Synthetic Depth Transfer for Monocular 3D Object Pose Estimation in the Wild

Yueying Kao,[1] Weiming Li,[1] Qiang Wang,[1] Zhouchen Lin,[2,1]* Wooshik Kim,[3] Sunghoon Hong[3]
[1]Samsung Research China - Beijing (SRC-B)
[2]Key Lab. of Machine Perception (MoE), School of EECS, Peking University
[3]Samsung Advanced Institute of Technology (SAIT)
{yueying.kao, weiming.li, qiang.w, zhouchen.lin, wooshik.kim, ar.sung.hong}@samsung.com

Abstract

Monocular object pose estimation is an important yet challenging computer vision problem. Depth features can provide useful information for pose estimation. However, existing methods rely on real depth images to extract depth features, leading to its difficulty on various applications. In this paper, we aim at extracting RGB and depth features from a single RGB image with the help of synthetic RGB-depth image pairs for object pose estimation. Specifically, a deep convolutional neural network is proposed with an RGB-to-Depth Embedding module and a Synthetic-Real Adaptation module. The embedding module is trained with synthetic pair data to learn a depth-oriented embedding space between RGB and depth images optimized for object pose estimation. The adaptation module is to further align distributions from synthetic to real data. Compared to existing methods, our method does not need any real depth images and can be trained easily with large-scale synthetic data. Extensive experiments and comparisons show that our method achieves best performance on a challenging public PASCAL 3D+ dataset in all the metrics, which substantiates the superiority of our method and the above modules.

Introduction

3D object pose estimation is to estimate an object's viewpoint (relative pose) with respect to a camera (including three angles: azimuth, elevation, and in-plane rotation). It is a core problem for many computer vision applications, such as robotics, augmented reality, autonomous driving and 3D scene interpretation. In the last decade, it has gained increasing attention and achieved promising success (Su et al. 2015; Sundermeyer et al. 2018).

Most existing methods (Su et al. 2015; Mousavian et al. 2017; Rad and Lepetit 2017) extract RGB (appearance) features from RGB images to estimate pose of objects. Despite of the significant progress in recent years, a major difficulty of these RGB based methods is induced by the 3D-2D projection process, where depth features are lost. Compared to RGB features, depth features are more invariant to illumination, texture, and background clutter. This makes them suitable to represent 3D geometry shape and thus important to infer 3D pose. Following this, depth features are extracted from real depth images for 3D object pose in depth-only based or RGB-D based methods (Balntas et al. 2017; Sahin and Kim 2018; Krull et al. 2015). In addition, a recent work (Rad, Oberweger, and Lepetit 2018) uses only depth features extracted from real RGB images with the help of real and synthetic depth images for pose estimation. However, real depth images are often unavailable in various real-world scenarios, due to various practical constraints such as sensor or computational cost limitations.

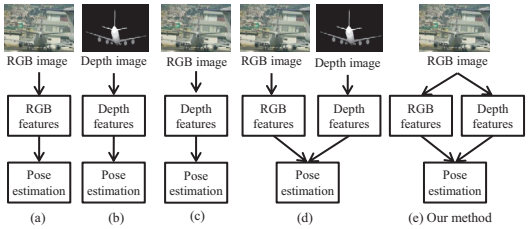

Figure 1: Different from previous methods (a),(b),(c),(d), our method extracts not only RGB features but also depth features, with the help of synthetic images, from only a single RGB image input for 3D object pose estimation (as shown in (e)). In (c), depth features are extracted from RGB images with the help of real depth images.

In this paper, different from previous approaches, we propose to extract RGB features and depth features from a single RGB image with the help of paired synthetic RGB-depth images for object pose estimation (as shown in Figure 1). Especially, our method does not need any real depth images in the training process. All the depth information needed in our task is transferred from synthetic data. This makes our method especially suitable for the challenging pose estimation in the wild task, where depth images are often unavailable (such as for far outdoor objects). Furthermore, collecting and labeling a large-scale training data of paired RGB and depth images is expensive and time-consuming. In stead of using real data, we render large-scale paired synthetic

*Z. Lin is supported by NSF China under grant no.s 61625301 and 61731018.

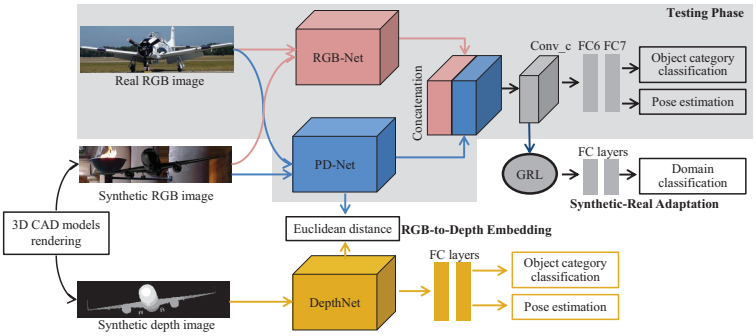

Figure 2: The overview of our proposed synthetic depth transfer method. At the training phase, the whole network is used. At the test phase in the grey area, only RGB-Net and PD-Net with the input of a real RGB image are utilized to estimate pose.

RGB and depth images with various poses from 3D CAD models with low cost. Despite the merits of using synthetic data, we have to address the issue of the significant domain gap between synthetic and real data, due to the difference in image formation settings, which often leads to a large performance drop on real data when the model is only trained with synthetic data. This requires the network design to include structures suitable to be trained with domain adaptation.

With all these considerations, this paper proposes a new network to extract RGB and depth features, with the help of synthetic data, from a single RGB image for object 3D pose estimation. Specifically, there are three streams in the network, as shown in Figure 2. One stream (RGB-Net) is trained to learn RGB features from RGB images for pose estimation. Another stream (DepthNet) is trained to learn object depth features from synthetic depth data for pose estimation. The third stream (PD-Net) is trained to extract pseudo depth features from RGB images with the guidance of DepthNet. The DepthNet and PD-Net jointly learn a depth oriented embedding space between RGB and depth images for pose estimation, which is called RGB-to-Depth Embedding. To train the network, a large number of synthetic RGB and depth image pairs are rendered with 3D CAD models. Then RGB features of RGB-Net and pseudo depth features of PD-Net from a same RGB image are combined to estimate the final pose of an object. Furthermore, due to the domain gap between real and synthetic RGB images, the combined features from different domains are aligned with a Synthetic-Real Adaptation module. In this way, synthetic depth features are transferred from synthetic depth images to real RGB images. At the testing phase, only RGB-Net and PD-Net are used to infer pose with a single RGB image of an object as input.

We evaluate the proposed method on a challenging public PASCAL 3D+ (Xiang, Mottaghi, and Savarese 2014) dataset. The experimental results in all the metrics show that our method outperforms the state-of-the-art methods (Tulsiani and Malik 2015; Su et al. 2015; Wu et al. 2016). Our ablative study demonstrates that: (1) the depth features extracted from RGB images are effective for pose estimation; (2) the fusion of RGB features and depth features extracted by our proposed network trained with only synthetic data still can achieve decent performance on real data; (3) the Synthetic-Real Adaptation can further improve our pose estimation performance.

In summary, our contributions are as follows:

1) We propose a framework to integrate synthetic RGB-depth image pairs to extract RGB and depth features to infer pose from single RGB images. By using paired synthetic data, we remove the availability obstacle of real depth image and can obtain large-scale training set with various poses.

2) To transfer depth feature from synthetic depth images to real RGB images for pose estimation, we propose two modules in our framework, an RGB-to-Depth Embedding and a Synthetic-Real Adaptation, which effectively transfers synthetic depth to RGB images and narrows the gap between real and synthetic RGB images.

3) Extensive experiments on the PASCAL 3D+ dataset demonstrate that our method achieves a decent improvement over state-of-the-art methods in all the metrics for 3D pose estimation, owing to the fusion of RGB and transferred depth features from synthetic data.

Related work

Estimation from RGB images. RGB images can provide appearance information for pose estimation, since objects with different poses have different appearance. Many RGB based pose estimation methods have been proposed. The early works include (Xiang, Mottaghi, and Savarese 2014; Pepik et al. 2012), which extend Deformable Part Models (DPM) to perform object detection and pose estimation. Later, CNN-based methods obtain great success for pose estimation from RGB images. Some methods (Su et al. 2015; Tulsiani and Malik 2015; Wang et al. 2018) take the pose estimation as a classification or regression problem, or a hybrid of them and train a CNN directly to estimate pose. For instance, a fine-grained pose classification formulation is proposed by (Su et al. 2015), with a geometric structure aware loss function considering the strong correlation of nearby views. Some methods (Pavlakos et al. 2017; Wu et al. 2016) predict 2D keypoints first from a single RGB image, and then predict pose with these keypoints. In addition, some methods (Li et al. 2017; Kao et al. 2018) take keypoints as auxiliary supervision and learn more powerful

presentations for pose estimation from a single RGB image.

Estimation from Depth or RGB-D images. Depth images are effective for pose estimation, which is demonstrated by many recent works (Balntas et al. 2017; Sahin and Kim 2018; Sock et al. 2017; Kehl et al. 2016). (Sahin and Kim 2018) address pose estimation with only depth images and obtain promising performance. In (Sundermeyer et al. 2018), depth images are used to refine the results inferred from RGB images. The studies of (Balntas et al. 2017; Krull et al. 2015; Li, Bai, and Hager 2018) focus on designing different RGB-D based approaches to estimate object pose. For example, in (Balntas et al. 2017) a depth image as a channel is concatenated with an RGB image and they are fed to a triplet network to learn an embedding for both object recognition and pose retrieval. (Krull et al. 2015) utilize a CNN as a probabilistic model to perform analysis-by-synthesis for object pose estimation based on RGB-D images. RGB-D images are also used to learn rich features for other tasks, such as object classification, object detection and segmentation. Since depth images are often unavailable in most real-world conditions, especially outdoor scenes, we aim to extract depth features from a single RGB image and fuse them with RGB features for pose estimation.

Estimation with Synthetic Data. Recently, many Domain Adaptation (DA) based methods (Ganin and Lempitsky 2015; Motiian et al. 2017) and various Generative Adversarial Networks (GAN) (Goodfellow et al. 2014) are proposed to generalize synthetic data to real data. Synthetic data have also been used for object pose estimation methods (Su et al. 2015; Szeto and Corso 2017; Grabner, Roth, and Lepetit 2018; Krull et al. 2015), due to the difficulty of collecting and labeling real data. These synthetic data are often rendered with 3D CAD models of objects. Su et al. (Su et al. 2015) render millions of synthetic RGB images together with a small amount of real images to train a CNN for pose estimation. Based on (Su et al. 2015), Szeto and Corso (Szeto and Corso 2017) render millions of synthetic RGB images with 2D keypoint information and propose a novel CNN that integrates an RGB image and a single keypoint map to predict viewpoint. However, they do not consider the domain shift between real data and synthetic data. In turn, an autoencoder for a novel 3D pose estimation in (Sundermeyer et al. 2018) is trained with only synthetic views of 3D models using domain randomization. (Sundermeyer et al. 2018) want to learn representations that are invariant to a significant domain gap between synthetic and real RGB images.

Recently, (Rad, Oberweger, and Lepetit 2018) aim to utilize synthetic depth images to extract mapped depth features from real RGB images without annotations for pose estimation. They make it in two steps by learning the feature mapping from real RGB to corresponding real depth images and bridging domain gap between real depth and synthetic depth images. However, real depth images are often unavailable in various real-world scenarios. The size of real data for network training is also small. In addition, since the paired real RGB-Depth images are without pose annotations, they cannot learn RGB features from real RGB images for pose estimation and the mapped depth features for pose are only based on synthetic depth data. The RGB features are proved to be very important for 3D pose estimation in the wild (Su et al. 2015; Tulsiani and Malik 2015). In comparison, our work does not use any real depth images. We can render virtually infinite paired synthetic RGB-Depth images with pose annotations to learn a supervised mapping from RGB to depth features effectively. The useful RGB features can also be learned and combined with transferred depth features for pose estimation. To further align the combined features from synthetic and real RGB images, a domain adaptation way (Ganin and Lempitsky 2015) is adopted. All these modules contribute to the improvement of our method.

Our Method

In this paper, our goal is to extract RGB features and depth features from a single RGB image for 3D pose estimation. Here we denote the 3D pose of an object as (α, β, θ), where α, β and θ are azimuth, elevation and in-plane rotation respectively. Following the previous works (Su et al. 2015; Szeto and Corso 2017), we formulate the pose estimation problem as a fine-grained classification problem, by dividing each angle into N bins ($N = 360$). Moreover, object category classification is also considered.

Network Overview

Figure 2 illustrates the proposed network. It consists of RGB-Net, PD-Net (Pseudo Depth net) and DepthNet. The basenet of the three nets we use is VGG-16 network (Simonyan and Zisserman 2015) before the first fully connected layer (FC6). RGB-Net is to learn RGB features from RGB images for object pose estimation. DepthNet is to learn object depth features from depth images for pose estimation. PD-Net is to learn pseudo depth features from RGB features by the *RGB-to-Depth Embedding* process. To integrate the complementarity of RGB features and depth features, the two features are fused for the final pose estimation. Since it is difficult to obtain pose annotation for large-scale paired RGB and depth images, we render millions of synthetic data for training the network. In addition, a *Synthetic-Real Adaptation* is also considered to reduce the domain gap of combined features between synthetic RGB and real RGB images. The RGB-to-Depth Embedding and Synthetic-Real Adaptation solve the synthetic depth transfer from synthetic depth images to real RGB images. At the test phase, only RGB-Net and PD-Net with the input of real RGB images are used to estimate pose.

RGB-to-Depth Embedding

Depth images can provide particular shape and geometric features under different poses. Different from RGB images, they are also invariant to illumination, texture and environment of objects. Since depth images are often unavailable in many practical applications, we aim to extract depth features from RGB images.

To extract depth features from RGB images, we propose to learn an RGB-to-Depth embedding space with synthetic data. This is motivated by multiple modalities embedding (Wang, Li, and Lazebnik 2016). For example, in (Wang,

Li, and Lazebnik 2016) for image and text, a joint embedding space is learned, where vectors from the two different modalities can be compared. Specifically, a two-branch network is used: one branch for images, and the other for text, which is then followed by L2 normalization at the output. We also aim to learn an embedding space for RGB images and depth images, where features from the two modalities can be compared. The difference is that we focus on the depth features for pose estimation in the embedding space. Inspired by these, we propose to learn a depth oriented embedding space between RGB and depth data with guidance of DepthNet. In this way, pseudo depth features can be extracted from RGB images by mapping the RGB images into the embedding space.

In detail, DepthNet with input of synthetic depth images x^s_{depth} is trained to learn depth features with pose estimation and object classification loss function L_d, where $L_d(x^s_{depth}, y) = L_{da}(x^s_{depth}, \alpha) + L_{de}(x^s_{depth}, \beta) + L_{dr}(x^s_{depth}, \theta) + L_{dc}(x^s_{depth}, c)$, and $y = \{\alpha, \beta, \theta, c\}$. $L_{da}(x^s_{depth}, \alpha)$, $L_{de}(x^s_{depth}, \beta)$ and $L_{dr}(x^s_{depth}, \theta)$ denote the loss functions for three angles respectively. $L_{dc}(x^s_{depth}, c)$ denotes object classification loss, where c denotes object category. PD-Net with input of synthetic RGB images x^s_{rgb} is to learn the mapping from RGB images to depth feature space with the guidance of DepthNet. Specifically, we minimize the distance of the feature maps of PD-Net and DepthNet, $Dis(f_d(x^s_{depth}), f_{ed}(x^s_{rgb}))$, where f_d denotes the feature maps from DepthNet, f_{ed} denotes the feature maps from PD-Net. In this paper, we use the feature maps of layers pool3, pool4 and pool5 of VGG-Net. The distance metric we use is Euclidean distance. The inputs of DepthNet and PD-Net are depth and RGB image pairs with the same pose. The final loss function for RGB-to-Depth embedding is

$$L_{Em} = L_d(x^s_{depth}, y) + \lambda Dis(f_d(x^s_{depth}), f_{ed}(x^s_{rgb})), \quad (1)$$

where $\lambda > 0$ is a constant. Training DepthNet and PD-Net with the final loss function L_{Em} is to learn the RGB-to-Depth embedding space for pose estimation.

Synthetic-Real Adaptation

Due to the difficulty of collecting and labeling real data for object pose estimation, we render a large number of synthetic data with pose annotations by 3D CAD models for learning RGB-to-Depth Embedding. However, synthetic RGB image x^s_{rgb} and real RGB image x^r_{rgb} (shown in Figure 2) look obviously different. They have a significant domain gap. In addition, when training networks, both RGB and depth features can be extracted from real and synthetic RGB images. Thus, we introduce a loss function of domain adaptation to align the combined RGB and depth features from real and synthetic RGB images,

$$L_{DA} = Da(x^s_{rgb}, x^r_{rgb}). \quad (2)$$

Domain adaptation (DA) has been widely studied in the literature. In this work, we apply a domain adaptation approach proposed by (Ganin and Lempitsky 2015). Specifically, a domain classifier is connected to the standard feature extractor layers (Conv_c) after feature combination via a gradient reversal layer (GRL), as shown in Figure 2. Here we set recognizing synthetic data or real data as a two-class classification problem. A softmax loss function is used. The domain classifier is to distinguish samples from two domains. In the feed-forward training process, the purpose is to minimize the domain classifier loss and other task losses. During back-propagation training, the GRL multiplies the gradient by a negative constant. The gradient reversal ensures that feature distributions of the two domains are made as indistinguishable as possible. In this adversarial training procedure, both RGB and depth features from different domains are adapted to be similar.

Training Objective

The loss of our whole training network includes the RGB-to-Depth Embedding loss, Synthetic-Real Adaptation loss and final pose estimation loss. The final pose estimation and object classification loss function is $L_{rgbd}(x_{rgb}, y) = L_a(x_{rgb}, \alpha) + L_e(x_{rgb}, \beta) + L_r(x_{rgb}, \theta) + L_c(x_{rgb}, c)$. The final pose estimation follows the fusion of RGB features and depth features from the same RGB images.

Since both unsupervised domain adaption (UDA) (Motian et al. 2017) and supervised domain adaption (SDA) (Ganin and Lempitsky 2015) are of interests to the research community, we consider UDA and SDA respectively in our network for different scenarios. UDA does not need target (real) data to be labeled, thus is attractive. SDA requires labeled target data and can obtain much better performance. For using UDA, real RGB images are unlabeled, only synthetic data are used to train RGB-to-Depth Embedding and final pose estimation. Our final objective loss will be

$$L = L_{Em} + L_{rgbd}(x^s_{rgb}, y) + L_{DA}. \quad (3)$$

For using SDA, real RGB images are labeled, and synthetic data are used to train RGB-to-Depth Embedding. Both real RGB images and synthetic RGB images are for final pose estimation. Our final objective loss will be

$$L = L_{Em} + L_{rgbd}(x_{rgb}, y) + L_{DA}, \quad (4)$$

where $x_{rgb} = \{x^r_{rgb}, x^s_{rgb}\}$ denotes a sample set including real RGB images and synthetic RGB images. For each angle classification in pose estimation, we use the geometric structure aware loss function proposed by (Su et al. 2015).

Implementation Details

Our proposed network is implemented by Caffe framework (Jia et al. 2014). The training process can be divided into three phases. 1) DepthNet and the following FC layers are initialized with VGG-Net trained on ImageNet (Deng et al. 2009) classification task (the same below). Synthetic depth images are used to train DepthNet for pose estimation with the loss function L_d. 2) RGB-to-Depth Embedding is trained with paired synthetic RGB and depth images. It includes DepthNet and PD-Net with the loss function L_{Em}. In this phase, we initialize DepthNet with parameters trained in the first phase, and initialize the PD-Net with VGG-Net trained on ImageNet. Synthetic depth images are fed into

DepthNet, and corresponding synthetic RGB images are fed into PD-Net. We set $\lambda = 0.01$. 3) we train the whole network with the final loss L. Real and synthetic RGB images are fed into RGB-Net, synthetic depth images are fed into DepthNet, and real and synthetic RGB images are also fed into PD-Net. We initialize DepthNet and PD-Net with parameters trained in the second phase, and initialize the RGB-Net with VGG-Net trained on ImageNet. To fuse the RGB features from RGB-Net and depth features from PD-Net, the pool5 from the two nets are concatenated and then followed by a convolutional layer Conv_c with 512 filters with size 3×3, and two fully connected layers, FC6, FC7. FC6 and FC7 both have 4096 nodes. In addition, Synthetic-Real Adaptation is also integrated to align the fused features from synthetic and real RGB images. It is implemented by connecting a GRL after layer Conv_c and being followed by two FC layers and a domain classifier. The number of nodes of the FC layers is set 1024 in this paper. In this way we train the whole network. At the test phase, real RGB images are fed into RGB-Net and PD-Net to infer the final pose.

Synthesizing Data for Training

There are two reasons for synthesizing data to train our network. The first is that large-scale depth images with pose annotations, especially companied with RGB images, are difficult to obtain in the real world. The other is that the most popular PASCAL 3D+ (Xiang, Mottaghi, and Savarese 2014) dataset contains about 27K object instances for object pose estimation. It is insufficient for training our network, even only the RGB-Net. Therefore we utilize a large number of 3D CAD models in ShapeNet (Chang et al. 2015) to render about two millions synthetic data, including RGB images and depth images with labeled pose. We extend the synthetic rendering pipeline proposed by (Su et al. 2015). Firstly, we sample lighting condition randomly and camera extrinsics from a real image training set (here we use PASCAL 3D+). Secondly, we render the CAD models to obtain paired synthetic RGB and depth images, and then randomly sample an image from the SUN397 (Xiao et al. 2010) dataset as background of the synthetic RGB image. Finally, we crop the paired RGB image and depth image with a same perturbed object bounding box. The cropping parameters are also learned from the real dataset.

Experiments

Experimental setup

Dataset. We evaluate the proposed method on a public PASCAL 3D+ (Xiang, Mottaghi, and Savarese 2014) dataset, including 12 object categories. There are annotations of pose, object classes and object bounding boxes in this dataset. The real images in this dataset are from PASCAL VOC detection training and validation set, and ImageNet dataset. 27,348 object instances from PASCAL training set and ImageNet images with ground truth (GT) bounding boxes, and synthetic images are used to train our network. We synthesize about 200K pairs of RGB images and depth images per category, and in total 2,168,764 pairs for 12 categories. All of them have accurate 3D pose and category annotations. The whole PASCAL 3D+ validation set is used to evaluate our performance.

Evaluation Metrics. To be consistent with previous works (Xiang, Mottaghi, and Savarese 2014; Su et al. 2015; Tulsiani and Malik 2015), we use $Acc_{\pi/6}$, $MedErr$ and AVP (Average Viewpoint Precision) as the evaluation metrics. $Acc_{\pi/6}$ and $MedErr$ (Tulsiani and Malik 2015) are based on the geodesic distance between predicted rotation matrix R_{pr} and ground truth rotation matrix R_{gt}, $\Delta(R_{pr}, R_{gt}) = ||\log(R_{pr}^T R_{gt})||_F/\sqrt{2}$. Rotation matrix can equivalently describe the three angles (azimuth, elevation, and in-plane rotation). $Acc_{\pi/6}$ is defined as the percentage of test instances where $\Delta(R_{pr}, R_{gt}) < \pi/6$. $MedErr$ is median error of $\Delta(R_{pr}, R_{gt})$ for all test instances. The two metrics (Tulsiani and Malik 2015) are presented to evaluate 3D pose estimation performance with ground truth (GT) bounding boxes. AVP (Xiang, Mottaghi, and Savarese 2014) is used to evaluate methods for joint detection and pose estimation. When computing AVP, the result is correct only if both of detection result and viewpoint (azimuth) are correct, similar to (Tulsiani and Malik 2015; Su et al. 2015; Xiang, Mottaghi, and Savarese 2014; Pepik et al. 2012).

Comparison with State-of-the-art Methods

To validate our method, we compare our method with state-of-the-art methods (Tulsiani and Malik 2015; Su et al. 2015; Kao et al. 2018; Wang et al. 2018; Mousavian et al. 2017; Grabner, Roth, and Lepetit 2018) with only a single RGB image as input. The methods (Tulsiani and Malik 2015; Kao et al. 2018; Wang et al. 2018; Mousavian et al. 2017; Grabner, Roth, and Lepetit 2018) only use real images to train their networks, although some of them augment the training data by using flipped images or the jittered bounding boxes. A baseline, RGB-Net (Real), which is similar to the RGB-Net in Figure 2 and R4CNN proposed by (Su et al. 2015), is trained only on the real RGB images. The real images are flipped to augment the training data. In addition, since R4CNN (Su et al. 2015) is trained on a combination of real images and synthetic (Syn) images with a basenet AlexNet (Krizhevsky, Sutskever, and Hinton 2012), another baseline (RGB-Net (Real+Syn)) is also implemented by training RGB-Net on the combination of real images and synthetic images with a basenet VGG-Net. The difference between RGB-Net (Real+Syn) and R4CNN is their basenet. The differences between RGB-Net (Real+Syn) and our method are our proposed RGB-to-Depth Embedding and Synthetic-Real Adaptation.

Pose Estimation with Ground Truth Bounding Box. Table 1 shows the performance of our method, baselines and state-of-the-art methods for 3D pose estimation with GT bounding boxes on PASCAL 3D+ dataset. Here our method (final) in this table means the whole network trained with SDA. We can see that our method outperforms all the state-of-the-art methods. By comparing the two baselines, it indicates that synthetic data can augment the training data effectively. In addition, the comparison of RGB-Net (Real+Syn) and our method verifies the effectiveness of our method. It demonstrates that the proposed RGB-to-Depth Embedding

Table 1: $Acc_{\pi/6}$ (%) and $MedErr$ of different methods for 3D pose estimation with GT bounding boxes on PASCAL 3D+.

	basenet	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
$Acc_{\pi/6}$ (V&K (Tulsiani and Malik 2015))	VGG-Net	81	77	59	93	98	89	80	62	88	82	80	80	81
$Acc_{\pi/6}$ (R4CNN (Su et al. 2015))	AlexNet	74	83	52	91	91	88	86	73	78	90	**86**	92	82
$Acc_{\pi/6}$ (ASFnet (Kao et al. 2018))	VGG-Net	86.6	88.1	58.6	93.3	**98.7**	86.5	78.5	82.6	89.8	85.0	84.1	90.1	85.2
$Acc_{\pi/6}$ (HCR-Net (Wang et al. 2018))	VGG-Net	81	**89**	67	95	97	89	79	76	**93**	87	83	91	86
$Acc_{\pi/6}$ (MultiBin (Mousavian et al. 2017))	VGG-Net	78	83	57	93	94	90	80	68	86	82	82	85	81.03
$Acc_{\pi/6}$ ((Grabner, Roth, and Lepetit 2018))	ResNet	83	82	64	95	97	**94**	80	71	88	87	80	86	83.92
$Acc_{\pi/6}$ (RGB-Net (Real))	VGG-Net	84.0	84.7	62.9	**97.6**	94.2	89.0	76.6	61.9	86.0	71.8	83.2	89.2	81.8
$Acc_{\pi/6}$ (RGB-Net (Real+Syn))	VGG-Net	85.8	83.9	63.4	92.0	89.6	90.9	85.2	81.0	84.6	**94.9**	84.1	**93.2**	85.7
$Acc_{\pi/6}$ (Ours (final))	VGG-Net	**88.0**	87.3	**67.2**	96.0	96.8	93.5	**86.9**	**95.2**	91.9	92.3	85.0	92.8	**89.4**
$MedErr$ (V&K (Tulsiani and Malik 2015))	VGG-Net	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
$MedErr$ (R4CNN (Su et al. 2015))	AlexNet	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
$MedErr$ (ASFnet (Kao et al. 2018))	VGG-Net	**7.4**	**10.7**	18.5	6.1	**1.8**	4.0	8.2	7.5	**9.0**	8.1	**3.7**	9.7	7.9
$MedErr$ (HCR-Net (Wang et al. 2018))	VGG-Net	9.2	12.0	16.5	6.2	2.4	4.5	12.2	8.1	11.2	8.2	4.67	11.2	8.9
$MedErr$ (MultiBin (Mousavian et al. 2017))	VGG-Net	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
$MedErr$ ((Grabner, Roth, and Lepetit 2018))	ResNet	10.0	15.6	19.1	8.6	3.3	5.1	13.7	11.8	12.2	13.5	6.7	11.0	10.9
$MedErr$ (RGB-Net (Real))	VGG-Net	8.6	11.9	16.3	6.5	2.0	3.8	10.0	11.8	12.0	10.1	4.6	9.9	8.9
$MedErr$ (RGB-Net (Real+Syn))	VGG-Net	8.8	11.7	18.6	6.3	2.5	4.5	8.3	8.3	11.6	7.7	4.6	8.9	8.5
$MedErr$ (Ours (final))	VGG-Net	7.7	11.5	**15.8**	**5.5**	2.0	**3.6**	**7.2**	**4.9**	9.3	**7.2**	4.3	**8.4**	**7.3**

and Synthetic-Real Adaptation contribute to the improvements from RGB-Net (Real+Syn) to our method on the two metrics.

Joint Detection and Pose Estimation. To further validate our method, we follow prior works (Xiang, Mottaghi, and Savarese 2014; Felzenszwalb et al. 2010; Su et al. 2015; Tulsiani and Malik 2015; Poirson et al. 2016; Wang et al. 2018) and test on the joint detection and pose estimation task. Table 2 shows the performance of our method and state-of-the-art methods. Firstly, we compare pose estimation performance with the same detection results. One of our methods in Table 2, Ours(final)+RCNN, uses the bounding boxes detected from RCNN (Girshick et al. 2014). The detected results are provided by (Tulsiani and Malik 2015) and its AP (Average Precision) on the 12 object categories of Pascal 3D+ dataset is 60.4%. HCR-Net (Wang et al. 2018) also uses the detected results. By comparing the three methods, we can see that our pose estimation method outperforms the other two methods. It demonstrates the benefit of our method. In addition, since different detection results with the same pose estimation methods may lead to different AVPs, we also show the pose estimation performance in Table 2 with SSD512 detector provided by (Liu et al. 2016). It is termed as Ours(final)+SSD512. The AP of SSD512 on the 12 object categories of PASCAL 3D+ dataset is 89.2%, which is much higher than RCNN. It indicates that higher detection performance can make higher AVP. Poirson et al. (Poirson et al. 2016) extend a SSD500 network (Extended SSD500) trained on PASCAL 3D+ dataset to detect object and estimate its pose simultaneously. Table 2 shows that our method with SSD512 detector outperforms other methods significantly. It verifies the effectiveness of our method.

Ablative Study

To investigate the importance and effect of synthetic data, RGB-to-Depth Embedding and Synthetic-Real Adaptation for our method, we do an ablative study and show the results of our method with or without one or more modules with GT bounding box in Table 3. We train 10 models. The PD-Net (Syn) means that PD-Net and DepthNet are trained on synthetic RGB-depth image pairs. The depth features extracted from PD-Net are used to evaluate pose estimation. Some models with SDA are trained on labeled real data and labeled synthetic data, while some models with UDA are trained on labeled synthetic data and unlabeled real data.

Effect of Synthetic Data. By comparing the performance of RGB-Net (Real) and RGB-Net (Real+Syn), it demonstrates that synthetic images can augment the training data and improves the performance effectively, although RGB-Net (Real+Syn) does not consider the domain gap between synthetic and real data.

Effect of RGB-to-Depth Embedding. From the results of PD-Net (Syn), RGB-Net (Syn) and RGB-Net+PD-Net (Syn), we can see that depth features only or RGB features only from synthetic data for pose estimation performs ordinarily. The fusion of the two features (RGB-Net+ PD-Net (Syn)) outperforms PD-Net (Syn) and RGB-Net (Syn) significantly. It demonstrates the complementarity between the RGB features and the depth features and their effectiveness for pose estimation. It also shows that the combination of the two features learned with only synthetic data can still obtain decent performance. In addition, the comparison between RGB-Net (Real+Syn) and RGB-Net+PD-Net (Real+Syn) also verifies the effectiveness of RGB-to-Depth Embedding.

Effect of Synthetic-Real Adaptation. From the results of RGB-Net (Syn) and RGB-Net (Real), we can see that there is a significant domain gap between synthetic and real data. Synthetic-Real Adaptation can improve the pose estimation performance effectively, by comparing the same networks with or without domain adaptation, regardless of its supervision condition. It is also demonstrated that the integration of Synthetic-Real Adaptation and RGB-to-Depth Embedding can further improve our pose estimation performance.

Qualitative Results

To further verify the effectiveness of our method, we show some examples whose error predicted by our final method $\Delta(R_{pr}, R_{gt}) < \pi/6$, while estimated by RGB-Net (Real+Syn) $\Delta(R_{pr}, R_{gt}) > \pi/6$ in Figure 3. We can see that

Table 2: Joint detection and pose estimation on PASCAL 3D+ dataset. We show AVPs for four quantization cases that the 360-degree views of azimuth are discretized to 4, 8, 16, 24 bins respectively.

AVP	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Avg.
VDPM-4V (Xiang, Mottaghi, and Savarese 2014)	34.6	41.7	1.5	-	26.1	20.2	6.8	3.1	30.4	5.1	10.7	34.7	19.5
VDPM-8V (Xiang, Mottaghi, and Savarese 2014)	23.4	36.5	1.0	-	35.5	23.5	5.8	3.6	25.1	12.5	10.9	27.4	18.7
VDPM-16V (Xiang, Mottaghi, and Savarese 2014)	15.4	18.4	0.5	-	46.9	18.1	6.0	2.2	16.1	10.0	22.1	16.3	15.6
VDPM-24V (Xiang, Mottaghi, and Savarese 2014)	8.0	14.3	0.3	-	39.2	13.7	4.4	3.6	10.1	8.2	20.0	11.2	12.1
DPM-VOC+VP-4V (Felzenszwalb et al. 2010)	37.4	43.9	0.3	-	48.6	36.9	6.1	2.1	31.8	11.8	11.1	32.2	23.8
DPM-VOC+VP-8V (Felzenszwalb et al. 2010)	28.6	40.3	0.2	-	38.0	36.6	9.4	2.6	32.0	11.0	9.8	28.6	21.5
DPM-VOC+VP-16V (Felzenszwalb et al. 2010)	15.9	22.9	0.3	-	49.0	29.6	6.1	2.3	16.7	7.1	20.2	19.9	17.3
DPM-VOC+VP-24V (Felzenszwalb et al. 2010)	9.7	16.7	2.2	-	42.1	24.6	4.2	2.1	10.5	4.1	20.7	12.9	13.6
R4CNN-4V (Su et al. 2015)	54.0	50.5	15.1	-	57.1	41.8	15.7	18.6	50.8	28.4	46.1	58.2	39.7
R4CNN-8V (Su et al. 2015)	44.5	41.1	10.1	-	48.0	36.6	13.7	15.1	39.9	26.8	39.1	46.5	32.9
R4CNN-16V (Su et al. 2015)	27.5	25.8	6.5	-	45.8	29.7	8.5	12.0	31.4	17.7	29.7	31.4	24.2
R4CNN-24V (Su et al. 2015)	21.5	22.0	4.1	-	38.6	25.5	7.4	11.0	24.4	15.0	28.0	19.8	19.8
Extended SSD500-4V (Poirson et al. 2016)	64.6	62.1	26.8	-	70.0	51.4	11.3	40.7	62.7	40.6	65.9	61.2	50.7
Extended SSD500-8V (Poirson et al. 2016)	58.6	56.4	19.9	-	62.4	45.2	10.6	34.7	58.6	38.8	61.2	49.7	45.1
Extended SSD500-16V (Poirson et al. 2016)	45.9	39.6	14.0	-	54.0	35.4	7.4	26.4	40.4	29.2	41.5	35.8	33.6
Extended SSD500-24V (Poirson et al. 2016)	33.4	29.4	9.2	-	54.7	35.7	5.5	22.9	30.3	27.5	44.1	24.3	28.8
V&K(RCNN)-4V (Tulsiani and Malik 2015)	63.1	59.4	23	-	69.8	55.2	25.1	24.3	61.1	43.8	59.4	55.4	49.1
V&K(RCNN)-8V (Tulsiani and Malik 2015)	57.5	54.8	18.9	-	59.4	51.5	24.7	20.4	59.5	43.7	53.3	45.6	44.5
V&K(RCNN)-16V (Tulsiani and Malik 2015)	46.6	42	12.7	-	64.6	42.8	20.8	18.5	38.8	33.5	42.4	32.9	36.0
V&K(RCNN)-24V (Tulsiani and Malik 2015)	37.0	33.4	10.0	-	54.1	40.0	17.5	19.9	34.3	28.9	43.9	22.7	31.1
HCR-Net+RCNN-4V (Wang et al. 2018)	63.3	63.4	24.1	-	71.8	55.7	25.6	29.9	68.0	53.9	62.4	59.4	52.6
HCR-Net+RCNN-8V (Wang et al. 2018)	59.1	54.2	19.3	-	64.3	51.7	23.7	24.9	56.7	50.4	55.1	48.2	46.4
HCR-Net+RCNN-16V (Wang et al. 2018)	45.0	36.6	13.0	-	61.7	42.3	16.4	21.5	35.2	37.7	46.5	33.3	34.4
HCR-Net+RCNN-24V (Wang et al. 2018)	36.4	28.8	9.0	-	58.6	36.9	12.1	14.9	31.5	31.4	43.8	22.9	29.3
Ours(final)+RCNN-4V	64.9	61.7	27.7	-	72.8	58.2	28.9	31.6	66.0	52.7	63.2	60.0	53.4
Ours(final)+RCNN-8V	59.7	57.1	18.0	-	64.9	54.5	28.0	29.1	61.7	50.9	56.9	50.9	48.3
Ours(final)+RCNN-16V	49.4	40.6	13.9	-	65.4	45.2	24.6	22.0	45.7	36.9	51.6	38.2	39.4
Ours(final)+RCNN-24V	37.2	29.9	9.2	-	57.2	44.5	21.0	23.6	34.8	32.8	49.6	30.3	33.6
Ours(final)+SSD512-4V	**79.6**	**80.2**	**55.3**	-	**87.5**	**79.4**	**63.1**	**50.7**	**81.2**	**72.4**	**76.9**	**88.4**	**74.1**
Ours(final)+SSD512-8V	**70.6**	**76.3**	**45.3**	-	**78.0**	**73.8**	**56.5**	**45.9**	**75.9**	**69.2**	**69.5**	**74.4**	**66.8**
Ours(final)+SSD512-16V	**59.3**	**49.1**	**30.0**	-	**78.2**	**60.7**	**45.9**	**40.1**	**56.6**	**52.9**	**61.8**	**57.9**	**53.8**
Ours(final)+SSD512-24V	**46.3**	**43.0**	**24.2**	-	**67.8**	**59.7**	**38.4**	**35.1**	**46.5**	**43.6**	**58.4**	**47.6**	**46.4**

Table 3: Ablative Study of our proposed method with ground truth bounding box on Pascal 3D+ dataset.

Model (training data)	$Acc_{\pi/6}$	$MedErr$
PD-Net (Syn)	48.8	38.3
RGB-Net (Syn)	43.9	41.2
RGB-Net+UDA (Real+Syn)	79.5	12.7
RGB-Net+PD-Net (Syn)	76.1	13.4
RGB-Net+PD-Net+UDA (Real+Syn)	80.8	12.3
RGB-Net (Real)	81.8	8.9
RGB-Net (Real+Syn)	85.7	8.5
RGB-Net+SDA (Real+Syn)	87.2	**7.3**
RGB-Net+PD-Net (Real+Syn)	88.2	7.4
RGB-Net+PD-Net+SDA (Real+Syn)	**89.4**	**7.3**

Figure 3: We show some examples whose error predicted by our final method (the third row) $\Delta(R_{pr}, R_{gt}) < \pi/6$, while estimated by RGB-Net (Real+Syn)(the second row) $\Delta(R_{pr}, R_{gt}) > \pi/6$.

Figure 4: Failure cases. For each image, its 3D model is rendered with our predicted pose, which is opposite to GT.

our method can handle instances with complex background, low resolution (small or far objects) and unusual pose much better. We also find that the method RGB-Net (Real+Syn) sometimes confuses the front view and rear view for aeroplane, bus, car, etc. Our method can correct this error for many cases. All of these owe to the RGB-to-Depth Embedding and Synthetic-Real Adaptation.

Additionally, we also analyze the error with our method and show failure cases in Figure 4. We follow (Tulsiani and Malik 2015) and define 'large objects' as the top third of instances and 'small objects' as the bottom third of instances. Their $Acc_{\pi/6}$'s are 92.4% and 85.3% respectively. There is a significant difference between them. Such a phenomenon is common in existing methods, because small (far) objects are often with very low resolution and the poses may also have ambiguities. In fact, even human can not recognize their correct pose, such as the 'train' in Figure 4.

Conclusion

In this paper, we focus on 3D pose estimation from a single RGB image. A novel network is proposed to learn RGB features and depth features from RGB images by training with

millions of paired synthetic data. In the network, an RGB-to-Depth Embedding method is developed to learn depth features from RGB images effectively. A Synthetic-Real Adaptation module is also integrated into the network to solve the domain gap between synthetic and real data. Experiments show that our method achieves a decent improvement over state-of-the-art methods in all the metrics and superiority of transferred synthetic depth features on the PASCAL 3D+ dataset.

References

Balntas, V.; Doumanoglou, A.; Sahin, C.; Sock, J.; Kouskouridas, R.; and Kim, T.-K. 2017. Pose guided rgbd feature learning for 3d object pose estimation. In *ICCV*, 3856–3864.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. ShapeNet: An information-rich 3d model repository. *https://www.shapenet.org/*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *TPAMI* 32(9):1627–1645.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 580–587.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680.

Grabner, A.; Roth, P. M.; and Lepetit, V. 2018. 3d pose estimation and 3d model retrieval for objects in the wild. In *CVPR*, 3022–3031.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 675–678.

Kao, Y.; Li, W.; Wang, Z.; Zou, D.; He, R.; Wang, Q.; Ahn, M.; and Hong, S. 2018. An appearance-and-structure fusion network for object viewpoint estimation. In *IJCAI*, 4929–4935.

Kehl, W.; Milletari, F.; Tombari, F.; Ilic, S.; and Navab, N. 2016. Deep learning of local RGB-D patches for 3d object detection and 6d pose estimation. In *ECCV*, 205–220.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 1097–1105.

Krull, A.; Brachmann, E.; Michel, F.; Ying Yang, M.; Gumhold, S.; and Rother, C. 2015. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *ICCV*, 954–962.

Li, C.; Bai, J.; and Hager, G. D. 2018. A unified framework for multi-view multi-class object pose estimation. In *ECCV*, 254–269.

Li, C.; Zia, M. Z.; Tran, Q.-H.; Yu, X.; Hager, G. D.; and Chandraker, M. 2017. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *CVPR*, 388–397.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single shot multibox detector. In *ECCV*, 21–37.

Motiian, S.; Piccirilli, M.; Adjeroh, D. A.; and Doretto, G. 2017. Unified deep supervised domain adaptation and generalization. In *ICCV*, 5715–5725.

Mousavian, A.; Anguelov, D.; Flynn, J.; and Kosecka, J. 2017. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 7074–7082.

Pavlakos, G.; Zhou, X.; Chan, A.; Derpanis, K. G.; and Daniilidis, K. 2017. 6-dof object pose from semantic keypoints. In *ICRA*, 2011 – 2018.

Pepik, B.; Stark, M.; Gehler, P.; and Schiele, B. 2012. Teaching 3d geometry to deformable part models. In *CVPR*, 3362–3369.

Poirson, P.; Ammirato, P.; Fu, C.-Y.; Liu, W.; Kosecka, J.; and Berg, A. C. 2016. Fast single shot detection and pose estimation. In *3DV*, 676–684.

Rad, M., and Lepetit, V. 2017. Bb8: a scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 3828–3836.

Rad, M.; Oberweger, M.; and Lepetit, V. 2018. Domain transfer for 3d pose estimation from color images without manual annotations. In *ACCV*, 69–84.

Sahin, C., and Kim, T.-K. 2018. Category-level 6d object pose recovery in depth images. In *ECCV*, 665–681.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Sock, J.; Hamidreza Kasaei, S.; Seabra Lopes, L.; and Kim, T.-K. 2017. Multi-view 6d object pose estimation and camera motion planning using rgbd images. In *ICCV*, 2228–2235.

Su, H.; Qi, C. R.; Li, Y.; and Guibas, L. J. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2686–2694.

Sundermeyer, M.; Marton, Z.-C.; Durner, M.; Brucker, M.; and Triebel, R. 2018. Implicit 3d orientation learning for 6d object detection from RGB images. In *ECCV*, 699–715.

Szeto, R., and Corso, J. J. 2017. Click here: Human-localized keypoints as guidance for viewpoint estimation. In *ICCV*, 1604–1613.

Tulsiani, S., and Malik, J. 2015. Viewpoints and keypoints. In *CVPR*, 1510–1519.

Wang, Z.; Li, W.; Kao, Y.; Zou, D.; Wang, Q.; Ahn, M.; and Hong, S. 2018. HCR-Net: a hybrid of classification and regression network for object pose estimation. In *IJCAI*, 1014–1020.

Wang, L.; Li, Y.; and Lazebnik, S. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*, 5005–5013.

Wu, J.; Xue, T.; Lim, J. J.; Tian, Y.; Tenenbaum, J. B.; Torralba, A.; and Freeman, W. T. 2016. Single image 3d interpreter network. In *ECCV*, 365–382.

Xiang, Y.; Mottaghi, R.; and Savarese, S. 2014. Beyond PASCAL: A benchmark for 3d object detection in the wild. In *WACV*, 75–82.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 3485–3492.