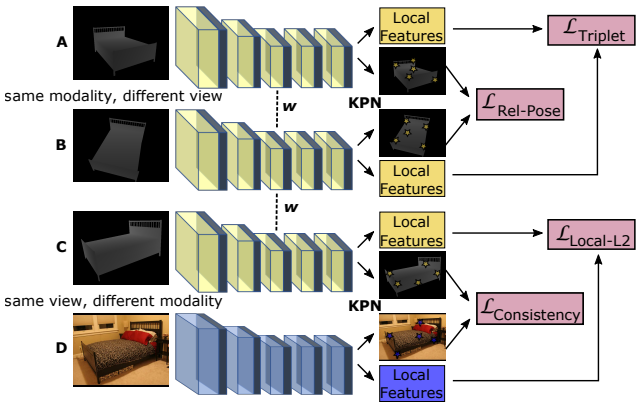


Figure 2: Outline of the proposed architecture depicting the four branches, their inputs, and the training objectives. The color coding of the CNNs signifies weight sharing.

annotated 3D keypoints on textured CAD models and generated a synthetic dataset which provides multiple layers of supervision during training, while Tekin *et al*. [39] learned to predict the 2D image locations of the projected vertices of an object's 3D bounding box before using the PnP algorithm for pose estimation. Furthermore, Tulsiani *et al*. [40] exploited the relationship between viewpoint and visible keypoints and refined an existing coarse pose estimation using keypoint predictions. Our work, rather than relying on existing keypoint annotations, optimizes the keypoint selection based on a relative pose estimation objective. Related approaches also learn keypoints [36, 7, 45, 47], but either rely on hand-crafted detectors to collect training data [45], or do not extend to real RGB pose estimation [36, 7, 47].

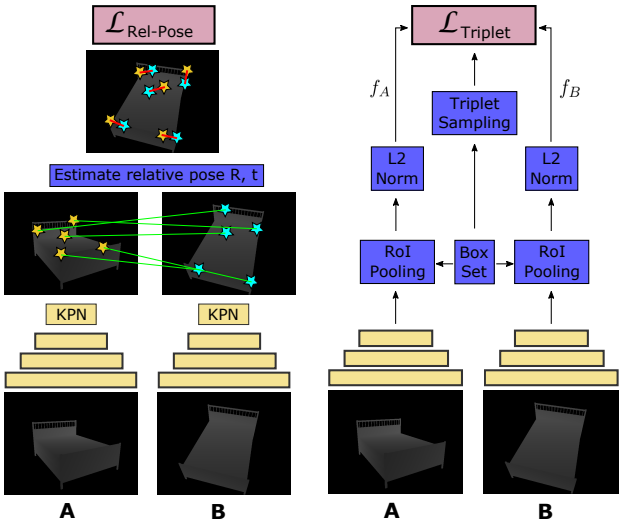
Synthetic data generation. In an attempt to address the scarcity of annotated data, some approaches rely on the generation of large amounts of synthetic data for training [35, 34, 8]. A common technique is to render textured CAD models and superimpose them on real backgrounds. In order to ensure diversity in the training data, rendering parameters such as pose, shape deformations, and illumination are randomly chosen. However, training exclusively on synthetic data has shown to be detrimental to the learned representations as the underlying statistics of real RGB images are usually very different.

3. Approach

We are interested in estimating the 3D pose of objects in RGB images by matching keypoints to the object's CAD model. Our work does not make use of pose annotations, but instead relies on CAD model renderings of different poses that are easily obtained with an off-the-shelf renderer, such as Blender [3]. These rendered depth images are used to learn keypoints and their representations optimized for the task of pose estimation. The learned representations are then transferred to the RGB domain. In summary, our work can be divided into four objectives: keypoint learning, view-invariant descriptors, modality-invariant descriptors, and modality consistent keypoints.

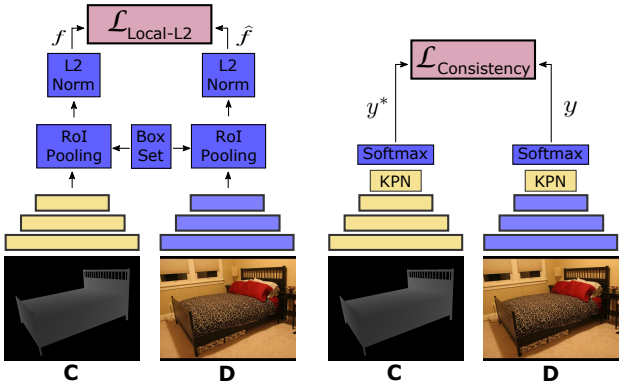
Specifically, each training input is provided as a quadruplet of images, consisting of a pair of rendered depth images sampled from the object's view sphere and a pair of aligned depth and RGB images (see Figure 2). For each image, we predict a set of keypoints and their local representations, but the optimization objectives differ for the various branches. For the first two branches A and B, L_{rel_pose} loss enforces the pose consistency of the keypoints selection and the similarity of keypoint descriptors for their matching is enforced using a triplet loss $L_{triplet}$. The two bottom branches C and D are utilized to enforce consistent keypoint prediction between the depth and the RGB modalities $L_{consistency}$ and for matching their local representations across the modalities L_{local_l2}. The general idea of our approach is to learn informative keypoints and their associated local descriptors from abundant rendered depth images and transfer this knowledge to the RGB data.

Architecture. Our proposed architecture is a Quadruplet convolutional neural network (CNN), where each branch has a backbone CNN (e.g., VGG) to learn feature representations and a keypoint proposal network (KPN) comprised of two convolutional layers. The output feature maps from the backbone's last convolutional layer are fed as input to the KPN. KPN produces a score map of dimensions $\frac{H}{s} \times \frac{W}{s} \times D$, where H and W are the input image's height and width respectively, s is the network stride, and $D = 2$ is a score whether the particular location is a keypoint or not. Softmax is then applied on D such that each location on the KPN output map has a 2-D probability distribution. This output map can be seen as a keypoint confidence score for a grid-based set of keypoint locations over the 2D image. The density of the keypoint sampling depends on the network stride s, which in our case was 16 (i.e. a keypoint proposal every 16 pixels). In order to extract a descriptor (dim-2048) for each keypoint, the backbone's feature maps are passed to the region-of-interest (RoI) pooling layer along with a set of bounding boxes each centered at a keypoint location. The first pair of branches (A, B) of the network are trained with a triplet loss applied to local features, while a relative pose loss is applied to the keypoint predictions. Branch D is trained using a Euclidean loss on the local features and with a consistency loss that attempts to align its keypoint predictions and local representations to those of branch C. Note that branches A, B, and C share their weights, while branch D is a different network. Since branch D receives as input a different modality than the rest and we desire branches C and D to produce the same outputs, their weights during training must be independent. In the following sections, we describe the details of the loss functions and training.



(a) Relative pose loss. (b) Triplet loss.

Figure 3: Relative pose and triplet losses.



(a) Local Euclidean loss. (b) Keypoint consistency loss.

Figure 4: Local Euclidean and keypoint consistency losses.

3.1. Keypoint Learning by Relative Pose Estimation

The overall idea behind learning keypoint predictions is to select keypoints that can be used for relative pose estimation between the input depth images in branches A and B. Specifically, given the two sets of keypoints, we establish correspondences in 3D space, estimate the rotation R and translation t, and project the keypoints from depth image A to depth image B. Any misalignment (re-projection error) between the projected keypoints is used to penalize the initial keypoint selections. A pictorial representation of the relative pose objective is shown in Figure 3a.

The relative pose objective is formulated as a least squares problem, which finds the rotation R and translation t for which the error of the weighted correspondences is minimal. Formally, for two sets of corresponding points: $P = \{p_1, p_2, \ldots, p_n\}, Q = \{q_1, q_2, \ldots, q_n\}$ we wish to estimate R and t such that:

$$(R,t) =_{R\in SO(3),t\in\mathbb{R}^3} \sum_{i=1}^{n} w_i ||(Rp_i + t) - q_i||^2 \qquad (1)$$

where $w_i = s_i^A + s_i^B$ is the weight of correspondence i and s_i^A and s_i^B are the predicted keypoint probabilities, as given by KPN followed by a Softmax layer, that belong to correspondence i from branches A and B respectively. Given a set of correspondences and their weights, an SVD-based closed-form solution for estimating R and t that depends on w can be found in [33]. The idea behind this formulation is that correspondences with high re-projection error should have low weights, therefore a low predicted keypoint score, while correspondences with low re-projection error should have high weights, therefore high predicted keypoint score. With this intuition, we formulate the relative pose loss as:

$$L_{rel\_pose} = \frac{1}{n} \sum_{i=1}^{n} w_i g(w_i) \qquad (2)$$

where $g(w_i) = ||(Rp_i + t) - q_i||^2$. Since our objective is to optimize the loss function with respect to estimated keypoint scores, we penalize each keypoint score separately by estimating the gradients for each correspondence and back-propagating them accordingly.

3.2. Learning Keypoint Descriptors

In order to match keypoint descriptors across viewpoints, we apply a triplet loss on local features extracted from branches A and B. This involves using the known camera poses of the rendered pairs of depth images and sampling of training keypoint triplets (anchor-positive-negative). Specifically, for a randomly selected keypoint as an anchor from the first image, we find the closest keypoint in 3D from the paired image and use it as a positive, and also select a further away point in 3D to serve as the negative. The triplet loss then optimizes the representation such that the feature distance between the anchor and the positive points is smaller than the feature distance between the anchor and the negative points plus a certain margin, and is defined as follows:

$$L_{triplet} = \frac{1}{N} \sum_{i}^{N} \max(0, ||f_i^a - f_i^p||^2 - ||f_i^a - f_i^n||^2 + m) \qquad (3)$$

where f_i^a, f_i^p, and f_i^n are the local features for the anchor, positive, and negative correspondingly of the i^{th} triplet example and m is the margin. Traditionally, the margin hyperparameter is manually defined as a constant throughout the training procedure; however, we take advantage of the 3D information and define the margin to be equal to $D_n - D_p$, where D_n is the 3D distance between the anchor and negative, and D_p is the 3D distance between the anchor and

positive. Ideally, D_p should be 0, but practically due to the sampling of the keypoints in the image space it is usually a small number close to 0. Essentially this ensures that the learned feature distances are proportional to the 3D distances between the examples and assumes that the features and 3D coordinates are normalized to unit vectors. Note that the triplet loss only affects the backbone CNN during training and not the KPN. A pictorial representation of the triplet objective is shown in Figure 3b.

3.3. Cross-modality Representation Learning

Finally, we can transfer the learned features and keypoint proposals from branches (A, B) to branch D, using branch C as a bridge, similar to knowledge distillation techniques [10]. To accomplish this, network parameters in branches A, B, and C are shared, and the outputs of branches C and D are compared and penalized according to any misalignment. The core idea is to enforce both the backbone and KPN in branches C and D to generate as similar outputs as possible. This objective can be accomplished by means of two key components that are described next.

Local Feature Alignment. In order to align local feature representations in branches C and D (see Figure 4a), we consider the predicted keypoints in branch C and compute each keypoint's feature representation, $f_i, i = 1, \ldots, k$. Keypoint features at corresponding spatial locations from branch D are represented as $\hat{f}_i, i = 1, \ldots, k,$. Formally, we optimize the following objective function:

$$L_{local\_l2} = \frac{1}{k} \sum_{i=1}^{k} \|\hat{f}_i - f_i\| \tag{4}$$

Since we want to align $\hat{f}_i$ with f_i, during backpropagation, we fix f_i as ground-truth and backpropagate gradients of L_{local_l2} only to the appropriate locations in branch D.

Keypoint Consistency. Enforcement of the keypoint consistency constraint requires the KPN from branch D to produce the same keypoint predictions as the KPN from branch C. It can be achieved using a cross-entropy loss, which is equivalent to a log loss with binary labels: $L = -\frac{1}{n}\sum_{i=1}^{n} y_i^* \log y_i$, where y_i^* is the ground-truth label and y_i is the prediction. This in our case becomes:

$$L_{consistency} = -\frac{1}{n} \sum_{i=1}^{n} y_i^C \log y_i^D \tag{5}$$

where y_i^C are the keypoint predictions from branch C, which serve as the ground-truth, and y_i^D are the keypoint predictions from branch D. This loss penalizes any misalignment between the keypoint predictions of the two branches and forces branch D to imitate the outputs of branch C. Figure 4b illustrates inputs to $L_{consistency}$.

Category	Chair		Sofa	
Metric	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$
Render for CNN [34]	4.3	2.1	11.6	1.2
Vps & Kps [40]	10.3	1.7	23.3	1.2
Deep3DBox [26]	10.8	1.9	25.6	**1.0**
Proposed	**13.4**	**1.6**	**30.2**	1.1

Table 1: Comparison with supervised approaches when trained on Pix3D and tested on Pascal3D+ on $Acc_{\frac{\pi}{6}}$ (%) and $MedErr$ (radians).

Overall objective. Our overall training objective is the combination of the losses described above:

$$\begin{aligned} L_{all} = \lambda_1 L_{triplet} + \lambda_2 L_{rel\_pose} \\ + \lambda_3 L_{local\_l2} + \lambda_4 L_{consistency} \end{aligned} \tag{6}$$

where each λ is the weight for the corresponding loss.

4. Experiments

In order to validate our approach, we perform experiments on the Pascal3D+ [43] dataset and the newly introduced Pix3D [35] dataset.We conduct four key experiments. First, we compare to supervised state-of-the-art methods by training on Pix3D and testing on Pascal3D+ (sec. 4.1); second, we perform an ablation study on Pix3D and evaluate the performance of different parts of our approach (sec. 4.2); third, we test how our model generalizes to new object instances by training only on a subset of provided instances and testing on unseen ones (sec. 4.3); and finally, data from an external dataset, such as NYUv2 [32] is used to train and test on Pix3D (sec. 4.4). The motivation for the fourth experiment is to demonstrate that our framework can utilize RGB-D pairs from another realistic dataset, where the alignment between the RGB and the depth is provided by the sensor. We use the geodesic distance for evaluation: $\Delta(R_1, R_2) = \frac{||\log(R_1^T R_2)||_F}{\sqrt{2}}$, reporting percentage of predictions within $\frac{\pi}{6}$ of the ground-truth $Acc_{\frac{\pi}{6}}$ and $MedErr$. Additionally, we show the individual accuracy of the three Euler angles, where the distance is the smallest difference between two angles: $\Delta(\theta_1, \theta_2) = \min(2\pi - ||\theta_1 - \theta_2||, ||\theta_1 - \theta_2||)$. For the last metric we also use a threshold of $\frac{\pi}{6}$.

Implementation details. We use VGGNet as each branch's backbone and start from ImageNet pretrained weights, while KPN is trained from scratch. We set the learning rate to 0.001 and all λ weights to 1. In order to regularize the relative pose loss such that it predicts keypoints inside objects, we add a mask term, realized as a multinomial logistic loss. The ground-truth is a binary mask of the object in the rendered depth. This loss is only applied on branches A and B with a smaller weight of 0.25. Finally, the bounding box dimensions for the RoI layer are set to 32×32.

Training data. All our experiments require a set of quadruplet inputs. For the first two inputs, we first sample from

Figure 5: Keypoint prediction examples on test images from the Pix3D dataset. Top, middle, and bottom rows show results from experiments of sections 4.2, 4.3, and 4.4 respectively. Note that we applied non-maximum suppression (NMS) on the keypoint predictions in order to select the highest scoring keypoint from each region.

Category	Bed					Chair					Desk				
Metric	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$
Baseline-A	51.4	39.1	35.2	7.3	1.7	30.2	43.2	20.0	3.3	2.0	28.9	30.9	20.4	2.6	2.2
Baseline-ZDDA	48.6	50.3	41.9	21.8	1.5	35.3	48.3	26.6	11.5	1.7	24.3	23.7	21.1	3.9	2.0
Proposed - joint	69.8	51.9	58.1	31.3	1.0	**55.3**	**62.7**	44.7	31.1	**0.9**	57.2	48.7	51.0	25.0	1.1
Proposed - alternate	**83.2**	**67.0**	**70.4**	**50.8**	**0.5**	54.7	60.1	**47.0**	**31.2**	1.0	**65.1**	**55.3**	**58.6**	**34.9**	**0.9**

Table 2: Azimuth (%), elevation (%), in-plane rotation (%) accuracy, $Acc_{\frac{\pi}{6}}$ (%) & $MedErr$ (radians) for sec 4.2 experiment.

each object's viewsphere and render a view every 15 degrees in azimuth and elevation for three different distances. Then, we sample rendered pairs such that their pose difference is between $\frac{\pi}{12}$ and $\frac{\pi}{3}$. For the last two inputs, we require aligned depth and RGB image pair. In order to demonstrate our approach on the Pix3D dataset, we generate these alignments using the dataset's annotations, however, we do not use annotations during training in any other capacity. As we show in sec. 4.4, alternatively the aligned depth and RGB images can be sampled from an existing RGB-D dataset or through hand-alignment [2]. Note that for each quadruplet, the selection of the first pair of inputs is agnostic to the pose of the object in the last two inputs. We further note that, given sufficient viewsphere sampling, what is important is how the quadruplet training data is generated (particularly pairs for branches A & B). If the pairs have a small pose difference (e.g., $\leq \frac{\pi}{12}$), the model does not adequately learn view-invariant representations. On the other hand, with larger pose differences (e.g., $\frac{\pi}{2}$), overlapping areas between the two views are small, so finding correspondences across views is harder. We found sampling pairs with a maximum pose difference of $\frac{\pi}{3}$ provides a good balance. A possible future extension can be to incorporate "interesting" viewpoints [46], which are typically task-dependent, into our pipeline for further improvements (e.g., reduced data requirements or training time).

Testing protocol. For every CAD model instance used in our experiments, we first create a repository of descriptors each assigned to a 3D coordinate. To do so, 20 rendered views are sampled from the viewing sphere of each object, similarly to how the training data are generated, and keypoints are extracted from each view. Note that for this procedure, we use the trained network that corresponds to branch A of our architecture. Then we pass a query RGB image through the network of branch D, generate keypoints and their descriptors and match them to the repository of the corresponding object instance. Finally, the established correspondences are passed to RANSAC and PnP algorithm to estimate the pose of the object. For every keypoint generation step we use the keypoints with the top 100 scores during database creation and top 200 scores for the testing RGB images. When testing on Pix3D, we have defined a test set which contains untruncated and unoccluded examples of all category instances, with 179, 1451, and 152 images in total for *bed*, *chair*, and *desk* category respectively. For Pascal3D+ we follow the provided test sets and make use of the ground-truth bounding boxes.

4.1. Comparison with supervised approaches

Given our approach does not use any pose annotations during training, it is challenging to evaluate it against existing state-of-the-art methods, which use pose annotations

Category	Bed					Chair					Desk				
Metric	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$
Baseline-A	38.2	39.6	30.6	9.7	1.9	28.6	41.4	20.3	3.7	1.9	37.6	34.4	28.8	5.6	2.0
Baseline-ZDDA	29.9	39.6	22.2	4.9	2.3	30.1	44.6	21.5	7.6	1.9	36.8	43.2	30.4	13.6	1.7
Proposed - joint	66.7	50.0	62.5	29.2	0.9	43.7	50.4	31.3	15.1	1.4	59.2	44.0	41.6	13.6	1.3
Proposed - alternate	**75.7**	**61.1**	**74.3**	**45.1**	**0.6**	**52.0**	**57.4**	**38.0**	**21.2**	**1.2**	**62.4**	**44.0**	**53.6**	**18.4**	**1.2**

Table 3: Results for azimuth (%), elevation (%), in-plane rotation (%) accuracy, $Acc_{\frac{\pi}{6}}$ (%) and $MedErr$ (radians) for the sec. 4.3 experiment.

Figure 6: Illustration of rendered estimated poses on test RGB images from the Pix3D dataset for the sec. 4.2 experiment.

during training. In addition, our method cannot be trained on Pascal3D+ because it requires paired RGB and depth images, which cannot be generated from the dataset's annotations. Therefore, we designed the following experiment for a fair comparison: we train all methods on Pix3D and test on Pascal3D+. We compare to the state-of-the-art methods of Deep3DBox [26], Render for CNN [34], and Viewpoints & Keypoints [40], all of which require pose annotations for RGB images. Other approaches, such as Pavlakos *et al*. [27], were considered for comparison but unfortunately they require semantic keypoint annotations during training which Pix3D does not provide. We conduct this evaluation on the common categories between Pix3D and Pascal3D+ (*chair* and *sofa*) and report results in Table 1.

As expected, all approaches generally underperform when applied on a new dataset. Our method demonstrates better generalization and achieves higher $Acc_{\frac{\pi}{6}}$ for both objects, even though it does not explicitly require 3D pose annotations during training. This is due to fundamental conceptual differences between these approaches and ours. These methods formulate viewpoint estimation as a classification problem where a large number of parameters in fully-connected layers are to be learned. This increases the demand for data and annotations and confines the methods mostly to data distributions that were trained on. On the other hand, we exploit CAD models to densely sample from the object's viewsphere, and explicitly bridge the gap between the synthetic data and real images, thereby reducing the demand for annotations. Furthermore, the learned local correspondences allow more flexibility in understanding the geometry of unseen objects, as we also show in sec. 4.3.

4.2. Ablation study

To understand each objective's contribution, we have carefully designed a set of baselines, which we train and test on Pix3D, and compare them on the task of pose estimation for the *bed*, *chair*, and *desk* categories.

Baseline-A. In order to assess the importance of the cross-modality representation learning (sec. 3.3), we learn view-invariant depth representations and depth keypoints and simply use these keypoints and representations during testing. In practice, this corresponds to removing the local euclidean and keypoint consistency losses, and using only the triplet and relative pose losses during training. Consequently this baseline is utilizing only depth data during training, but is applied on RGB images during testing.

Baseline-ZDDA. Another baseline would be to only learn RGB-D modality invariant representations, i.e., similar features for RGB and depth images, which can then be used to match RGB images to depth renderings from CAD models. In practice, this would correspond to training our proposed approach with only the local feature alignment objective by sampling all possible keypoint locations. This is similar in spirit to and an improved version of ZDDA [28], a domain adaptation approach that maps RGB and depth modalities to the same point in the latent space.

Joint and alternate training. Finally we use all objectives in our approach and investigate two different training strategies. First we try training all objectives jointly in a single optimization session and report this baseline as *Proposed-joint*. Second, we define a three-step alternating training, where we initially optimize using only the triplet and relative pose losses (i.e. branches A, B, C), then we opti-

mize only with the local euclidean and keypoint consistency losses (i.e. branch D), and in the last step all objectives are jointly optimized together. This baseline is reported as *Proposed-alternate*. Note that also experiments in sec. 4.1 and 4.4 follow this training paradigm.

Results. We first show, in Figure 5 (top row), qualitative keypoint prediction results on test images, where we see keypoint predictions that generally satisfy our intuition of good keypoints. We then adopt the testing protocol described above to report quantitative pose estimation results for test RGB images. Performance analysis is shown in Table 2 for the three object categories. As can be noted from the results, our proposed model generally achieves higher accuracy when compared to the baseline approaches. In particular, the improvements over Baseline-A suggests that keypoint and representation modality adaptation enforced in our model is critical. Furthermore, the improvements over Baseline-ZDDA suggests that simply performing modality adaptation for the RGB and depth features is not sufficient, and learning keypoints and view-invariant representations, as is done in our method, is important to achieve good performance. Finally, we observe that alternating training outperforms the joint strategy, demonstrating the importance of learning good keypoints and representations first, before transferring to the RGB modality.

4.3. Model transferability

In this section, we demonstrate the transfer capability, where the goal is for a model, trained according to the proposed approach, to generalize well to category instances **not seen** during training. This is key to practical usability of the approach since we cannot possibly have relevant CAD models of all instances of interest during training. To this end, the baselines introduced in sec. 4.2 are re-used with the following experimental protocol: during training, quadruplets are sampled from a subset of the available instances for each category, and test on RGB images corresponding to all other instances. For instance, for the *bed* category, we use 10 instances for training and 9 instances for testing. Similarly, for *chair* and *desk*, we use 111 and 12 instances respectively for training and the rest for testing. During testing, we use the same protocol as above. We present qualitative keypoint predictions in Figure 5 (middle row) and report quantitative performance in Table 3. We see our model shows good transferability, providing (a) a similar level of detail in the predicted keypoints as before, (b) improved accuracy when compared to the baselines, and (c) absolute accuracies that are not too far from those in Table 2.

4.4. Framework flexibility

While the results above use RGB-D pairs from Pix3D for model training, in principle, our approach can be used in conjunction with other datasets that provide aligned RGB-D pairs as well. Such capability will naturally make it easier to train models with our framework, leading to improved framework flexibility. To demonstrate this aspect, we train our model as before, but now for input to branches C and D, we use aligned RGB-D pairs from the NYUv2 [32] dataset. Since these pairs contain noisy depth images from a real depth sensor, we synthetically apply realistic noise on the clean rendered depth images, used for branches A and B, using DepthSynth [29]. This ensures branches A, B, and C still receive the same modality as input. Note that we do not test on NYUv2, but rather we use it to collect auxiliary training data and perform testing on Pix3D. Similarly to all other experiments, we do not use any pose annotations for the RGB images as part of training our model and we follow the previous testing protocol. Figure 5 (bottom row), shows some keypoint prediction results on test data from Pix3D. In Table 4, we report quantitative results. We can make several observations- while the numbers are lower than those with the proposed method in Table 2, which is expected, they are higher than all the baselines reported in Table 2. Please note that the baselines were trained with alignment from Pix3D, whereas our model here was trained with alignment from NYUv2. These results, along with those in the previous section, show the potential of our approach in learning generalizable models for estimating object pose, while not explicitly requiring pose annotations during training.

Metric	Az.	El.	Pl.	$Acc_{\frac{\pi}{6}} \uparrow$	$MedErr \downarrow$
Bed	65.9	54.1	44.0	24.0	1.0
Chair	44.3	51.0	31.0	15.2	1.6
Desk	50.0	45.4	31.6	7.2	1.9

Table 4: Results for sec. 4.4 experiment. All numbers are % except $MedErr$ (radians).

5. Conclusions

We proposed a new framework for 3D object pose estimation in RGB images, which does not require either textured CAD models or 3D pose annotations for RGB images during training. We achieve this by means of a novel end-to-end learning pipeline that guides our model to discover keypoints in rendered depth images optimized for relative pose estimation as well as transfer the keypoints and representations to the RGB modality. Our experiments have demonstrated the effectiveness of the proposed method on unseen testing data compared to supervised approaches, suggesting that it is possible to learn generalizable models without depending on pose annotations.

Acknowledgments. This paper is based primarily on the work done during the first author's internship at Siemens Corporate Technology. This research is supported in part by the NSF NRI grant 1527208.

References

[1] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014.

[2] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5965–5974, 2016.

[3] Blender. https://www.blender.org/.

[4] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[5] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011.

[6] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3583–3592, 2016.

[7] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, Jan Ernst, and Jana Košecká. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1965–1973, 2018.

[8] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Inferring 3d object pose in rgb-d images. *arXiv preprint arXiv:1502.04652*, 2015.

[9] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 858–865. IEEE, 2011.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[11] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Sahin Caner, Manhardt Fabian, Tombari Federico, Kim Tae-Kyun, Matas Jiri, and Rother Carsten. Bop: Benchmark for 6d object pose estimation. In *ECCV*, 2018.

[12] Moos Hueting, Pradyumna Reddy, Vladimir Kim, Nathan Carr, Ersin Yumer, and Niloy Mitra. Seethrough: finding chairs in heavily occluded indoor scene images. *CoRR abs/1710.10473*, 2017.

[13] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2422–2431. IEEE, 2017.

[14] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised intrinsic image decomposition. In *Advances in Neural Information Processing Systems*, pages 5936–5946, 2017.

[15] Yueying Kao, Weiming Li, Zairan Wang, Dongqing Zou, Ran He, Qiang Wang, Minsu Ahn, Sunghoon Hong, et al. An appearance-and-structure fusion network for object viewpoint estimation. In *IJCAI*, pages 4929–4935, 2018.

[16] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the International Conference on Computer Vision (ICCV 2017), Venice, Italy*, pages 22–29, 2017.

[17] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *European Conference on Computer Vision*, pages 205–220. Springer, 2016.

[18] Abhijit Kundu, Yin Li, and James 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*, pages 3559–3568, 2018.

[19] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[20] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2992–2999, 2013.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[22] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[23] Siddharth Mahendran, Haider Ali, and René Vidal. 3d pose regression using convolutional neural networks. In *IEEE International Conference on Computer Vision*, volume 1, page 4, 2017.

[24] Francisco Massa, Bryan C Russell, and Mathieu Aubry. Deep exemplar 2d-3d detection by adapting from real to rendered views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6024–6033, 2016.

[25] Roozbeh Mottaghi, Yu Xiang, and Silvio Savarese. A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 418–426, 2015.

[26] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Košecká. 3d bounding box estimation using deep learning and geometry. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5632–5640. IEEE, 2017.

[27] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *Robotics and Automation*

(ICRA), 2017 IEEE International Conference on, pages 2011–2018. IEEE, 2017.

[28] Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *European Conference on Computer Vision*, pages 793–810. Springer, 2018.

[29] Benjamin Planche, Ziyan Wu, Kai Ma, Shanhui Sun, Stefan Kluckner, Oliver Lehmann, Terrence Chen, Andreas Hutter, Sergey Zakharov, Harald Kosch, et al. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition. In *3D Vision (3DV), 2017 International Conference on*, pages 1–10. IEEE, 2017.

[30] Patrick Poirson, Phil Ammirato, Cheng-Yang Fu, Wei Liu, Jana Kosecka, and Alexander C Berg. Fast single shot detection and pose estimation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 676–684. IEEE, 2016.

[31] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Domain transfer for 3d pose estimation from color images without manual annotations. *arXiv preprint arXiv:1810.03707*, 2018.

[32] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[33] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *no*, 3:1–5, 2017.

[34] Hao Su, Charles R Qi, Yangyan Li, and Leonidas 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*, pages 2686–2694, 2015.

[35] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018.

[36] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *arXiv preprint arXiv:1807.03146*, 2018.

[37] Jie Tang, Stephen Miller, Arjun Singh, and Pieter Abbeel. A textured object recognition pipeline for color and depth image data. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3467–3474. IEEE, 2012.

[38] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In *European Conference on Computer Vision*, pages 462–477. Springer, 2014.

[39] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.

[40] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[41] Zairan Wang, Weiming Li, Yueying Kao, Dongqing Zou, Qiang Wang, Minsu Ahn, and Sunghoon Hong. Hcr-net: A hybrid of classification and regression network for object pose estimation. In *IJCAI*, pages 1014–1020, 2018.

[42] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. 3d interpreter networks for viewer-centered wireframe modeling. *International Journal of Computer Vision*, pages 1–18, 2018.

[43] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 75–82. IEEE, 2014.

[44] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[45] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[46] Shanghong Zhao and Wei Tsang Ooi. Modeling 3d synthetic view dissimilarity. *The Visual Computer*, 32(4):429–443, 2016.

[47] Xingyi Zhou, Arjun Karpur, Chuang Gan, Linjie Luo, and Qixing Huang. Unsupervised domain adaptation for 3d keypoint estimation via view consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 137–153, 2018.

[48] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3936–3943. IEEE, 2014.