DGGAN: Depth-image Guided Generative Adversarial Networks for Disentangling RGB and Depth Images in 3D Hand Pose Estimation

Liangjian Chen[1], Shih-Yao Lin[2], Yusheng Xie[3], Yen-Yu Lin[4], Wei Fan[2], and Xiaohui Xie[1]

[1]University of California, Irvine , [2]Tencent America , [3]Amazon , [4]National Chiao Tung University ,
{*liangjc2,xhx*}*@ics.uci.edu* , {*shihyaolin,davidwfan*}*@tencent.com* , *yusheng@amazon.com* ,
lin@cs.nctu.edu.tw

Abstract

Estimating 3*D hand poses from RGB images is essential to a wide range of potential applications, but is challenging owing to substantial ambiguity in the inference of depth information from RGB images. State-of-the-art estimators address this problem by regularizing* 3*D hand pose estimation models during training to enforce the consistency between the predicted* 3*D poses and the ground-truth depth maps. However, these estimators rely on both RGB images and the paired depth maps during training. In this study, we propose a conditional generative adversarial network (GAN) model, called Depth-image Guided GAN (DGGAN), to generate realistic depth maps conditioned on the input RGB image, and use the synthesized depth maps to regularize the* 3*D hand pose estimation model, therefore eliminating the need for ground-truth depth maps. Experimental results on multiple benchmark datasets show that the synthesized depth maps produced by DGGAN are quite effective in regularizing the pose estimation model, yielding new state-of-the-art results in estimation accuracy, notably reducing the mean* 3*D end-point errors (EPE) by* 4.7%*,* 16.5%*, and* 6.8% *on the RHD, STB and MHP datasets, respectively.*

1. Introduction

Vision-based 3D hand pose estimation (3D HPE) aims to estimate the 3D keypoint coordinates of a given hand image. 3D HPE has drawn increasing attention owing to its wide applications to human-computer interaction (HCI) [1, 21], sign language understanding [34], augmented/virtual reality (AR/VR) [22, 15], and robotics [1]. RGB images and depth maps are two the most commonly used input data for the 3D HPE task. An example of a hand image and its corresponding depth map is shown in Figure 1(a). Depth map can provide 3D information related to the distance of the surface of human hands. Training networks with depth maps has been proven to achieve significant progress on the 3D HPE task [4, 16]. In addition, with the depth information provided by the depth maps, the hand segmentation task can be effectively solved. Unfortunately, capturing depth maps often requires specific sensors (*e.g*. Microsoft Kinect, RealSense), which limits the usability of those state-of-the-art methods based on depth maps. Commercial depth sensors are usually much more expensive than RGB cameras. On the other hand, RGB images are the most commonly used input data in the HPE task because it can be easily captured by abundant low-cost optical sensors such as webcams and smartphones. However, 3D HPE from RGB images is a challenging task.

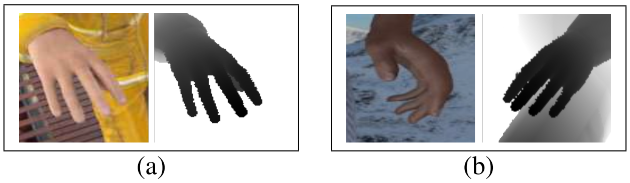


Figure 1. Training examples in a generic 3D HPE dataset: (a) paired RGB and depth images; (b) unpaired RGB and depth images. Our work does not rely on paired training data and therefore is applicable to both RGB-only and depth-only 3D HPE tasks.

In the absence of depth information, estimating 3D hand pose from a monocular RGB image is intrinsically an ill-posed problem. To address this issue, the state-of-the-art methods such as [4, 10] leverage both RGB hand images and their paired depth maps for the 3D HPE task. Their 3D hand pose inference process takes an RGB image and the paired depth information into account. They first regress 3D hand poses on RGB images, and then utilize a separate branch to regularize the predicted 3D hand pose by using the paired depth maps. The objective of the depth regularizer is to make the predicted 3D keypoint positions consistent with the provided depth map. It results in two major advantages: 1) training networks with depth maps can efficiently improve the hand pose estimator by using the depth information to reduce the ambiguity and 2) enabling 3D HPE based on merely RGB images during the inference stage. These approaches require paired RGB and depth training images. Unfortunately, most existing hand pose datasets only contain either depth maps or RGB images, instead of both. It makes the aforementioned approaches not applicable to such datasets. Besides, the unpaired RGB and depth training images cannot be explorited for them. Figure 1(b) shows an example of unpaired RGB and depth map images.

To tackle this problem, we propose a novel generative adversarial networks, called *Depth-image Guided GAN* (DGGAN). Our network contains two modules: *depth-map reconstruction* and *hand pose estimation*. The main idea of our approach is to directly reconstruct the depth map from an input RGB hand image in the absence of paired RGB and depth training images. Given an RGB image, our depth-map reconstruction module aims to infer its depth map. Our hand pose estimation module takes RGB and depth information into account to infer the 3D hand pose. In the hand pose estimation module, we infer the 2D hand keypoints on the input RGB image, and regress the 3D hand pose by using the inferred 2D keypoints. The depth map is then used to regularize the inferred 3D hand pose. Unlike most existing 3D HPE models, the real depth maps used to train our DGGAN model do not require any paired RGB images. Once DGGAN is learned, the proposed HPE module directly infers the hand pose by using an RGB image and guided (regularized) by a DGGAN-inferred depth map. Since the depth-map can be inferred by our depth-map reconstruction module, the proposed DGGAN no longer requires paired RGB and depth images. Our DGGAN jointly trains the two modules in an end-to-end trainable network architecture. Experimental results on multiple benchmark datasets demonstrate that our DGGAN not only reconstructs the depth map of an input RGB image, but also significantly improves the 3D hand pose estimator via an additional depth regularizer.

The main contributions of this study are summarized as follows:

1. We propose a depth-map guided adversarial neural networks (DGGAN) for 3D hand pose estimation from RGB images. Our network can jointly infer the depth information from input RGB images and estimate the 3D hand poses.

2. We introduce a depth-map reconstruction module to infer the depth maps from input RGB images while learning to predict 3D hand poses. Our DGGAN is trained on readily accessible hand depth maps that are not paired with RGB images.

3. Experimental results demonstrate that our approach achieves new state-of-the-art in 3D hand pose prediction accuracy on three benchmark datasets, including the RHD, STB, and MHP datasets.

2. Related Work

Research topics related to this work are discussed below.

2.1. 3D HPE from Depth Images

3D HPE from depth mapshas been extensively studied. Existing approaches in this field make noticeable advances [29, 33, 8, 31, 9, 11, 20]. Wan *et al.* [29] propose a dense regression approach to fit the parameters of a deformed hand model. Ge *et al.* [9, 11] present PointNet[24] to extract hand features and regress hand joint locations by referring to the extracted features. Wu *et al.* [31] adopt the intermediate dense guidance map supervision to generate hand heatmaps. Although the existing methods achieve very accurate estimation results, they typically rely on the hand data captured by high-precision depth sensors, which are still expensive to have in practice and usually require data collection in a lab environment. Different from the models in the aforementioned methods, our model performs inference on RGB data without the need of depth maps.

2.2. 3D HPE from Monocular RGB Images

Due to the wide availability of RGB cameras, 3D HPE from monocular RGB images is becoming increasingly popular in computer vision applications. Many recent methods aim at estimating hand joint locations directly from a single RGB image [4, 16, 10, 37, 22, 6, 32, 3, 28]. Zimmermann *et al.* [37] use 2D convolutional neural networks (CNN) to extract features from an RGB image, and regress the 3D hand joint locations. However, their method suffers from depth ambiguity due to the absence of depth information. Developing the methods upon the work by Zimmermann *et al.*, Iqbal *et al.* [16] and Cai *et al.* [4] inherit and adopt a similar 2D CNN architecture for extracting image features. Iqbal *et al.* use depth maps as intermediate guidance while Cai *et al.* treat depth maps as a regularizer in a weakly supervised manner. Though these two methods make substantial progress in terms of estimation accuracy, there currently exist few datasets that fulfill their requirement of paired depth maps and RGB images. Ge *et al.* [10] take one step further by predicting the hand mesh from an RGB image and then the 3D hand joint locations based on the mesh. However, their method requires paired mesh information which is even rarer among all existing datasets.

Compared with these methods, our method also uses depth information during training, but it does not require any paired RGB images and depth maps. Thus, it is much more flexible since it can consume RGB images and depth maps from different datasets or sources.

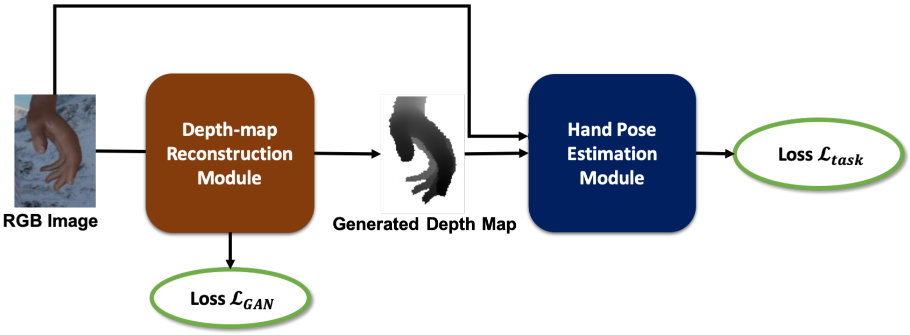

RGB Image
Depth-map Reconstruction Module
Generated Depth Map
Hand Pose Estimation Module
Loss $\mathcal{L}_{task}$
Loss $\mathcal{L}_{GAN}$

Figure 2. Overview of the proposed DGGAN. DGGAN consists of two modules, a *depth-map reconstruction module* shown in Figure 4 and a *hand pose estimation module* shown in Figure 4. The former module trained using the GAN loss aims at inferring the depth map of a hand based on the input RGB image and making the generated depth map looks realistic. The latter module trained using the task loss estimates hand poses from the input RGB and the GAN-reconstructed depth images.

2.3. 3D Mesh Estimation from RGB Images

To further enhance 3D HPE [2, 3, 10, 18], hand mesh estimation can be included. Namely, the model estimates not only the hand joints but also the hand surface mesh. However these methods such as [10] have a common drawback: They require additional mesh annotations which are even more expensive to obtain than joint locations. Thus, they are typically trained on synthetic datasets due to this limitation. Seungryul *et al.* [3] introduce an iterative learning method to refine mesh shapes and achieve very good performance. However, like 3D hand joint locations, hand meshes highly rely on additional supervision from hand segment maps which are typically not available in nowadays hand pose datasets. The method by boukhayma *et al.* [3] is the only extra-data-free method, but its performance is limited.

2.4. GAN-based Image Translation

Generating images using generative adversarial networks (GAN) [13] has gained remarkable progress. Many approaches explore how to better manipulate images by applying GAN models [14, 17, 36, 7]. Isola *et al.* [17] propose the Pix2Pix network which translates label or edges maps to synthesized photos, reconstructs objects from edge maps, or colorizes images. Zhu *et al.* [36] introduce the cycle-consistent generated adversarial network (CycleGAN). CycleGAN uses the cycle consistency loss to disentangle the input and output pair and therefore does not need paired input. Hoffman *et al.* [14] propose cycle-consistent adversarial domain adaptation (CyCADA). Compared to CycleGAN, CyCADA contains a segmentation loss. As a result, CyCADA not only translates images from one modality to another but also deals with a specific visual task.

Applying the generative adversarial model to RGB hand images for hand pose estimation is also gaining popularity. Muller *et al.* [22] introduce the geometry consistent GAN (GeoConGAN) to generate synthetic image data for training. Chen *et al.* [6] propose the tonality-alignment generative adversarial networks (TAGAN) for producing more realistic images from synthetic images for hand pose estimator training. However, these methods only focus on generating RGB images. None of them generates depth maps for assisting hand pose estimator training.

3. Our Approach

Our goal is to estimate the 3D hand pose from a monocular RGB hand image. Although the existing state-of-the-art methods [3, 25, 33] have shown that training networks with RGB and depth images can improve the 3D hand pose estimators, few 3D hand pose datasets consist of paired RGB and depth images. To deal with the lack of paired data issue, we propose a novel adversarial neural network, called depth-map guided generated adversarial networks (DGGAN) illustrated in Figure 2, which can jointly learn to infer the depth map from an RGB image of hand and to estimate 3D hand pose. In the following, we give an overview of the proposed DGGAN and describe the two major modules of DGGAN in detail.

3.1. Overview of DGGAN

The proposed DGGAN consists of two major modules, a *depth-map reconstruction* module and a *hand pose estimation* module. Its network architecture is shown in Figure 2.

Given an RGB hand image $\mathbf{I}$, we want to estimate the K 3D hand joint locations $\mathbf{J}^{xyz} \in \mathbb{R}^{3\times K}$. Each column in the $3\times K$ matrix is a vector of size 3 and represents the (x, y, z) coordinates of a joint, i.e., $\mathbf{J}^{xyz} = [J_1^{xyz}, J_2^{xyz}, \ldots, J_K^{xyz}]$.

The two modules in the proposed DGGAN G are trained by using the GAN loss $\mathcal{L}_{GAN}$ and the task loss $\mathcal{L}_{task}$, respectively. The objective of learning G is formulated as a min-max game:

$$G^* = \arg\min_G \max_D (\lambda_t \mathcal{L}_{task} + \lambda_g \mathcal{L}_{GAN}), \qquad (1)$$

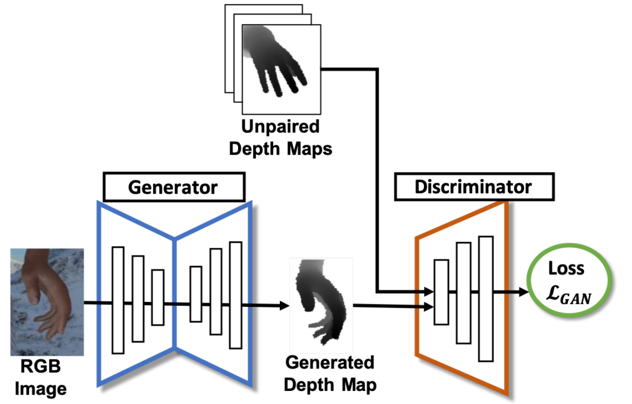


Figure 3. Network architecture of the depth-map reconstruction module.

where λ_t and λ_g control the relative importance of these two loss terms.

Given an RGB hand image, our depth-map reconstruction module tries to generate its corresponding depth map. A set of unpaired training depth images is adopted to train the depth-map reconstruction module so that its inferred depth maps are similar to real ones. To achieve that, the discriminator in this module works on distinguishing real depth maps from fake (generated) ones. Section 3.2 describes the details of depth-map reconstruction. The depth map inferred from the depth-map reconstruction module together with the input RGB image are fed to the hand pose estimation module for estimating the 3D hand pose. In the hand pose estimation module, the input RGB image is used to regress the 3D hand pose. The inferred depth-map is adopted to regularize the predicted 3D hand pose. The loss for hand pose estimation $\mathcal{L}_{task}$ is adopted for optimization. Section 3.3 describes the details.

3.2. Depth-map Reconstruction Module

The depth-map reconstruction module aims at relaxing the requirement of paired RGB and depth images during training. This module is constructed via an adversarial network that infers the depth map according to an input RGB image. Figure 3 shows the network architecture of this module. In the training phase, our network requires both depth and RGB training images. Nevertheless, the RGB and depth images do not need to be paired. We consider the process of inferring depth map from its corresponding RGB image as an unsupervised adaptation problem, where the RGB modality S and depth modality T are both provided. We are given a set of RGB images X_S and a set of real depth maps X_T. To translate from S to T, we adopt an encoder-decoder architecture $G_{S\to T}$. The generator $G_{S\to T}$ is trained to generate a realistic depth map to fool the discriminator D while D id derived to distinguish the real data x_t and generated fake data $G_{S\to T}(x_s)$. The loss for the depth-reconstruction modules is as follows:

$$\begin{aligned}\mathcal{L}_{GAN}(G_{S\to T}, D, X_S, X_T) = \\ \mathbb{E}_{x_t\sim X_T}[\log D(x_t)] + \\ \mathbb{E}_{x_s\sim X_S}[\log(1 - D(G_{S\to T}(x_s)))].\end{aligned} \tag{2}$$

This loss also provides semantic constraints to force the generator to produce more realistic depth maps. By taking as input unpaired RGB and depth images, our depth-map reconstruction module becomes applicable to vastly more hand pose datasets. Furthermore, we can train the network with a large amount of unpaired RGB and depth images.

3.3. Hand Pose Estimation Module

Given an inferred depth map computed by the depth-map reconstruction module, we combine it with the input RGB image and feed both to the hand pose estimation module. The network architecture of the hand pose estimation module is shown in Figure 4. The hand pose estimation module calculates the task loss $\mathcal{L}_{task}$, which is composed of two terms $\mathcal{L}_{task} = \mathcal{L}_{2D} + \mathcal{L}_z$. The 3D hand regression loss $\mathcal{L}_{2D}$ and depth regularization loss $\mathcal{L}_z$ are described in section 3.3.1 and 3.3.2, respectively.

3.3.1 3D Hand Pose Regression

Previous studies [4] show that depth information can be used to build a powerful regularizer. We leverage the depth regularizer for improving the result of 3D HPE. Unlike most previous works where the ground-truth depth maps are needed, our model uses a synthetic depth map generated by the depth-map reconstruction module. Our experimental results show that training with such synthetic depth maps substantially helps improve the result of direct regression.

3D hand pose regression takes an RGB image and an inferred depth map as input and outputs joint locations in two steps. In the first step, we adopt a popular variant of the CPM architecture [5, 30] as the 2D joint location predictor. This predictor consists of six stages. Each stage contains seven convolutional layers followed by a Rectified Linear Unit (ReLu). It predicts K heatmaps $\{H_s^k\}_{k=1}^K$ for K different hand joints. The pixel value in k^{th} heatmap at stage s, H_s^k, indicates the confidence that the k^{th} joint is located at this position. Following the convention [30], the ground-truth heatmap is denoted as $\{H_*^k\}_{k=1}^K$. Each H_*^k is the Gaussian blur of the Dirac-δ distribution centered at the ground-truth location of k^{th} joint. We train this part of Hand Pose module by standard backpropogation and the mean square error (MSE) loss. In addition to the MSE loss, we add the intermediate supervision for each stage. The final loss for 2D location prediction is

$$\mathcal{L}_{2D} = \frac{1}{6K}\sum_{s=1}^{6}\sum_{k=1}^{K}||H_s^k - H_*^k||_F^2. \tag{3}$$

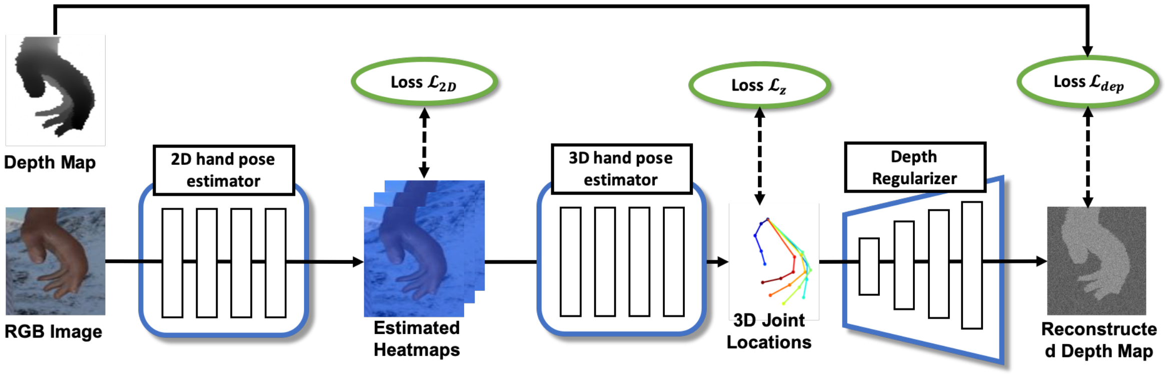


Figure 4. Architecture of the hand pose estimation module. This module takes paired RGB images and inferred depth maps as inputs. 2D CPM consumes an RGB image as input and produces the hand joint heatmap. The joint heatmap is fed to the regression network to estimate the 3D joint locations with the aid of a depth regularizer. The depth regularizer reconstructs the depth map from 3D joint locations and is trained using L1 loss and the GAN-synthesized depth map as guidance.

In the second step, the regression network takes the heatmap from CPM as input, and outputs the relative depth. Its architecture is a mini-CPM (one stage instead of six) followed by three fully connected layers. $Z \in \mathbb{R}^{K\times 1}$ denotes the relative depth of each hand joint. We employ smooth L1 loss between Z and the ground-truth Z^*. The loss of depth regression $\mathcal{L}_z$ is summarized as follows:

$$\mathcal{L}_z = \frac{1}{K}\sum_{k=1}^{K}\begin{cases}\frac{1}{2}(Z_k - Z_k^*)^2, \text{ if } |Z_k - Z_k^*| \le 0.5\\ |Z_k - Z_k^*|, \text{ otherwise.}\end{cases} \tag{4}$$

3.3.2 Depth Regularizer

To provide supervision on every pixel on a depth map, we employ the depth regularizer (DR) proposed in [4]. The depth regularizer takes the relative depth as input and predicts a relative depth map D. It reshapes $Z \in \mathbb{R}^{K\times 1}$ to a $K \times 1 \times 1$ tensor, which is considered as a K-channel image input. We then up-sample this image from K-channel with resolution 1×1 to 1-channel with the original depth map resolution $(n \times m)$ through the 6 layers of transposed CNN.

We take L1 norm between D and the ground-truth relative depth map D^* as depth regularizer loss $\mathcal{L}_{dep}$, i.e.,

$$\mathcal{L}_{dep} = ||D - \hat{D}^*||, \tag{5}$$

where D^* is obtained by input depth map $\hat{D}^*$ as follows

$$D^* = \frac{\hat{D}^* - \hat{D}^*}{\max \hat{D}^* - \min \hat{D}^*}. \tag{6}$$

Note that, we only use the ground-truth depth map $\hat{D}^*$ during the initialization stage. It would be replaced by DGGAN-generated depth maps once the initialization stage ends.

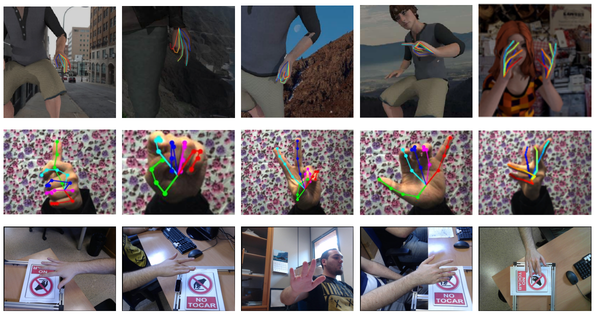


Figure 5. Some examples of the three benchmark datasets used for evaluation. **Top Row:** The RHD dataset [37] provides synthetic hand images with 3D hand keypoint annotations. **Middle Row:** The STB dataset [35] contains real hand images with 3D keypoints. **Bottom Row:** The MHP [12] offers real hand images with 3D keypoints.

Combining the loss terms described in Section 3.3 and Section 3.3.2, we summarize the loss function for the hand pose estimation module as

$$\mathcal{L}_{task} = \lambda_z * \mathcal{L}_z + \lambda_{2D} * \mathcal{L}_{2D} + \lambda_{dep} * \mathcal{L}_{dep}, \tag{7}$$

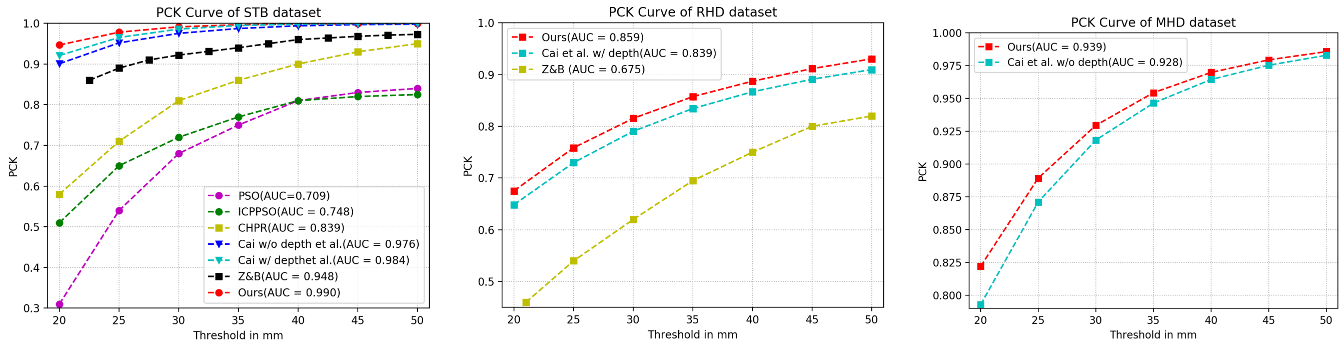
where λ_z, λ_{2D}, λ_{dep} control the importance of three different loss terms, respectively.

4. Experimental Settings

This section introduces our experimental settings. The selected benchmark datasets for performance evaluation are first given. The evaluation metric and training details are then presented.

4.1. Datasets for Evaluation

We conduct the experiments on three benchmark datasets, including the stereo hand tracking benchmark



(a) (b) (c)

Figure 6. Comparisons with the state-of-the-art approaches on the (a) STB, (b) RHD, and (c) MHP datasets for 3D hand pose estimation.

(STB) [35], the render hand pose dataset (RHD) [37], and the multi-view hand pose (MHP) dataset[12].

The STB dataset is a dataset of real hands. It contains two different subsets called SK and BB. The images in SK are captured by Point Grey Bumblebee2 stereo camera while images in BB are from a depth sensor. In our experiments, we use the BB subset for DGGAN training, and leverage the SK subset for unpaired testing.

RHD is a synthetic dataset. Zhang *et al.* [35] use a 3D simulator, Maya, to render the images from 20 different characters doing 39 actions. Each data entry consists of an RGB image and the corresponding depth image, and both 2D/3D annotations. This dataset is challenging since its images are captured with various view points and of many different hand shapes.

The MHP dataset provides color hand images as well as the bounding boxes of hands and the 2D and 3D location of each joint. It consists of hand imaegs of 21 people with different hand movements. For each frame, it provides the images from four different angles of view. The 2D and 3D annotations are obtained by Leap Motion Controller.

Before training, we first crop the hand regions from the original canvas to make sure that hand parts have dominating proportion in the frame. Notice that the STB and MHP datasets use the center of a palm rather than a wrist as one of its hand keypoints. Hence, we revise the annotation to move the center of the palm to the wrist in the same way performed in [4].

4.2. Evaluation Metric

Following the previous works [4, 6, 37], we evaluate the results of hand pose estimation by using 1) the *area under the curve* (AUC) on *percentage of correct keypoints* (PCK) between threshold 20mm and 50mm (AUC 20_50) and 2) the *end-point-error* (EPE): the distance between predicted 3D joint locations and the ground truth. In Table 1, we report the AUC 20_50 as well as the mean and the median of EPE over all hand keypoints.

4.3. Training

During training, we first initialize the weights of the depth-map reconstruction and hand pose estimation modules in the proposed DGGAN. Both modules are initialized by fitting the STB dataset (see 4.1) but trained separately. Then, we connect the two modules and fine-tune the whole network in an end-to-end manner. For training with the RHD and STB dataset, the discriminator is derived to distinguish the $G_{S\rightarrow T}(x_s)$ and x_t, a randomly chosen depth-map from the respective dataset. For the MHP dataset, we simply randomly assign a depth-map from RHD dataset as x_t because the MHP dataset does not contain any dense depth maps.

5. Experimental Results

For evaluation on the STB dataset, we choose PSO [19], ICPPSO [25], and CHPR [27] as the baselines. In addition, we select the state-of-the-art approaches, Z&B [37] and that by Cai *et al.* [4] for comparison.

On the RHD dataset, we compare our method with Z&B [37] and that in [4]. Also, on the MHP dataset, we compare our method to that in [4]. Note that Cai *et al.* [4] have not released their code yet. We re-implement their method and report the results according to our implementation.

5.1. Ablation Study

For analyzing the effectiveness of the proposed DGGAN, we conduct ablation studies for DGGAN on three different datasets. The detailed results are summarized in Table 1. Specifically, we conduct the experiments for the following three different settings:

1. Regression: It represents training the regression network only on RGB images and without any depth regularizer.

2. Regression + DR + DGGAN: We learne the depth-regularized regression network using RGB images

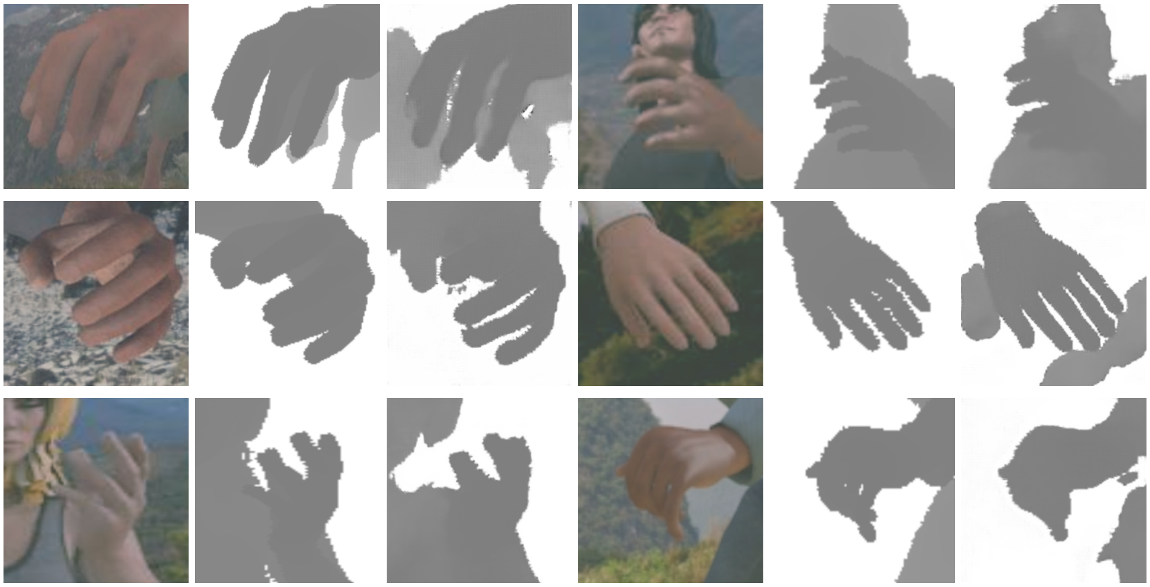


Figure 7. Comparison between the generated and ground-truth depth maps on the RHD dataset. The first and fourth columns show the RGB images. The second and fifth columns display the real depth maps. The third and sixth columns give the generated depth maps.

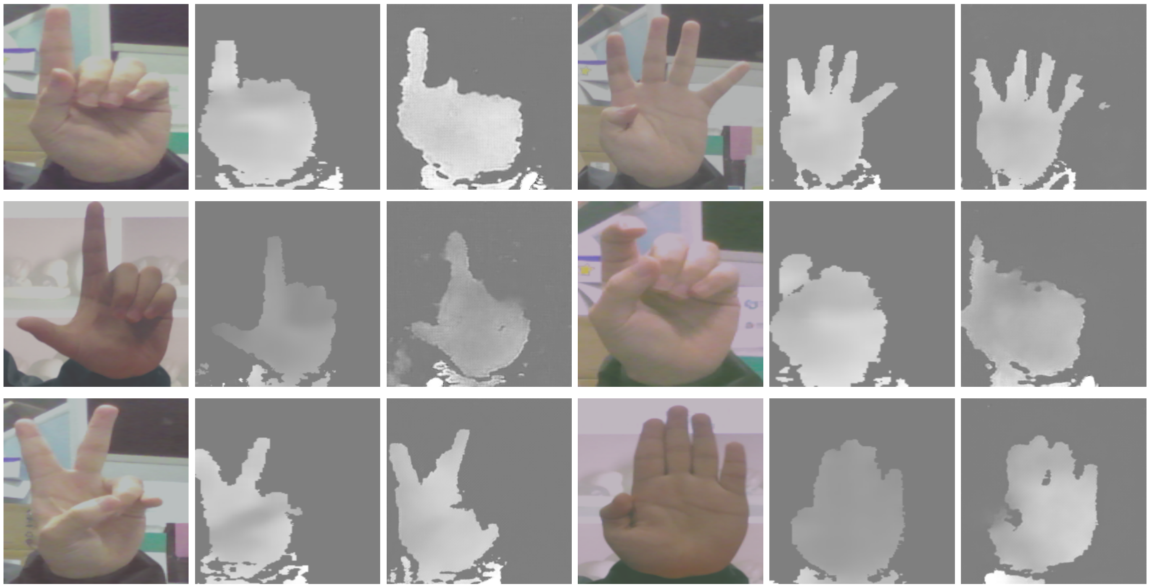


Figure 8. Comparison between the generated and ground-truth depth maps on the STB dataset. The first and fourth columns show the RGB images. The second and fifth columns display the real depth maps. The third and sixth columns give the generated depth maps.

with the depth maps generated by DGGAN.

3. Regression + DR + true depth map: We derive the depth-regularized regression network using RGB images with their paired true depth maps.

To measure the effectiveness of the generated depth maps, we compare settings Regression and Regression + DR + DGGAN. As illustrated in Table 1, using the generated depth map significantly boosts the performance of the model in Regression. The AUC 20_50 is improved by **0.043**, **0.024**, **0.011** on the RHD, STB, and MHP datasets, respectively. The EPE mean is also considerable reduced by **13.2%** and **19.7%** and **7.3%** on the RHD, STB and MHP datasets respectively.

To compare the generated depth map with the real depth maps, we conduct two more experiments. Comparing results of Regression + DR + true depth map and Regression + DR + DGGAN shows that the generated depth maps are

Table 1. 3D pose estimation results on the RHD, STB, MHP datasets. ↑: higher is better. ↓: lower is better. *Regression* is the previous State-of-the-art without using paired depth maps.

	AUC 20-50 ↑	EPE mean (mm) ↓	EPE median (mm) ↓
RHD Dataset			
Regression	0.816	21.5	13.96
Regression + DR + DGGAN	**0.839**	**19.0**	**13.17**
Regression + DR + true depth map	0.859	18.0	13.16
STB Dataset			
Regression	0.976	10.91	9.11
Regression + DR + DGGAN	**0.990**	**9.11**	**7.70**
Regression + DR + true depth map	0.984	10.05	8.44
MHP Dataset			
Regression	0.928	14.08	10.75
Regression + DGGAN	**0.939**	**13.12**	**9.91**

Table 2. EPE mean comparison on the STB dataset between our approach and the method by Boukhayma *et al.* [3]

Method	EPE mean (mm) ↓
Regression + DR + DGGAN (Ours)	**9.11**
Boukhayma *et al.* [3]	9.76

a key factor of performance boosting. On the RHD dataset, training with the generated depth maps is only slightly worse than the true RHD depth maps by 0.02 in AUC 20_50 and 1 mm in EPE mean. However, on the STB dataset, the results of training with generated depth maps are even better than training with the real depth maps (by 0.006 in AUC 20_50 and 0.94 mms in EPE mean). This result is probable due to the fact that the depth maps collected from depth sensors are less stable and noisier than the depth maps collected from a 3D simulator. By training the DGGAN with unpaired high-quality depth maps from RHD, our generator can potentially reduce the noise, and further benefit the training in the hand pose estimation module. It is worth noting that Regression + DR + true depth map requires the paired depth and RGB image.

In addition to the quantitative analysis, Figure 7 and Figure 8 provide some examples for visual comparison between the generated and true depth maps on the RHD and STB datasets, respectively. We can see that the generated depth maps are visually very similar to the ground-truth ones.

5.2. Comparison with State-of-the-arts

We select the state-of-the-art approaches [3, 4, 23, 26, 35, 22, 37] for comparison. The comparison results are reported in Figure 6 and Table 2. As shown in Figure 6 and Table 2, our approach outperforms all existing state-of-the-art methods. Although the results of the method by Cai *et al.* [4] come close to ours, we emphasize that our DGGAN has an crucial advantage of *not* requiring any paired RGB and depth images.

6. Conclusion

The lack of large-scale datasets of paired RGB and depth images is one of the major bottlenecks for improving 3D hand pose estimation. To address this limitation, we propose a conditional GAN-based model called DGGAN to bridge the gap between RGB images and depth maps. DGGAN synthesizes depth maps from RGB images to regularize the 3D hand pose prediction model during training, eliminating the need of paired RGB images and depth maps conventionally used to train such models.

The proposed DGGAN is integrated into a 3D hand pose prediction framework, and is trained end-to-end together for 3D pose estimation. DGGAN not only generates more realistic hand depth images, which can be used in many other applications such as 3D shape estimation but also results in significant improvement in 3D hand pose estimation, achieving new state-of-the-art results.

Acknowledgement. This work was supported in part by Ministry of Science and Technology (MOST) under grants MOST 107-2628-E-001-005-MY3 and MOST 108-2634-F-007-009.

References

[1] S. Antoshchuk, M. Kovalenko, and J. Sieck. Gesture recognition-based human–computer interaction interface for multimedia applications. In *Digitisation of Culture: Namibian and International Perspectives*. 2018.

[2] S. Baek, K. I. Kim, and T.-K. Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR*, 2019.

[3] A. Boukhayma, R. d. Bem, and P. H. Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019.

[4] Y. Cai, L. Ge, J. Cai, and J. Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[6] L. Chen, S.-Y. Lin, Y. Xie, H. Tang, Y. Xue, Y.-Y. Lin, X. Xie, and W. Fan. Tagan: Tonality-alignment generative adversarial networks for realistic hand pose synthesis. In *BMVC*, 2019.

[7] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang. Cr-DoCo: Pixel-level domain transfer with cross-domain consistency. In *CVPR*, 2019.

[8] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*, 2017.

[9] L. Ge, Y. Cai, J. Weng, and J. Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *CVPR*, 2018.

[10] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019.

[11] L. Ge, Z. Ren, and J. Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*, September 2018.

[12] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. Large-scale multiview 3d hand pose dataset. *arXiv preprint arXiv:1707.03742*, 2017.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[14] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[15] Y.-P. Hung and S.-Y. Lin. Re-anchorable virtual panel in three-dimensional space, 2016. US Patent 9,529,446.

[16] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, 2018.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[18] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018.

[19] J. Kennedy. Particle swarm optimization. *Encyclopedia of machine learning*, 2010.

[20] S. Li and D. Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. *arXiv preprint arXiv:1812.02050*, 2018.

[21] S.-Y. Lin, C.-K. Shie, S.-C. Chen, and Y.-P. Hung. Airtouch panel: a re-anchorable virtual touch panel. In *MM*, 2013.

[22] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018.

[23] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *WACV*, 2018.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[25] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *CVPR*, 2014.

[26] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, June 2018.

[27] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *CVPR*, 2015.

[28] B. Tekin, F. Bogo, and M. Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. *CVPR*, 2019.

[29] C. Wan, T. Probst, L. Van Gool, and A. Yao. Dense 3d regression for hand pose estimation. In *CVPR*, 2018.

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[31] X. Wu, D. Finnegan, E. O'Neill, and Y.-L. Yang. Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In *ECCV*, September 2018.

[32] L. Yang and A. Yao. Disentangling latent hands for image synthesis and pose estimation. *CVPR*, 2019.

[33] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Yong Chang, K. Mu Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, J. Yuan, X. Chen, G. Wang, F. Yang, K. Akiyama, Y. Wu, Q. Wan, M. Madadi, S. Escalera, S. Li, D. Lee, I. Oikonomidis, A. Argyros, and T.-K. Kim. Depth-based 3d hand pose estimation: From current achievements to future goals. In *CVPR*, 2018.

[34] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *ICMI*, 2011.

[35] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[37] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *CVPR*, 2017.