StarMap for Category-Agnostic Keypoint and Viewpoint Estimation

Xingyi Zhou[1]([✉]) , Arjun Karpur[1] , Linjie Luo[2] , and Qixing Huang[1]

[1] The University of Texas at Austin, Austin, USA
{zhouxy,akarpur,huangqx}@cs.utexas.edu
[2] Snap Inc., Venice, USA
linjie.luo@snap.com

Abstract. Semantic keypoints provide concise abstractions for a variety of visual understanding tasks. Existing methods define semantic keypoints separately for each category with a fixed number of semantic labels in fixed indices. As a result, this keypoint representation is infeasible when objects have a varying number of parts, e.g. chairs with varying number of legs. We propose a category-agnostic keypoint representation, which combines a multi-peak heatmap (StarMap) for all the keypoints and their corresponding features as 3D locations in the canonical viewpoint (CanViewFeature) defined for each instance. Our intuition is that the 3D locations of the keypoints in canonical object views contain rich semantic and compositional information. Using our flexible representation, we demonstrate competitive performance in keypoint detection and localization compared to category-specific state-of-the-art methods. Moreover, we show that when augmented with an additional depth channel (DepthMap) to lift the 2D keypoints to 3D, our representation can achieve state-of-the-art results in viewpoint estimation. Finally, we show that our category-agnostic keypoint representation can be generalized to novel categories.

Keywords: 3D vision · Category-agnostic · Keypoint estimation Viewpoint estimation · Pose estimation

1 Introduction

Semantic keypoints, such as joints on a human body or corners on a chair, provide concise abstractions of visual objects regarding their compositions, shapes, and poses. Accurate semantic keypoint detection forms the basis for many visual understanding tasks, including human pose estimation [4,22,25,51], hand pose estimation [46,52], viewpoint estimation [24,35], feature matching [15], fine-grained image classification [47], and 3D reconstruction [9,10,36,39].

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01246-5_20) contains supplementary material, which is available to authorized users.
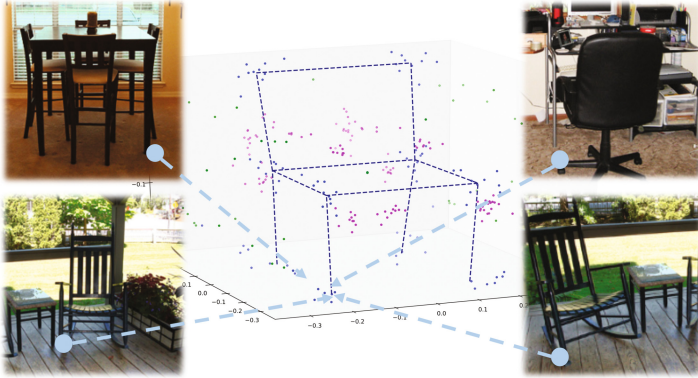
Fig. 1. Illustration of Canonical View Semantic Feature. It is shared across all object categories. We show 2 categories: chair (in blue) and table (in green). For the left frontal leg of chair on bottom left, it has (i) the same CanViewFeature with the same chair keypoint from a different viewpoint (bottom right), (ii) similar feature with another chair instance's corresponding keypoint (top right), and (iii) similar feature with left frontal leg from a table (top left). We *Can View* this feature in 3D space (middle). (Color figure online)

Existing methods define a fixed number of semantic keypoints for each object category in isolation [22, 24, 35, 40]. A standard approach is to allocate a heatmap channel for each keypoint. Or in other words, keypoints are inferred as separate heat maps according to their *encoding order*. This approach, however, is not suitable when objects have a varying number of parts, e.g. chairs with varying numbers of legs. The approach is even more limiting when we want to share and use keypoint labels of multiple different categories. In fact, keypoints of different categories do share rich compositional similarities. For instance, chairs and tables may share the same configuration of legs, and motorcycles and bicycles all contain wheels. Category-specific keypoint encodings fail to capture both the intra-category part variations and the inter-category part similarities.

In this paper, we propose a novel, category-agnostic keypoint representation. Our representation consists of two components: (1) a single channel, multi-peak heatmap, termed *StarMap*, for all keypoints of all objects; and (2) their respective feature (Fig. 1), termed *CanViewFeature*, which is defined as the 3D locations in a normalized canonical object view (or a world coordinte system). Specifically, StarMap combines the separate keypoint heat maps in previous approaches [24, 35] into a single heat map, and thus unifies the detection of different keypoints. CanViewFeature provides semantic discrimination between keypoints, i.e., through their locations in the normalized canonical object view. One intuition behind this representation is that the distribution of keypoints' 3D locations in the canonical object view encodes rich semantic and compositional information. For example, the locations of all legs are close to the ground,

and they are below the seats. Our representation can be obtained via *supervised* training on any standard datasets with 3D viewpoint annotations, such as Pascal3D+ [42] and ObjectNet3D [41].

Our representation provides the flexibility to represent varying numbers of keypoints across different categories by eliminting the hard-encoding of keypoints. Additionally, we demonstrate that our representation can still achieve competitive results in keypoint detection and localization compared to the state-of-the-art category-specific approaches [15,35] (Sect. 4.2) by using simple nearest neighbor association on the category-level keypoint templates.

One direct application of our representation is viewpoint estimation [19,28, 35], which can be achieved by solving a perspective-n-points (PnP) [12] problem to align the *CanViewFeature* with the *StarMap*. Further, we observed considerable performance gains in this task by augmenting the *StarMap* with an additional depth channel (*DepthMap*) to lift the 2D image coordinates into 3D. We report state-of-the-art performance compared to previous viewpoint estimation methods [19,24,28,35] with ablation studies on each component. Finally, we show our method works well when applied to unseen categories. Full code is publicly available at https://github.com/xingyizhou/StarMap.

2   Related Works

Keypoint Estimation. Keypoint estimation, especially human joint estimation [4,22,31,33,49] and rigid object keypoint estimation [40,50], is a widely studied problem in computer vision. In the simplest case, a 2D/3D keypoint can be represented by a 2/3-dimension vector and learned by supervised regression. Toshev et al. [33] first trained a deep neural network for 2D human pose regression and Li et al. [13] extended this approach to 3D. Starting from Tompson et al. [32], the heatmap representation has dominated the 2D keypoint estimation community and has achieved great success in both 2D human pose estimation [22,38,44] and *single category* man-made object keypoint detection [39,40]. Recently, the heatmap representation has been generalized in various different directions. Cao et al. [4] and Newell et al. [21] extended the single peak heatmap (for single keypoint detection) to a multi-peak heatmap where each peak is one instance of a *specific type* of keypoint, enabling bottom-up, multi-person pose estimation. Pavlakos et al. [25] lifted the 2D pixel heatmap to a 3D voxel heatmap, resulting in an end-to-end 3D human pose estimation system. Tulsiani et al. [35] and Pavlakos et al. [24] stacked keypoint heatmaps from different object categories together for multi-category object keypoint estimation. Despite good performance gained by these approaches, they share a common limitation: each heatmap is only trained for a *specific keypoint type* from a specific object. Learning each keypoint individually not only ignores the intra-category variations or inter-category similarities, but also makes the representation inherently impossible to be generalized to unknown keypoint configurations for novel categories.

Viewpoint Estimation. Viewpoint estimation, i.e., estimating an object's orientation in a given frame, is a practical problem in computer vision and

robotics [11,24]. It has been well explored by traditional techniques that solve for transformations between corresponding points in the world and image views; this is known as the Perspective-n-Point Problem [12,17]. Lately, viewpoint estimation accuracy and utility have been greatly improved in the deep learning era. Tulsiani et al. [35] introduced viewpoint estimation as a bin classification problem for each viewing angle (azimuth, elevation and in-plane rotation). Mousavian et al. [19] augmented the bin classification scheme by adding regression offsets within each bin so that predictions could be more fine-grained. Szeto et al. [29] used annotated keypoints as additional input to further improve bin classification. To combat scarcity of training data and generic features, Su et al. [28] proposed to synthesize images with known 3D viewpoint annotations and proposed a geometry-aware loss to further boost the estimation performance. Recently, Pavlakos et al. [24] proposed to use detected *semantic* keypoint followed by a PnP algorithm [12] to solve for the resulting viewpoint matrix and achieved state-of-the-art results. However, this method relies on category-specific keypoint annotation and is not generalizable. On the contrary, our approach is both accurate and category-agnostic, by utilizing category-agnostic keypoints.

General Keypoint Detection. There are several related concepts similar to our general semantic keypoint. The most well-known one is the SIFT descriptor [16], which aims to detect a large number of *interest* points based on local and *low level* image statistics. Also, the heatmap representation has been used in saliency detection [8] and visual attention [43], which detects a *region* of image which is "important" in the context. Similarly, Altwaijry et al. [1] used the heatmap representation to detect a set of points that is useful for feature matching. The key difference between our keypoint and the above concepts is that their keypoints do not contain semantic meanings and are not annotated by humans, making them less useful in high level vision tasks such as pose estimation.

To our best knowledge, we are the first to propose a category-agnostic keypoint representation and show that it is directly applicable to viewpoint estimation.

3   Approach

In this section, we describe our approach for learning a category-agnostic keypoint representation from a single RGB image. We begin with describing the representation in Sect. 3.1. We then introduce how to learn this representation in Sect. 3.2. Finally, we show a direct application of our representation in viewpoint estimation in Sect. 3.3.

3.1   Category-Agnostic Keypoint Representation

A desired general purpose keypoint representation should be both adaptive (i.e., should be able to represent different content of different visual objects) and semantically meaningful (i.e., should convey certain semantic information for downstream applications).
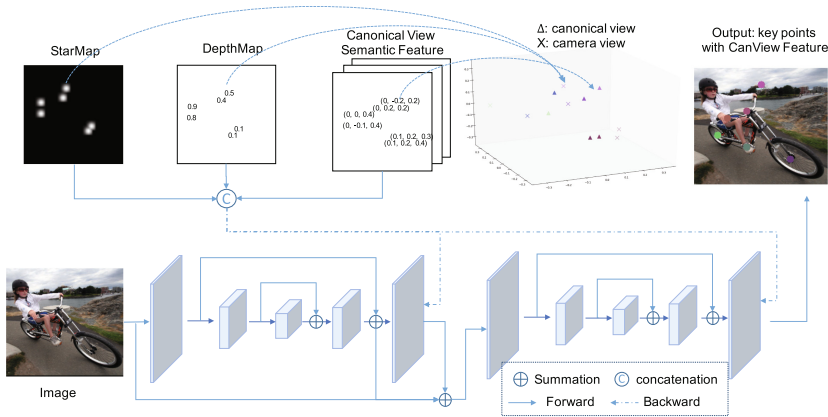
Fig. 2. Illustration of our framework. For an input image, our network predicts three component: StarMap, Canonical View Feature, and DepthMap. Varying number of keypoints are extracted at the peak location of StarMap and their Depth and Can-ViewFeature can be accessed at the corresponding channels.

So far the most widely used keypoint representation is the category specific stacked keypoint vector [33], which represents object keypoints by a $N \times D$ vector (N for number of keypoints and D for dimensions), or multi-channel heatmaps [22,32], which associate each channel with one specific keypoint on a specific object category, e.g., 16-channel heatmaps for human [22,32], 10-channel heatmaps for chair [40]. Although these representations are certainly semantically meaningful (e.g., the first channel of human heatmaps is the left ankle), it does not satisfy the adaptive property, e.g., chairs with legged bases and swivel bases cannot be learned together due to varying number of keypoints. As a result, they can not be considered as the same category based on their different keypoint configurations. To generalize heatmaps to multiple categories, a popular approach is to stack all heatmaps from all categories [24,35] (resulting in $\sum N_c$ output channels, where N_c is the number of keypoints of category c). In such a representation, keypoints from different objects are completely separated, e.g. seat corners from swivel chairs are irrelevant to seat corners from chairs. To merge keypoints from different objects, one has to establish consistent correspondences [48] between different keypoints across multiple categories, which is difficult or sometimes impossible.

In this paper, we introduce a hybrid representation that meets all desired properties. As illustrated in Fig. 2, our hybrid representation consists of three components, *StarMap*, *CanViewFeature* and *DepthMap*. In particular, StarMap specifies the image coordinates of keypoints where the number of keypoints can *vary* across different categories; CanViewFeature specifies the 3D locations of keypoints in a canonical coordinate system, which provide an identity for each keypoint; DepthMap lifts 2D keypoints into 3D. As we will see later, it enhances

the performance of using this representation for the application of viewpoint estimation. Now we describe each component in more details.

StarMap. As shown in Fig. 2 (top left), StarMap is a single channel heatmap whose local maximums encode the image locations of the underlying points. It is motivated by the success of using one heatmap to encode occurrences of one keypoint on multiple persons [4, 21]. In our setting, we generalize the idea to encode *all* keypoints of each object. This is in contrast to [4, 21], which use multi-peak heatmaps to detect multiple *instances* of the *same* specific keypoint. In our implementation, given a heatmap, we extract the corresponding keypoints by detecting all local maximums, with respect to the 8-ring neighborhood whose values are above 0.05.

When comparing multi-channel heatmaps and a single channel heatmap, one intuition is that multi-channel heatmaps, which are category-specific and keypoint-specific representations, lead to better accuracy. However, as we will see later, using a single channel allows us to train the representation from bigger training data (multiple categories), leading to an overall better keypoint predictor. We also argue that a single-channel representation (1 channel vs 100+ channels on Pascal3D+ [42]) is favored when computational and memory resources are limited. On the other hand, *StarMap* alone does not provide the semantic meaning of each detected point. This drawback motivates the second component of our hybrid keypoint representation.

CanViewFeature. CanViewFeature collects the 3D locations of the keypoints in the canonical view. In our implementation, we allocate three channels for *CanViewFeature*. Specifically, after detecting a keypoint (peak) in *StarMap*, the values of these three channels at the corresponding pixel specify the 3D location in the canonical coordinate system. The design of *CanViewFeature* is motivated from recent works on embedding visual objects into latent spaces [30, 37]. Such latent spaces provide a shared platform for comparing and linking different visual objects. Our representation shares the same abstract idea, yet we make the embedding *explicit* in 3D (where we *can view* the learned representation) and learnable in a *supervised* manner. This enables additional applications such as viewpoint estimation, as we will discuss later. When considering the space of keypoint configurations in the canonical space, it is easy to find that the feature is invariant to object pose and image appearance (scale, translation, rotation, lighting), little-variance to object shape (e.g., left frontal wheels from different cars are always in the left frontal area), and little variance to object category (e.g., frontal wheels from different categories are always in bottom frontal area).

Although *CanViewFeature* only provides 3D locations, we can leverage this to classify the keypoints, by using nearest neighbor association on the category-level keypoint templates.

DepthMap. *CanViewFeature* and *StarMap* are related to each other via a similarity transform (rotation, translation, scaling) and a perspective projection. It is certainly possible to solve a non-linear optimization problem to recover the

underlying similarity transform. However, since the network predictions are not perfect, we found that this approach leads to sub-optimal results.

To stabilize this process and make the relation even simpler, we augment *StarMap* with one additional channel called *DepthMap*. The encoding is the same as *CanViewFeature*. More precisely, we first extract keypoints at peak locations and then access the corresponding pixels to obtain the depth values. When the camera intrinsic parameters are present, we use them to convert image coordinates and depth value into the true 3D location of the corresponding pixel. Otherwise, we assume weak-perspective projection, and directly use the image coordinates and depth value as an approximation of the underlying 3D location.

3.2   Learning Hybrid Keypoint Representation

Data Preparation. Training our hybrid representation requires annotations of 2D keypoints, their corresponding depths, and their corresponding 3D locations in the canonical view. We remark that such training data is feasible to obtain and publicly available [41,42]. 2D keypoint annotations per image are straightforward to retrieve [23] and thus widely available [2,3,14]. Also, annotating 3D keypoints of a CAD model [45] is not a hard task, given an interactive 3D UI such as MeshLab [5]. The canonical view of a CAD model is defined as the front view of an object with the largest 3D bounding box dimension scaled to $[-0.5, 0.5]$ (meaning it is zero centered). Note that just a few 3D CAD models need to be annotated for each category (about 10 per category), because keypoint configuration variation is orders of magnitude smaller than the image appearance variation. Given a collection of images and a small set of CAD models of the corresponding categories, a human annotator is asked to select the closest CAD model to the image's content, as done in Pascal3D+ and ObjectNet3D [41,42]. A coarse viewpoint is also annotated by manually dragging the selected CAD model to align the image appearance. In summary, all the annotations required to train our hybrid representation are relatively easy to acquire. We refer to [41,42] for more details on how to annotate such data.

We now describe how we calculate the depth annotation. Ideally, the transformation between the canonical view and image pixel coordinate is a full-perspective camera model:

$$s[u \ v \ 1]^T = \mathcal{A}[R|t][\overline{x} \ \overline{y} \ \overline{z} \ 1]^T, s.t., R^T R = I \qquad (1)$$

where $\mathcal{A}$ describes intrinsic camera parameters, (u, v) is the 2D keypoint location in the image coordinate system, $(\overline{x}, \overline{y}, \overline{z})$ is the 3D location in canonical coordinate system. R, t, and s are the rotation matrix (i.e. viewpoint), translation vector, and scale factor, respectively. However, the camera intrinsic parameters are most likely unavailable in testing scenarios. In those cases, a weak-perspective camera model is often applied to *approximate* the 3D-to-2D transformation for keypoint estimation [24,49], by changing Eq. 1 to

$$s[u - c_x \ v - c_y \ d]^T = [R|t][\overline{x} \ \overline{y} \ \overline{z} \ 1]^T, s.t., R^T R = I \qquad (2)$$

where (u, v) specifies the location of the keypoint, d is its associated depth, and (c_x, c_y) denotes the center of the image.

Letting $[x, y, z] = [R|t][\overline{x}, \overline{y}, \overline{z}, 1]^T$ be the transformed 3D keypoints in the metric space, we have $[u, v, d] = [x/s + c_x, y/s + c_y, z/s]$ (with unknown s), which transforms one point from the 3D metric space to the 2D pixel space with an augmented depth value d. In training, let N_c be the number of keypoints in category c. Both the viewpoint transformation matrix $[R|t]$ and the canonical points $\{\overline{x}_i, \overline{y}_i, \overline{z}_i\}_{i=1}^{N_c}$ are known, and we can calculate the rotated keypoints $\{x_i, y_i, z_i\}_{i=1}^{N_c}$. Moreover, the corresponding 2D keypoints $\{(u_i, v_i)\}_{i=1}^{N_c}$ are known, so we can simply solve the scale factor s by aligning the (u, v) and (x, y) plane bounding box size: $s = \frac{max(max_i(x_i) - min_i(x_i), max_i(y_i) - min_i(y_i))}{max(max_i(u_i) - min_i(u_i), max_i(v_i) - min_i(v_i))}$, which gives rise to the underlying depth value.

Network Training. As described above, we have full supervision for all of our 3 output components. Training is done as a supervised heatmap regression, i.e., we minimize the $L2$ distance between the output 5-channel heatmap and their ground truth. Note that for *CanViewFeature* and *DepthMap*, we only care about the output at peak locations. Following [20,21], we ignore the non-peak output locations rather than forcing them to be zero. This can be simply implemented by multiplying a mask matrix to both the network output and ground truth and then using a standard $L2$ loss.

Implementation Details. Our implementation is done in the PyTorch framework. We use a 2-stacks HourglassNetwork [22], which is the state-of-the-art architecture for 2D human pose estimation [2]. We trained our network using curriculum learning, i.e., we first train the network with only *StarMap* output for 90 epochs and then fine-tune the network with the *CanViewFeature* followed by *DepthMap* supervision for additional 90 epochs each. The whole training stages took about 2 days on one GTX 1080 TI GPU. All the hyper-parameters are set to the default values in the original Hourglass implementation [22].

3.3   Application in Viewpoint Estimation

The output of our approach (StarMap, DepthMap and CanViewFeature) can directly be used to estimate the viewpoint of the input image with respect to the canonical view (i.e., camera pose estimation). Specifically, Let $\mathbf{p}_i = (u_i - c_x, v_i - c_y, d_i)$ be the un-normalized 3D coordinate of keypoint p_i, where (c_x, c_y) is the image center. Let $\mathbf{q}_i$ be its counterpart in the canonical view. With $w_i \in [0, 1]$ we denote this keypoint's value on the heatmap, which indicates a confidence score. We solve for a similarity transformation between the image coordinate system and world coordinate system that is parameterized by a scalar $s \in \mathbb{R}^+$, a rotation $R \in SO(3)$, and a translation $\mathbf{t}$. This is done by minimizing the following objective function:

$$s^\star, R^\star, \mathbf{t}^\star = \underset{s, R, \mathbf{t}}{\text{argmin}} \sum_{i=1}^{N_I} w_i \|sR\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2. \tag{3}$$

Note that (3) admits an explicit solution as described in [7], which we include here for completeness. The optimal rotation is given by

$$R^\star = U \text{diag}(1, 1, \text{sign}(M))V^T, \qquad M := \sum_{i=1}^{N_I} w_i(\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{q}_i - \overline{\mathbf{q}}) \qquad (4)$$

where $U\Sigma V^T = M$ is the SVD and $\overline{\mathbf{p}}$, $\overline{\mathbf{q}}$ are the mean of $\mathbf{p}_i$, $\mathbf{q}_i$.

4 Experiments

In this section, we perform experimental evaluations on the proposed hybrid keypoint representation. We begin with describing the experimental setup in Sect. 4.1. We then evaluate the accuracy of our keypoint detector and the application in viewpoint estimation in Sects. 4.2 and 4.3, respectively. We then present advanced analysis of our hybrid keypoint representation in Sect. 4.4. Finally, we show that our category-agnostic keypoint representation can be extended to novel categories in Sect. 4.5. Table 5 collect some qualitative results, and more results are deferred to the supplementary material.

4.1 Experimental Setup

We use Pascal3D+ [42] as our major evaluation benchmark. This dataset contains 12 man-made object categories with 2 K to 4 K images per category. We make use of the following annotations in our training: object bounding box, category-specific 2D keypoints (annotations from [3]), approximate 3D CAD model of the object, viewpoint of the image, and category-specific 3D keypoint annotations (corresponds with the 2D keypoint configuration) in the canonical coordinate system defined on each CAD model. Following [28,35], evaluation is done on the subset of the validation set that is non-truncated and non-occluded, which contains 2113 samples in total. As the evaluation protocols and baseline approaches vary across different tasks, we will describe them for each specific set of evaluations.

4.2 Keypoint Localization and Classification

We first evaluate our method on the keypoint estimation task, which specifies the locations of the predicted keypoints. Since keypoint locations alone do not carry the identities of each keypoint and cannot be used as identity-specific evaluation, we perform the evaluation by using two protocols – namely, with identification inferred from our learned CanViewFeature or with oracle assigned identification. Specifically, for the first protocol, for each category, we calculate the mean of the locations of each keypoint in the world coordinate system among all CAD models and use this as the *category-level* template. We then associate each keypoint with the ID of its nearest mean annotated keypoint in the template. For the second protocol, we assume a perfect ID assignment (or keypoint classification)

by assigning the output keypoint ID as the closest annotation (in image coordinates). The second protocol can also be thought of as randomly perturbing the annotated keypoint order and picking the best one. Following the conventions [15,35], we use PCK($\alpha = 0.1$), or Percentage of Correct Keypoints, as the evaluation metric. PCK considers a keypoint to be correct if its $L2$ 2D pixel distance from the ground truth keypoint location is less than $0.1 \times max(h, w)$, where h and w are the object's bounding box dimensions.

Table 1. 2D Keypoint Localization Results. The results are shown in PCK($\alpha = 0.1$). Top: our result with nearest canonical feature as keypoint identification. Bottom: results with oracle keypoint identification.

PCK ($\alpha = 0.1$)	Aero	Bike	Boat	Bottle	Bus	Car	Chair	Table	Mbike	Sofa	Train	Tv	Mean
Long [15]	53.7	60.9	33.8	72.9	70.4	55.7	18.5	22.9	52.9	38.3	53.3	49.2	48.5
Tulsiani [35]	66.0	77.8	52.1	83.8	88.7	81.3	65.0	47.3	68.3	58.8	72.0	65.1	68.8
Pavlakos [24]	84.1	86.9	62.3	87.4	96.0	93.4	76.0	N/A	N/A	78.0	58.4	84.8	82.5
Ours	75.2	83.2	54.8	87.0	94.4	90.0	75.4	58.0	68.8	79.8	54.0	85.8	78.6
Pavlakos [24] Oracle Id	92.3	93.0	79.6	89.3	97.8	96.7	83.9	N/A	N/A	85.1	73.3	88.5	89.0
Ours Oracle Id	93.1	92.6	84.1	92.4	98.4	96.0	91.7	90.0	90.1	89.7	83.0	95.2	92.2

The keypoint localization and classification results are shown in Table 1. We show 3 state-of-the-art methods [15,24,35] for *category-specific* keypoint localization for comparison. The evaluation of [24] is done by ourselves based on their published model. For the first protocol, our result of 78.6% mean PCK($\alpha = 0.1$) is marginally better than the state-of-the-arts in 2014 [15,35], probably because we used a more up-to-date HourglassNetwork [22]. Our performance is slightly worse than [24], who uses the same Hourglass architecture but with stacked category-specific channels output ($\sum_c N_c$ output channels in total), which is expected. This is due to the error caused by incorrect keypoint ID association. We emphasize that all counterpart methods are category-specific, thus requiring ground truth object category as input while ours is general.

The second protocol (Bottom of Table 1) factors out the error caused by incorrect keypoint ID association. For a fair comparison, we also allow [24] to change its output order with the oracle nearest location (to eliminate the common left-right flip error [26]). We can see our score is 92.2%, which is 3.2% higher than that of Pavlakos et al [24]. This is quite encouraging since our approach is designed to be a general purpose keypoint predictor. This result shows that it is advantageous to train a unified network to predict keypoint locations, as this allows to train a single network with more relevant training data.

4.3   Viewpoint Estimation

Some qualitative results are shown in Table 5, and more results can be found in the supplementary material.

As a direct application, we evaluate our hybrid representation on the task of viewpoint estimation. The objective of viewpoint estimation is to predict the azimuth (a), elevation (e), and in-plane rotation (θ) of the image object with respect to the world coordinate system. In our experiment, we follow the conventions [28,35] by measuring the angle between the predicted rotation vector and the ground truth rotation vector: $\Delta(R_{pred}, R_{gt}) = \frac{||logm(R_{pred}^T R_{gt})||_{\mathcal{F}}}{\sqrt{2}}$, where $R = R_Z(\theta)R_X(e - \pi/2)R_Z(-a)$ transforms the viewpoint representation (a, e, θ) into a rotation matrix. Here R_X, R_Y and R_Z are rotations along X, Y and Z axis, respectively.

We consider two metrics that are commonly applied in the literature [19,24, 28,35], namely, Median Error, which is the median of the rotation angle error, and Accuracy at θ, which is the percentage of keypoints whose error is less than θ. We use $\theta = \frac{\pi}{6}$, which is a default setting in the literature.

A popular approach for solving viewpoint estimation is to cast the problem as bin classification by discretiziing the space of (a, e, θ) [18,19,28,35]. Since network architecture governs the performance of a neural network, we re-train the baseline models [35] with more modern network architectures [6]. We implemented a ResNet18 (**Res18-Specific**) with the same hyper-parameters as [35] (we also tried VGG [27] or ResNet50 [6] but observed very similar or worse performance).

We also want to remark that although viewpoint estimation itself is not a category-specific task, all the studied preview works have used a category-specific formulation, e.g., use *separate* last-layer bin classifiers for each category, resulting in $3 \times N_{categories} \times N_{bins}$ output units [34]. We also provide a general $3 \times N_{bins}$ viewpoint estimator as a baseline (**Res18-General**).

Table 2 compares our approach with previous techniques. Our method outperforms all previous methods and baselines in both testing metrics. Specifically with respect to MedErr, our approach achieved 10.4, which is lower than the prior state-of-the-art result reported in Mousavian et al [19]. In terms of $Acc_{\frac{\pi}{6}}$, our method outperforms the state-of-the-art result of Su et al [28]. This is a quite positive result, since [28] uses additional rendered images for training.

We further evaluate $Acc_{\frac{\pi}{18}}$, which assesses the percentage of very accurate predictions. In this case, we simply compare against our re-implemented Res18, which achieved similar results with other state-of-the-art techniques. As shown in Table 2, our approach is significantly better than Res18-General/Specific with respect to $Acc_{\frac{\pi}{18}}$. This shows the advantage of performing keypoint alignment for pose estimation.

Note that it is also possible to directly align CanViewFeature with StarMap for viewpoint estimation by a weak-perspective PnP [24] algorithm (**PnP** in Table 2). In this case, utilizing DepthMap outperforms the direct alignment by 8.1% in terms of $Acc_{\frac{\pi}{6}}$ and 1.75% in terms of $Acc_{\frac{\pi}{18}}$, respecctively. On one hand, this shows the usefulness of DepthMap, particularly when the prediction is noisy. On the other hand, the performance of both approaches becomes similar when the predictions are very accurate ($Acc_{\frac{\pi}{18}}$). This is expected since both approaches should output identical results when the predictions are perfect.

Table 2. Viewpoint Estimation on Pascal3D+ [42]. We compare our results with the state-of-the-arts and baselines. The results are shown in Median Error (lower better) and Accuracy (higher better).

	Aero	Bike	Boat	Bottle	Bus	Car	Chair	Table	Mbike	Sofa	Train	Tv	Mean
$MedErr$ (Tulsiani [35])	13.8	17.7	21.3	12.9	5.8	9.1	14.8	15.2	14.7	13.7	8.7	15.4	13.6
$MedErr$ (Pavlakos [24])	8.0	13.4	40.7	11.7	2.0	5.5	10.4	N/A	N/A	9.6	8.3	32.9	N/A
$MedErr$ (Mousavian [19])	13.6	12.5	22.8	8.3	3.1	5.8	11.9	12.5	12.3	12.8	6.3	11.9	11.1
$MedErr$ (Su [28])	15.4	14.8	25.6	9.3	3.6	6.0	9.7	10.8	16.7	9.5	6.1	12.6	11.7
$MedErr$ (Mahendran [18])	14.2	18.7	27.2	9.5	3.0	6.9	15.8	14.4	16.4	10.7	6.6	14.3	13.1
$MedErr$ (Res18-General)	14.3	16.7	26.9	13.2	5.8	8.8	17.7	26.7	15.7	14.4	8.8	16.2	13.3
$MedErr$ (Res18-Specific)	14.7	15.8	25.6	13.1	5.7	8.6	16.3	18.1	15.1	13.8	8.2	14.1	12.8
$MedErr$ (PnP)	9.5	14.0	43.6	9.9	3.3	6.6	11.4	64.9	14.3	11.5	7.7	21.8	11.2
$MedErr$ (Ours)	10.1	14.5	30.0	9.1	3.1	6.5	11.0	23.7	14.1	11.1	7.4	13.0	**10.4**
$Acc_{\frac{\pi}{6}}$ (Tulsiani [35])	0.81	0.77	0.59	0.93	0.98	0.89	0.80	0.62	0.88	0.82	0.80	0.80	0.8075
$Acc_{\frac{\pi}{6}}$ (Pavlakos [24])	0.81	0.78	0.44	0.79	0.96	0.90	0.80	N/A	N/A	0.74	0.79	0.66	N/A
$Acc_{\frac{\pi}{6}}$ (Mousavian [19])	0.78	0.83	0.57	0.93	0.94	0.90	0.80	0.68	0.86	0.82	0.82	0.85	0.8103
$Acc_{\frac{\pi}{6}}$ (Su [28])	0.74	0.83	0.52	0.91	0.91	0.88	0.86	0.73	0.78	0.90	0.86	0.92	0.82
$Acc_{\frac{\pi}{6}}$ (Res18-General)	0.79	0.75	0.53	0.90	0.96	0.93	0.62	0.57	0.85	0.82	0.81	0.77	0.7875
$Acc_{\frac{\pi}{6}}$ (Res18-Specific)	0.79	0.77	0.54	0.93	0.95	0.93	0.75	0.57	0.84	0.79	0.81	0.84	0.8121
$Acc_{\frac{\pi}{6}}$ (PnP)	0.80	0.70	0.37	0.88	0.94	0.86	0.76	0.48	0.80	0.92	0.74	0.57	0.7416
$Acc_{\frac{\pi}{6}}$ (Ours)	0.82	0.86	0.50	0.92	0.97	0.92	0.79	0.62	0.88	0.92	0.77	0.83	**0.8225**
$Acc_{\frac{\pi}{18}}$ (Res18-General)	0.28	0.18	0.17	0.27	0.82	0.61	0.23	0.33	0.18	0.15	0.61	0.27	0.3502
$Acc_{\frac{\pi}{18}}$ (Res18-Specific)	0.29	0.21	0.21	0.30	0.86	0.62	0.28	0.33	0.21	0.18	0.59	0.30	0.3777
$Acc_{\frac{\pi}{18}}$ (PnP)	0.52	0.36	0.13	0.50	0.83	0.65	0.48	0.29	0.31	0.44	0.61	0.27	0.4643
$Acc_{\frac{\pi}{18}}$ (Ours)	0.49	0.34	0.14	0.56	0.89	0.68	0.45	0.29	0.28	0.46	0.58	0.37	**0.4818**

4.4 Analysis of Our Hybrid Keypoint Representation

Analysis of CanViewFeature. We use the ground-truth keypoint location, and compare their learned 3D locations for keypoint classification with popular point features used in the literature, namely, SIFT [16] and Conv5 of VGG [27]. For CanViewFeature, we still follow the same procedure of using nearest neighbor for keypoint classification. For SIFT and Conv5, a linear SVM is used to classify the keypoints [15].

Table 3 compares CanViewFeature with the two baseline approaches from [15]. We can see that CanViewFeature is significantly better than baseline

Table 3. Results for keypoint classification on Pascal3D+ Dataset [42]. We show keypoint classification accuracy of each category.

	Aero	Bike	Boat	Bottle	Bus	Car	Chair	Table	Mbike	Sofa	Train	Tv	Mean
SIFT [15]	35	54	41	76	68	47	39	69	49	52	74	78	57
Conv [15]	44	53	42	78	70	45	41	68	53	52	73	76	58
Ours	77	79	64	96	95	92	84	66	71	90	65	94	81

Table 4. Error analysis on Pascal3D+. We show results in Median Error and Accuracy.

	Aero	Bike	Boat	Bottle	Bus	Car	Chair	Table	Mbike	Sofa	Train	Tv	Mean
$MedErr$ (Ours)	10.1	14.5	30.0	9.1	3.1	6.5	11.0	23.7	14.1	11.1	7.4	13.0	10.43
$MedErr$ (GT Star)	9.2	13.3	31.3	8.2	3.1	5.7	10.7	78.2	13.8	10.1	7.0	13.4	9.92
$MedErr$ (GT Star+SCSF)	7.7	12.9	22.0	8.0	3.0	5.9	9.3	14.6	10.8	8.3	6.3	12.9	9.1
$MedErr$ (GT Star+Depth)	6.2	6.2	14.1	2.4	2.1	3.9	6.5	72.9	7.0	5.4	6.8	1.9	4.7
$Acc_{\frac{\pi}{6}}$ (Ours)	0.82	0.86	0.50	0.92	0.97	0.92	0.79	0.62	0.88	0.92	0.77	0.83	0.8225
$Acc_{\frac{\pi}{6}}$ (GT Star)	0.85	0.84	0.50	0.92	0.96	0.93	0.80	0.38	0.85	0.90	0.77	0.82	0.8211
$Acc_{\frac{\pi}{6}}$ (GT Star+SCSF)	0.86	0.84	0.63	0.95	0.99	0.95	0.88	0.62	0.84	0.92	0.88	0.85	0.8651
$Acc_{\frac{\pi}{6}}$ (GT Star+Depth)	0.86	0.93	0.63	0.95	0.97	0.91	0.82	0.38	0.87	0.92	0.84	0.93	0.8637

approaches. This shows the advantage of using a shared keypoint representation for training a general purpose keypoint detector.

Ablation Study on Representation Components. To better understand the importance of each component of our representation and whether they are well-trained, we provide error analysis by replacing each output component with its ground truth. To this end, we use viewpoint estimation as the task for evaluation, and Table 4 summarizes the results. Specifically, replacing StarMap with its ground truth does not provides much performance gains in both metrics, indicating that StarMap is fairly accurate. This is justified by the high keypoint accuracy reported in Sect. 4.2. Moreover, replacing either CanViewFeature or DepthMap with the underlying ground truth provides considerable performance gains in terms of $Acc_{\frac{\pi}{6}}$. In particular, using perfect DepthMap leads noticeable decrease in median error. This is expected since the general task of estimating pixel depth remains quite challenging.

4.5 Keypoint and Viewpoint Induction for Novel Categories

Our keypoint representation is category-agnostic and is free to be extended to novel object categories [34].

We note that Pascal3D+ [42] only contains 12 categories and it is hard to learn common inter-category information with such limited category samples. To further verify the generalization ability of our method, we used a newly published large scale 3D dataset, ObjectNet3D [41]. ObjectNet3D [41] has the

Table 5. Qualitative results of our full pipeline on Pascal3D+ [42] Dataset. 1st column: the input image; 2nd column: our predicted StarMap (shown on image); 3rd column: extracted keypoints after taking local maximum on StarMap, we show ground truth in large dots and prediction in small circled dots (The RGB color of the point encodes xyz coordinate for correspondence; 4th column: our predicted CanViewFeature (triangle) and their ground truth (circle); 5th column: our prediced 3D uvd coordinates, obtained by uv from StarMap and d from DepthMap; 6th column: rotated 3D point with our predicted viewpoint (cross) and ground truth viewpoint (triangle).
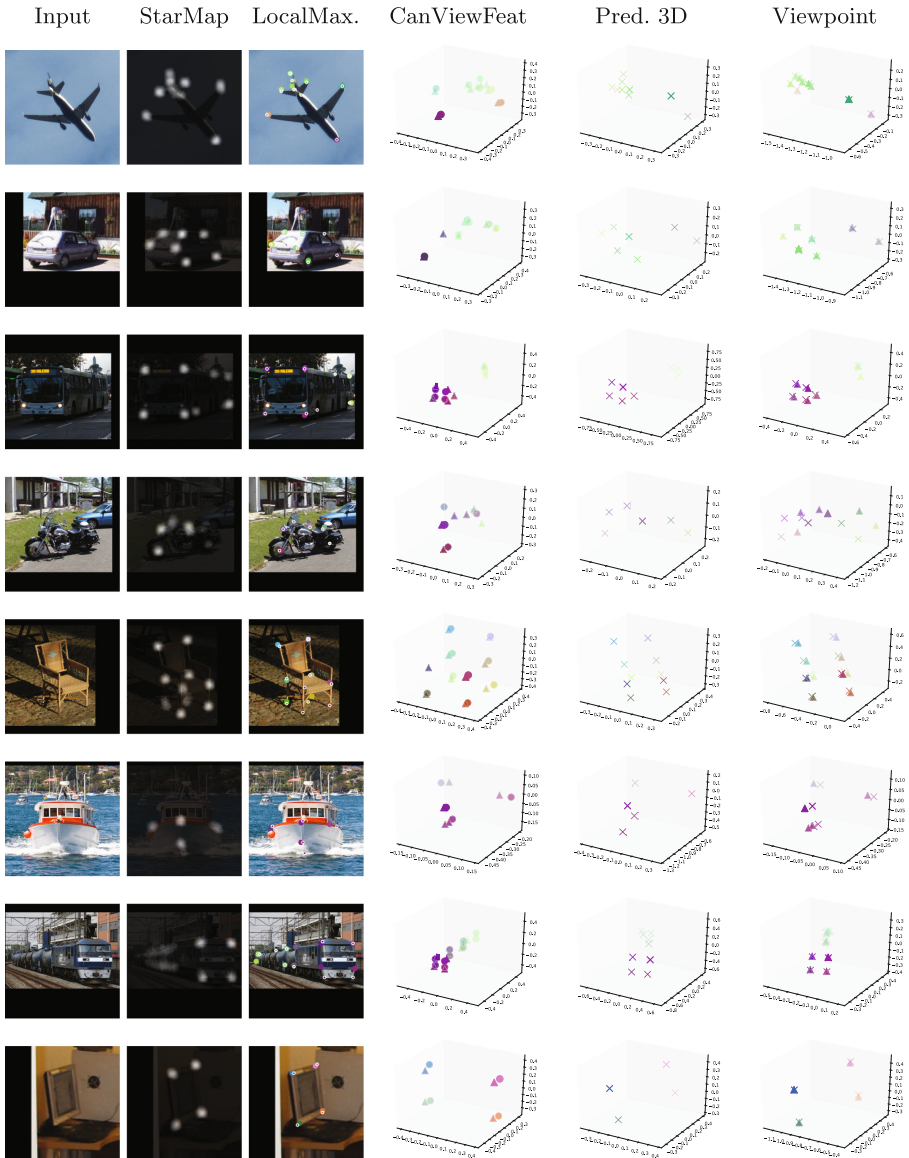
Input	StarMap	LocalMax.	CanViewFeat	Pred. 3D	Viewpoint

Table 6. Viewpoint estimation for novel categories results on ObjectNet3D+ [41]. We shown our results in $Acc_{\frac{\pi}{6}}$.

	Bed	Bookshelf	Calculator	Cellphone	Computer	Filing cabinet	Guitar	Iron	Knife
$Acc_{\frac{\pi}{6}}$ (Sup)	0.73	0.78	0.91	0.57	0.82	0.84	0.73	0.03	0.18
$Acc_{\frac{\pi}{6}}$ (Novel)	0.37	0.69	0.19	0.52	0.73	0.78	0.61	0.02	0.09
	Microwave	Pen	Pot	Rifle	Slipper	Stove	Toilet	Tub	Wheelchair
$Acc_{\frac{\pi}{6}}$ (Sup)	0.94	0.13	0.56	0.04	0.12	0.87	0.71	0.51	0.60
$Acc_{\frac{\pi}{6}}$ (Novel)	0.88	0.12	0.51	0.00	0.11	0.82	0.41	0.49	0.14

same annotations as Pascal3D+ [42] but with 100 categories. We evenly hold out 20 categories (every 5 categories sorted in the alphabetical order) from the training data and only used them for testing. Because Shoe and Door do not have keypoint annotation, we remove them from the testing set, resulting in 18 novel categories. Please refer to the supplementary for details on dataset details.

We compare the performance gap between including and withholding the 18 categories during training. The results are shown in Table 6. As expected, the viewpoint estimation accuracy of most categories drops. For some categories (Iron, Knife, Pen, Rifle, Slipper), both experiments fail (with accuracy lower than 20%). One explanation is that these 5 failed categories are small and narrow objects, whose annotations may not be accurate. For example, the keypoint annotations on ObjectNet3D [41] for small object are not always well-defined (see qualitative results in supplementary), e.g., Key and Spoon have dense keypoints annotation on their silhouette. For half of the 18 novel objects (bookshelf, cellphone, computer, filing cabinet, guitar, microwave, pot, stove, tub), the performance gap between including and withholding training data is less than 10%. This indicates that our representation is fairly general and can extend viewpoint estimation to novel categories.

References

1. Altwaijry, H., Veit, A., Belongie, S.J., Tech, C.: Learning to detect and match keypoints with deep architectures. In: BMVC (2016)
2. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: new benchmark and state of the art analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
3. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 168–181. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15567-3_13
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR, vol. 1, p. 7 (2017)

5. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: MeshLab: an open-source mesh processing tool. In: Scarano, V., Chiara, R.D., Erra, U. (eds.) Eurographics Italian Chapter Conference, The Eurographics Association (2008). https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136

6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

7. Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. JOSA A **4**(4), 629–642 (1987)

8. Huang, X., Shen, C., Boix, X., Zhao, Q.: Salicon: reducing the semantic gap in saliency prediction by adapting deep neural networks. In: ICCV (2015)

9. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. arXiv (2018)

10. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: Computer Vision and Pattern Regognition (CVPR) (2015)

11. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2938–2946. IEEE (2015)

12. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate O(n) solution to the PnP problem. Int. J. Comput. Vis. **81**(2), 155 (2009)

13. Li, S., Chan, A.B.: 3d human pose estimation from monocular images with deep convolutional neural network. In: Asian Conference on Computer Vision, pp. 332–347. Springer, Cham (2014)

14. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

15. Long, J.L., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: Advances in Neural Information Processing Systems, pp. 1601–1609 (2014)

16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)

17. Lu, C.P., Hager, G.D., Mjolsness, E.: Fast and globally convergent pose estimation from video images. IEEE Trans. Pattern Anal. Mach. Intell. **22**(6), 610–622 (2000)

18. Mahendran, S., Ali, H., Vidal, R.: Joint object category and 3d pose estimation from 2d images. arXiv preprint arXiv:1711.07426 (2017)

19. Mousavian, A., Anguelov, D., Flynn, J., Košecká, J.: 3d bounding box estimation using deep learning and geometry. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5632–5640. IEEE (2017)

20. Newell, A., Deng, J.: Pixels to graphs by associative embedding. In: Advances in Neural Information Processing Systems. pp. 2168–2177 (2017)

21. Newell, A., Huang, Z., Deng, J.: Associative embedding: end-to-end learning for joint detection and grouping. In: Advances in Neural Information Processing Systems, pp. 2274–2284 (2017)

22. Newell, A., Yang, K., Deng, J.: Stacked Hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29

23. Papadopoulos, D.P., Uijlings, J.R., Keller, F., Ferrari, V.: Extreme clicking for efficient object annotation. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4940–4949. IEEE (2017)

24. Pavlakos, G., Zhou, X., Chan, A., Derpanis, K.G., Daniilidis, K.: 6-DOF object pose from semantic keypoints. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2011–2018. IEEE (2017)
25. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1263–1272. IEEE (2017)
26. Ronchi, M.R., Perona, P.: Benchmarking and error diagnosis in multi-instance pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), October 2017
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. 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: Proceedings of the IEEE International Conference on Computer Vision, pp. 2686–2694 (2015)
29. Szeto, R., Corso, J.J.: Click here: human-localized keypoints as guidance for viewpoint estimation. arXiv preprint arXiv:1703.09859 (2017)
30. Taylor, J., Shotton, J., Sharp, T., Fitzgibbon, A.: The vitruvian manifold: inferring dense correspondences for one-shot human pose estimation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 103–110. IEEE (2012)
31. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–656 (2015)
32. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in neural information processing systems. pp. 1799–1807 (2014)
33. Toshev, A., Szegedy, C.: Deeppose: human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653–1660 (2014)
34. Tulsiani, S., Carreira, J., Malik, J.: Pose induction for novel object categories. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 64–72 (2015)
35. Tulsiani, S., Malik, J.: Viewpoints and keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1510–1519 (2015)
36. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Computer Vision and Pattern Regognition (CVPR) (2017)
37. Wei, L., Huang, Q., Ceylan, D., Vouga, E., Li, H.: Dense human body correspondences using convolutional networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, pp. 1544–1553 (2016)
38. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4732 (2016)
39. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, W.T., Tenenbaum, J.B.: MarrNet: 3D shape reconstruction via 2.5D sketches. In: Advances In Neural Information Processing Systems (2017)
40. 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

41. 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

42. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond PASCAL: a benchmark for 3d object detection in the wild. In: 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 75–82. IEEE (2014)

43. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057 (2015)

44. Yang, W., Li, S., Ouyang, W., Li, H., Wang, X.: Learning feature pyramids for human pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), vol. 2 (2017)

45. Yi, L., et al.: A scalable active framework for region annotation in 3d shape collections. ACM Trans. Graph. (TOG) **35**(6), 210 (2016)

46. Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J.Y., Lee, K.M., Molchanov, P., Kautz, J., Honari, S., Ge, L., et al.: 3d hand pose estimation: From current achievements to future goals. arXiv preprint arXiv:1712.03917 (2017)

47. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based R-CNNs for fine-grained category detection. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_54

48. Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 117–126 (2016)

49. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3d human pose estimation in the wild: a weakly-supervised approach. In: The IEEE International Conference on Computer Vision (ICCV), October 2017

50. Zhou, X., Karpur, A., Gan, C., Luo, L., Huang, Q.: Unsupervised domain adaptation for 3d keypoint prediction from a single depth scan. arXiv preprint arXiv:1712.05765 (2017)

51. Zhou, X., Sun, X., Zhang, W., Liang, S., Wei, Y.: Deep kinematic pose regression. arXiv preprint arXiv:1609.05317 (2016)

52. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. arXiv preprint arXiv:1606.06854 (2016)