This paper has been submitted for publication on November 15, 2016.

Learning from Simulated and Unsupervised Images through Adversarial Training

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb
Apple Inc.
{a_shrivastava, tpf, otuzel, jsusskind, wenda_wang, rwebb}@apple.com

arXiv:1612.07828v1 [cs.CV] 22 Dec 2016

Abstract

With recent progress in graphics, it has become more tractable to train models on synthetic images, potentially avoiding the need for expensive annotations. However, learning from synthetic images may not achieve the desired performance due to a gap between synthetic and real image distributions. To reduce this gap, we propose Simulated+Unsupervised (S+U) learning, where the task is to learn a model to improve the realism of a simulator's output using unlabeled real data, while preserving the annotation information from the simulator. We develop a method for S+U learning that uses an adversarial network similar to Generative Adversarial Networks (GANs), but with synthetic images as inputs instead of random vectors. We make several key modifications to the standard GAN algorithm to preserve annotations, avoid artifacts and stabilize training: (i) a 'self-regularization' term, (ii) a local adversarial loss, and (iii) updating the discriminator using a history of refined images. We show that this enables generation of highly realistic images, which we demonstrate both qualitatively and with a user study. We quantitatively evaluate the generated images by training models for gaze estimation and hand pose estimation. We show a significant improvement over using synthetic images, and achieve state-of-the-art results on the MPIIGaze dataset without any labeled real data.

1. Introduction

Large labeled training datasets are becoming increasingly important with the recent rise in high capacity deep neural networks [4, 18, 44, 44, 1, 15]. However, labeling such large datasets is expensive and time-consuming. Thus the idea of training on synthetic instead of real images has become appealing because the annotations are automatically available. Human pose estimation with Kinect [32] and, more recently, a plethora of other tasks have been tackled using synthetic data [40, 39, 26, 31].

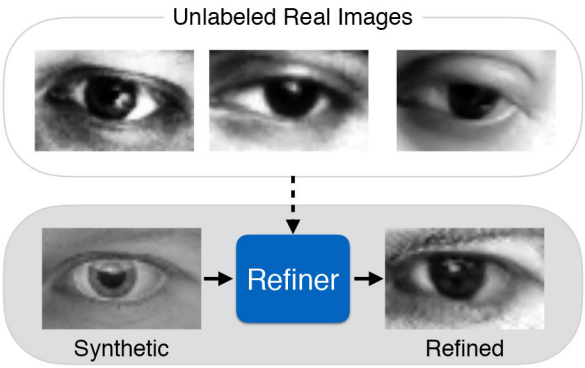

Figure 1. Simulated+Unsupervised (S+U) learning. The task is to learn a model that improves the realism of synthetic images from a simulator using unlabeled real data, while preserving the annotation information.

However, learning from synthetic images can be problematic due to a gap between synthetic and real image distributions – synthetic data is often not realistic enough, leading the network to learn details only present in synthetic images and fail to generalize well on real images. One solution to closing this gap is to improve the simulator. However, increasing the realism is often computationally expensive, the renderer design takes a lot of hard work, and even top renderers may still fail to model all the characteristics of real images. This lack of realism may cause models to overfit to 'unrealistic' details in the synthetic images.

In this paper, we propose Simulated+Unsupervised (S+U) learning, where the goal is to improve the realism of synthetic images from a simulator using unlabeled real data. The improved realism enables the training of better machine learning models on large datasets without any data collection or human annotation effort. In addition to adding realism, S+U learning should preserve annotation information for training of machine learning models – *e.g.* the gaze direction in Figure 1 should be preserved. Moreover, since machine learning models can be sensitive to artifacts in the synthetic data, S+U learning should generate images without artifacts.

We develop a method for S+U learning, which we term SimGAN, that refines synthetic images from a simulator using a neural network which we call the 'refiner network'. Figure 2 gives an overview of our method: a synthetic image is generated with a black box simulator and is refined using the refiner network. To add realism – the first requirement of an S+U learning algorithm – we train our refiner network using an adversarial loss, similar to Generative Adversarial Networks (GANs) [7], such that the refined images are indistinguishable from real ones using a discriminative network. Second, to preserve the annotations of synthetic images, we complement the adversarial loss with a self-regularization loss that penalizes large changes between the synthetic and refined images. Moreover, we propose to use a fully convolutional neural network that operates on a pixel level and preserves the global structure, rather than holistically modifying the image content as in *e.g.* a fully connected encoder network. Third, the GAN framework requires training two neural networks with competing goals, which is known to be unstable and tends to introduce artifacts [29]. To avoid drifting and introducing spurious artifacts while attempting to fool a single stronger discriminator, we limit the discriminator's receptive field to local regions instead of the whole image, resulting in multiple local adversarial losses per image. Moreover, we introduce a method for improving the stability of training by updating the discriminator using a history of refined images rather than the ones from the current refiner network.

Contributions:

1. We propose S+U learning that uses unlabeled real data to refine the synthetic images generated by a simulator.

2. We train a refiner network to add realism to synthetic images using a combination of an adversarial loss and a self-regularization loss.

3. We make several key modifications to the GAN training framework to stabilize training and prevent the refiner network from producing artifacts.

4. We present qualitative, quantitative, and user study experiments showing that the proposed framework significantly improves the realism of the simulator output. We achieve state-of-the-art results, without any human annotation effort, by training deep neural networks on the refined output images.

1.1. Related Work

The GAN framework learns two networks (a generator and a discriminator) with competing losses. The goal of the generator network is to map a random vector to a realistic image, whereas the goal of the discriminator is to distinguish the generated and the real images. The GAN framework was first introduced by Goodfellow *et al.* [7] to generate visually realistic images and, since then, many improvements and interesting applications have been proposed [29]. Wang and Gupta [38] use a Structured GAN to learn surface normals and then combine it with a Style GAN to generate natural indoor scenes. Im *et al.* [12] propose a recurrent generative model trained using adversarial training. The recently proposed iGAN [45] enables users to change the image interactively on a natural image manifold. CoGAN by Liu *et al.* [19] uses coupled GANs to learn a joint distribution over images from multiple modalities without requiring tuples of corresponding images, achieving this by a weight-sharing constraint that favors the joint distribution solution. Chen *et al.* [2] propose InfoGAN, an information-theoretic extension of GAN, that allows learning of meaningful representations. Tuzel *et al.* [36] tackled image superresolution for face images with GANs. Li and Wand [17] propose a Markovian GAN for efficient texture synthesis. Lotter *et al.* [20] use adversarial loss in an LSTM network for visual sequence prediction. Yu *et al.* [41] propose the SeqGAN framework that uses GANs for reinforcement learning. Many recent works have explored related problems in the domain of generative models, such as PixelRNN [37] that predicts pixels sequentially with an RNN with a softmax loss. The generative networks focus on generating images using a random noise vector; thus, in contrast to our method, the generated images do not have any annota-

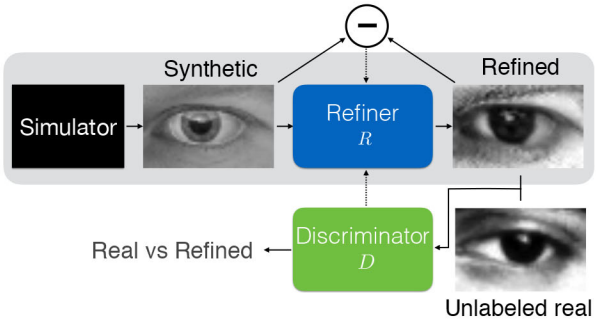

Figure 2. Overview of SimGAN. We refine the output of the simulator with a refiner neural network, R, that minimizes the combination of a local adversarial loss and a 'self-regularization' term. The adversarial loss fools a discriminator network, D, that classifies an image as real or refined. The self-regularization term minimizes the image difference between the synthetic and the refined images. This preserves the annotation information (*e.g.* gaze direction), making the refined images useful for training a machine learning model. The refiner network R and the discriminator network D are updated alternately.

tion information that can be used for training a machine learning model.

Many efforts have explored using synthetic data for various prediction tasks, including gaze estimation [40], text detection and classification in RGB images [8, 14], font recognition [39], object detection [9, 24], hand pose estimation in depth images [35, 34], scene recognition in RGB-D [10], semantic segmentation of urban scenes [28], and human pose estimation [23, 3, 16, 13, 25, 27]. Gaidon *et al*. [5] show that pre-training a deep neural network on synthetic data leads to improved performance. Our work is complementary to these approaches, where we improve the realism of the simulator using unlabeled real data.

Ganin and Lempitsky [6] use synthetic data in a domain adaptation setting where the learned features are invariant to the domain shift between synthetic and real images. Wang *et al*. [39] train a Stacked Convolutional Auto-Encoder on synthetic and real data to learn the lower-level representations of their font detector ConvNet. Zhang *et al*. [42] learn a Multichannel Autoencoder to reduce the domain shift between real and synthetic data. In contrast to classical domain adaptation methods that adapt the features with respect to a specific prediction task, we bridge the gap between image distributions through adversarial training. This approach allows us to generate very realistic images which can be used to train any machine learning model, potentially for multiple tasks.

2. S+U Learning with SimGAN

The goal of Simulated+Unsupervised learning is to use a set of unlabeled real images $\mathbf{y}_i \in \mathcal{Y}$ to learn a refiner $R_{\boldsymbol{\theta}}(\mathbf{x})$ that refines a synthetic image $\mathbf{x}$, where $\boldsymbol{\theta}$ are the function parameters. Let the refined image be denoted by $\tilde{\mathbf{x}}$, then

$$\tilde{\mathbf{x}} := R_{\boldsymbol{\theta}}(\mathbf{x}).$$

The key requirement for S+U learning is that the refined image $\tilde{\mathbf{x}}$ should look like a real image in appearance while preserving the annotation information from the simulator.

To this end, we propose to learn $\boldsymbol{\theta}$ by minimizing a combination of two losses:

$$\mathcal{L}_R(\boldsymbol{\theta}) = \sum_i \ell_{\text{real}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathcal{Y}) + \lambda \ell_{\text{reg}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathbf{x}_i), \quad (1)$$

where $\mathbf{x}_i$ is the ith synthetic training image, and $\tilde{\mathbf{x}}_i$ is the corresponding refined image. The first part of the cost, ℓ_{real}, adds realism to the synthetic images, while the second part, ℓ_{reg}, preserves the annotation information by minimizing the difference between the synthetic and the refined images. In the following sections, we expand this formulation and provide an algorithm to optimize for $\boldsymbol{\theta}$.

2.1. Adversarial Loss with Self-Regularization

To add realism to the synthetic image, we need to bridge the gap between the distributions of synthetic and real images. An ideal refiner will make it impossible to classify a given image as real or refined with high confidence. This motivates the use of an adversarial discriminator network, $D_{\boldsymbol{\phi}}$, that is trained to classify images as real vs refined, where $\boldsymbol{\phi}$ are the the parameters of the discriminator network. The adversarial loss used in training the refiner network, R, is responsible for 'fooling' the network D into classifying the refined images as real. Following the GAN approach [7], we model this as a two-player minimax game, and update the refiner network, $R_{\boldsymbol{\theta}}$, and the discriminator network, $D_{\boldsymbol{\phi}}$, alternately. Next, we describe this intuition more precisely.

The discriminator network updates its parameters by minimizing the following loss:

$$\mathcal{L}_D(\boldsymbol{\phi}) = -\sum_i \log(D_{\boldsymbol{\phi}}(\tilde{\mathbf{x}}_i)) - \sum_j \log(1 - D_{\boldsymbol{\phi}}(\mathbf{y}_j)). \quad (2)$$

This is equivalent to cross-entropy error for a two class classification problem where $D_{\boldsymbol{\phi}}(.)$ is the probability of the input being a synthetic image, and $1 - D_{\boldsymbol{\phi}}(.)$ that of a real one. We implement $D_{\boldsymbol{\phi}}$ as a ConvNet whose last layer outputs the probability of the sample being a refined image. For training this network, each mini-batch consists of randomly sampled refined synthetic images $\tilde{\mathbf{x}}_i$'s and real images $\mathbf{y}_j$'s. The target labels for the cross-entropy loss layer are 0 for every $\mathbf{y}_j$, and 1 for every $\tilde{\mathbf{x}}_i$. Then $\boldsymbol{\phi}$ for a mini-batch is updated by taking a stochastic gradient descent (SGD) step on the mini-batch loss gradient.

In our implementation, the realism loss function ℓ_{real} in (1) uses the trained discriminator D as follows:

$$\ell_{\text{real}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathcal{Y}) = -\sum_i \log(1 - D_{\boldsymbol{\phi}}(R_{\boldsymbol{\theta}}(\mathbf{x}_i))). \quad (3)$$

By minimizing this loss function, the refiner forces the discriminator to fail classifying the refined images as synthetic. In addition to generating realistic images, the refiner network should preserve the annotation information of the simulator. For example, for gaze estimation the learned transformation should not change the gaze direction, and for hand pose estimation the location of the joints should not change. This is an essential ingredient to enable training a machine learning model that uses the refined images with the simulator's annotations. To enforce this, we propose using a self-regularization loss that minimizes the image difference between the synthetic and the refined image. Thus, the overall refiner

Algorithm 1: Adversarial training of refiner network $R_{\boldsymbol{\theta}}$

Input: Sets of synthetic images $\mathbf{x}_i \in \mathcal{X}$, and real images $\mathbf{y}_j \in \mathcal{Y}$, max number of steps ($T$), number of discriminator network updates per step (K_d), number of generative network updates per step (K_g).
Output: ConvNet model $R_{\boldsymbol{\theta}}$.
for $t = 1, \ldots, T$ **do**
  for $k = 1, \ldots, K_g$ **do**
    1. Sample a mini-batch of synthetic images $\mathbf{x}_i$.
    2. Update $\boldsymbol{\theta}$ by taking a SGD step on mini-batch loss $\mathcal{L}_R(\boldsymbol{\theta})$ in (4) .
  end
  for $k = 1, \ldots, K_d$ **do**
    1. Sample a mini-batch of synthetic images $\mathbf{x}_i$, and real images $\mathbf{y}_j$.
    2. Compute $\tilde{\mathbf{x}}_i = R_{\boldsymbol{\theta}}(\mathbf{x}_i)$ with current $\boldsymbol{\theta}$.
    3. Update $\boldsymbol{\phi}$ by taking a SGD step on mini-batch loss $\mathcal{L}_D(\boldsymbol{\phi})$ in (2).
  end
end

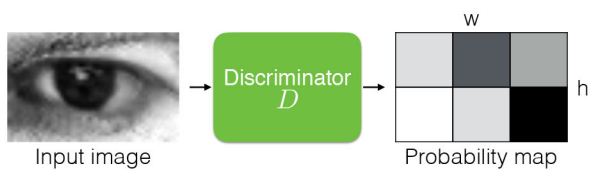

Figure 3. Illustration of local adversarial loss. The discriminator network outputs a $w \times h$ probability map. The adversarial loss function is the sum of the cross-entropy losses over the local patches.

loss function (1) used in our implementation is:

$$\mathcal{L}_R(\boldsymbol{\theta}) = -\sum_i \log(1 - D_{\boldsymbol{\phi}}(R_{\boldsymbol{\theta}}(\mathbf{x}_i))) + \lambda \|R_{\boldsymbol{\theta}}(\mathbf{x}_i) - \mathbf{x}_i\|_1, \quad (4)$$

where $\|.\|_1$ is ℓ_1 norm. We implement $R_{\boldsymbol{\theta}}$ as a fully convolutional neural net without striding or pooling. This modifies the synthetic image on a pixel level, rather than holistically modifying the image content as in *e.g.* a fully connected encoder network, and preserves the global structure and the annotations. We learn the refiner and discriminator parameters by minimizing $\mathcal{L}_R(\boldsymbol{\theta})$ and $\mathcal{L}_D(\boldsymbol{\phi})$ alternately. While updating the parameters of $R_{\boldsymbol{\theta}}$, we keep $\boldsymbol{\phi}$ fixed, and while updating $D_{\boldsymbol{\phi}}$, we fix $\boldsymbol{\theta}$. We summarize this training procedure in Algorithm 1.

2.2. Local Adversarial Loss

Another key requirement for the refiner network is that it should learn to model the real image characteristics without introducing any artifacts. When we train a single strong discriminator network, the refiner network tends to over-emphasize certain image features to fool the current discriminator network, leading to drifting and producing artifacts. A key observation is that any local patch we sample from the refined image, should have similar statistics to a real image patch. Therefore, rather than defining a global discriminator network, we can define discriminator network that classifies all local image patches separately. This not only limits the receptive field, and hence the capacity of the discriminator network, but also provides many samples per image for learning the discriminator network. This also improves training of the refiner network because we have multiple 'realism loss' values per image.

In our implementation, we design the discriminator D to be a fully convolutional network that outputs $w \times h$ dimensional probability map of patches belonging to fake class, where $w \times h$ are the number of local patches in the image. While training the refiner network, we sum the cross-entropy loss values over $w \times h$ local patches, as illustrated in Figure 3.

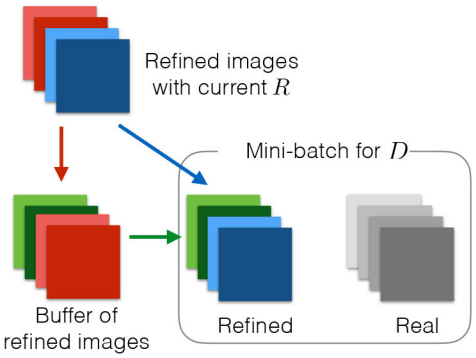

Figure 4. Illustration of using a history of refined images. See text for details.

2.3. Updating Discriminator using a History of Refined Images

Another problem of adversarial training is that the discriminator network only focuses on the latest refined images. This may cause (i) diverging of the adversarial training, and (ii) the refiner network re-introducing the artifacts that the discriminator has forgotten about. Any refined image generated by the refiner network at any time during the entire training procedure is a 'fake' image for the discriminator. Hence, the discriminator should be able to classify all these images as fake. Based on this observation, we introduce a method to improve the stability of adversarial training by updating the discriminator using a history of refined images, rather than only the ones in the current mini-batch. We slightly modify Algorithm 1 to have a buffer of refined images generated by previous networks. Let B be the size of the buffer and b be the mini-batch size used in Algorithm 1.

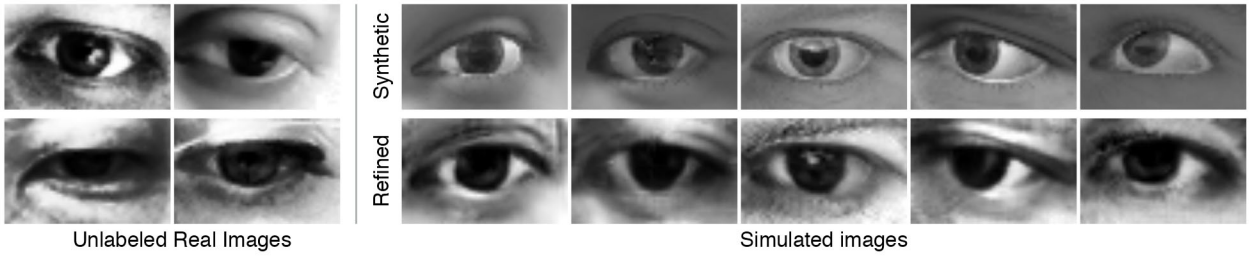

Figure 5. Example output of SimGAN for the UnityEyes gaze estimation dataset [40]. (Left) real images from MPIIGaze [43]. Our refiner network does not use any label information from MPIIGaze dataset at training time. (Right) refinement results on UnityEye. The skin texture and the iris region in the refined synthetic images are qualitatively significantly more similar to the real images than to the synthetic images. More examples are included in the supplementary material.

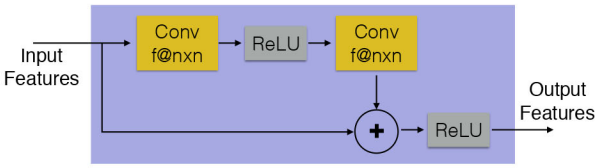

Figure 6. A ResNet block with two $n \times n$ convolutional layers, each with f feature maps.

At each iteration of discriminator training, we compute the discriminator loss function by sampling $b/2$ images from the current refiner network, and sampling an additional $b/2$ images from the buffer to update parameters ϕ. We keep the size of the buffer, B, fixed. After each training iteration, we randomly replace $b/2$ samples in the buffer with the newly generated refined images. This procedure is illustrated in Figure 4.

3. Experiments

We evaluate our method for appearance-based gaze estimation in the wild on the MPIIGaze dataset [40, 43], and hand pose estimation on the NYU hand pose dataset of depth images [35]. We use fully convolutional refiner network with ResNet blocks (Figure 6) for all our experiments.

3.1. Appearance-based Gaze Estimation

Gaze estimation is a key ingredient for many human computer interaction (HCI) tasks. However, estimating the gaze direction from an eye image is challenging, especially when the image is of low quality, *e.g.* from a laptop or a mobile phone camera – annotating the eye images with a gaze direction vector is challenging even for humans. Therefore, to generate large amounts of annotated data, several recent approaches [40, 43] train their models on large amounts of synthetic data. Here, we show that training with the refined synthetic images generated by SimGAN significantly outperforms the state-of-the-art for this task.

The gaze estimation dataset consists of 1.2M synthetic images from eye gaze synthesizer UnityEyes [40] and 214K real images from the MPIIGaze dataset [43] – samples shown in Figure 5. MPIIGaze is a very challenging eye gaze estimation dataset captured under extreme illumination conditions. For UnityEyes we use a single generic rendering environment to generate training data without any dataset-specific targeting.

Qualitative Results: Figure 5 shows examples of synthetic, real and refined images from the eye gaze dataset. As shown, we observe a significant qualitative improvement of the synthetic images: SimGAN successfully captures the skin texture, sensor noise and the appearance of the iris region in the real images. Note that our method preserves the annotation information (gaze direction) while improving the realism.

'Visual Turing Test': To quantitatively evaluate the visual quality of the refined images, we designed a simple user study where subjects were asked to classify images as real or refined synthetic. Each subject was shown a random selection of 50 real images and 50 refined images in a random order, and was asked to label the images as either real or refined. The subjects were constantly shown 20 examples of real and refined images while performing the task. The subjects found it very hard to tell the difference between the real images and the refined images. In our aggregate analysis, 10 subjects chose the correct label 517 times out of 1000 trials ($p = 0.148$), which is not significantly better than chance. Table 1 shows the confusion matrix. In contrast, when testing on original synthetic images vs real images, we showed 10 real and 10 synthetic images per subject, and the subjects chose correctly 162 times out of 200 trials ($p \leq 10^{-8}$), which is significantly better than chance.

Quantitative Results: We train a simple convolutional neural network (CNN) similar to [43] to predict the eye gaze direction (encoded by a 3-dimensional vector for x, y, z) with l_2 loss. We train on UnityEyes and test on MPIIGaze. Figure 7 and Table 2 compare the performance of a gaze estimation CNN trained on synthetic data to that of another CNN trained on refined

	Selected as real	Selected as synt
Ground truth real	224	276
Ground truth synt	207	293

Table 1. Results of the 'Visual Turing test' user study for classifying real vs refined images. Subjects were asked to distinguish between refined synthetic images (output from our method) and real images (from MPIIGaze). The average human classification accuracy was 51.7%, demonstrating that the automatically generated refined images are visually very hard to distinguish from real images.

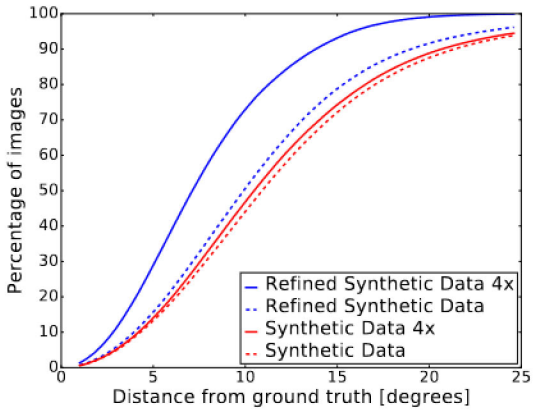

Figure 7. Quantitative results for appearance-based gaze estimation on the MPIIGaze dataset with real eye images. The plot shows cumulative curves as a function of degree error as compared to the ground truth eye gaze direction, for different numbers of training examples of synthetic and refined synthetic data. Gaze estimation using the refined images instead of the synthetic images results in significantly improved performance.

synthetic data, the output of SimGAN. We observe a large improvement in performance from training on the SimGAN output, a 22.3% absolute percentage improvement. We also observe a large improvement from training on more training data – here 4x refers to 100% of the training dataset. The quantitative evaluation confirms the value of the qualitative improvements observed in Figure 5, and shows that machine learning models generalize significantly better using SimGAN.

Table 3 shows a comparison to the state-of-the-art. Training the CNN on the refined images outperforms the state-of-the-art on the MPIIGaze dataset, with a relative improvement of 21%. This large improvement shows the practical value of our method in many HCI tasks.

Implementation Details: The refiner network, $R_{\boldsymbol{\theta}}$, is a residual network (ResNet) [11]. Each ResNet block consists of two convolutional layers containing 64 feature maps as shown in Figure 6. An input image of size 55×35 is convolved with 3×3 filters that output 64 feature maps. The output is passed through 4 ResNet blocks. The output of the last ResNet block is passed to a 1×1 convolutional layer producing 1 feature map corresponding to the refined synthetic image.

Training data	% of images within d
Synthetic Data	62.3
Synthetic Data 4x	64.9
Refined Synthetic Data	69.4
Refined Synthetic Data 4x	**87.2**

Table 2. Comparison of a gaze estimator trained on synthetic data and the output of SimGAN. The results are at distance $d = 7$ degrees from ground truth. Training on the refined synthetic output of SimGAN outperforms training on synthetic data by 22.3%, without requiring supervision for the real data.

Method	R/S	Error
Support Vector Regression (SVR) [30]	R	16.5
Adaptive Linear Regression ALR) [21]	R	16.4
Random Forest (RF) [33]	R	15.4
kNN with UT Multiview [43]	R	16.2
CNN with UT Multiview [43]	R	13.9
k-NN with UnityEyes [40]	S	9.9
CNN with UnityEyes Synthetic Images	S	11.2
CNN with UnityEyes Refined Images	S	**7.8**

Table 3. Comparison of SimGAN to the state-of-the-art on the MPIIGaze dataset of real eyes. The second column indicates whether the methods are trained on Real/Synthetic data. The error the is mean eye gaze estimation error in degrees. Training on refined images results in a 2.1 degree improvement, a relative 21% improvement compared to the state-of-the-art.

The discriminator network, $D_{\boldsymbol{\phi}}$, contains 5 convolution layers and 2 max-pooling layers as follows: (1) Conv3x3, stride=2, feature maps=96, (2) Conv3x3, stride=2, feature maps=64, (3) MaxPool3x3, stride=1, (4) Conv3x3, stride=1, feature maps=32, (5) Conv1x1, stride=1, feature maps=32, (6) Conv1x1, stride=1, feature maps=2, (7) Softmax.

Our adversarial network is fully convolutional, and has been designed such that the receptive field of the last layer neurons in $R_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\phi}}$ are similar. We first train the $R_{\boldsymbol{\theta}}$ network with just self-regularization loss for $1,000$ steps, and $D_{\boldsymbol{\phi}}$ for 200 steps. Then, for each update of $D_{\boldsymbol{\phi}}$, we update $R_{\boldsymbol{\theta}}$ twice, *i.e.* K_d is set to 1, and K_g is set to 50 in Algorithm 1.

The eye gaze estimation network is similar to [43], with some changes to enable it to better exploit our large synthetic dataset. The input is a 35×55 grayscale image that is passed through 5 convolutional layers followed by 3 fully connected layers, the last one encoding the 3-dimensional gaze vector: (1) Conv3x3, feature maps=32, (2) Conv3x3, feature maps=32, (3) Conv3x3, feature maps=64, (4) MaxPool3x3, stride=2, (5) Conv3x3, feature maps=80, (6) Conv3x3, feature maps=192, (7) MaxPool2x2,

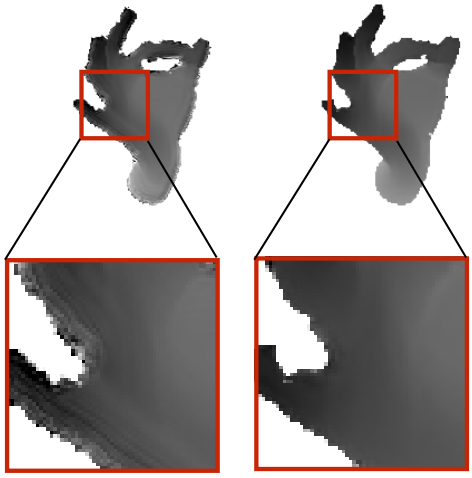

Figure 8. Importance of using a local adversarial loss. (Left) an example image that has been generated with a standard 'global' adversarial loss on the whole image. The noise around the edge of the hand contains obvious unrealistic depth boundary artifacts. (Right) the same image generated with a local adversarial loss that looks significantly more realistic.

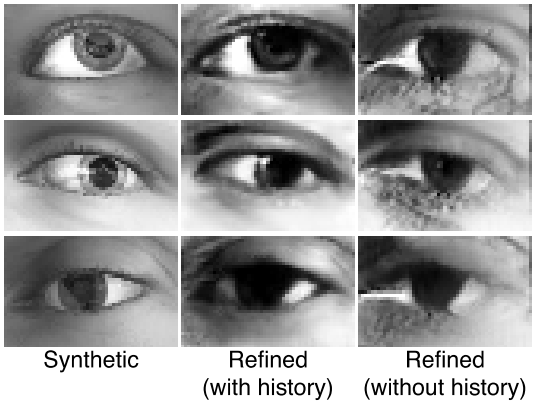

Figure 9. Using a history of refined images for updating the discriminator. (Left) synthetic images; (middle) result of using the history of refined images; (right) result without using a history of refined images (instead using only the most recent refined images). We observe obvious unrealistic artifacts, especially around the corners of the eyes.

stride=2, (8) FC9600, (9) FC1000, (10) FC3, (11) Euclidean loss. All networks are trained with a constant 0.001 learning rate and 512 batch size, until the validation error converges.

3.2. Hand Pose Estimation from Depth Images

Next, we evaluate our method for hand pose estimation in depth images. We use the NYU hand pose dataset [35] that contains $72,757$ training frames and $8,251$ testing frames captured by 3 Kinect cameras – one frontal and 2 side views. Each depth frame is labeled with hand pose information that has been used to create

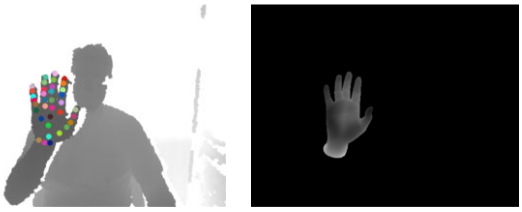

Figure 10. NYU hand pose dataset. (Left) depth frame; (right) corresponding synthetic image.

a synthetic depth image. Figure 10 shows one such example frame. We pre-process the data by cropping the pixels from real images using the synthetic images. The images are resized to 224×224 before passing them to the ConvNet. The background depth values are set to zero and the foreground values are set to original depth value minus 2000 (assuming that the background is at 2000 millimeters).

Qualitative Results: Figure 11 shows example output of SimGAN on the NYU hand pose test set. As is apparent from the figure, the main source of noise in real depth images is from depth discontinuity at the edges. SimGAN is able to learn to model this kind of noise without requiring any label information for the real images, resulting in more realistic-looking images for this domain as well.

Quantitative Results: We train a fully convolutional hand pose estimator CNN similar to Stacked Hourglass Net [22] on real, synthetic and refined synthetic images of the NYU hand pose training set, and evaluate each model on all *real* images in the NYU hand pose test set. We train on the same 14 hand joints as in [35]. Many state-of-the-art hand pose estimation methods are customized pipelines that consist of several steps. We use only a single deep neural network to analyze the effect of improving the synthetic images to avoid bias due to other factors. Figure 12 and Table 4 present quantitative results on NYU hand pose. Training on refined synthetic data – the output of SimGAN which does not require any labeling for the real images – significantly outperforms the model trained on real images with supervision, by 8.8%. The proposed method also outperforms training on synthetic data. We also observe a large improvement as the number of training examples is increased, which comes with zero annotation cost to us as we train on the output of a simulator – here 3x corresponds to training on all views.

Implementation Details: The architecture is the same as for eye gaze estimation, except the input image size is 224×224, filter size is 7×7, and 10 ResNet blocks are used. The discriminative net D_ϕ is: (1) Conv7x7, stride=4, feature maps=96, (2) Conv5x5, stride=2, feature maps=64, (3) MaxPool3x3, stride=2, (4) Conv3x3,

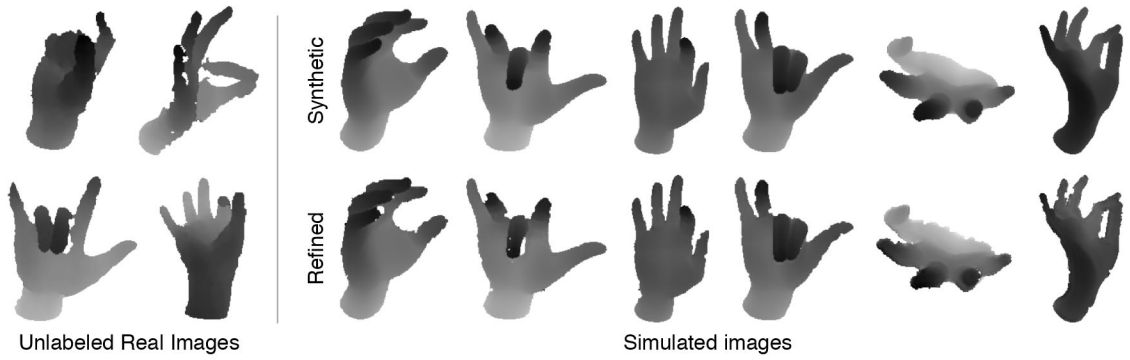

Figure 11. Example refined test images for the NYU hand pose dataset [35]. (Left) real images, (right) synthetic images and the corresponding refined output images from the refiner network. The major source of noise in the real images is the non-smooth depth boundaries. The refiner network learns to model the noise present in the real images, importantly without requiring any labels for the real images.

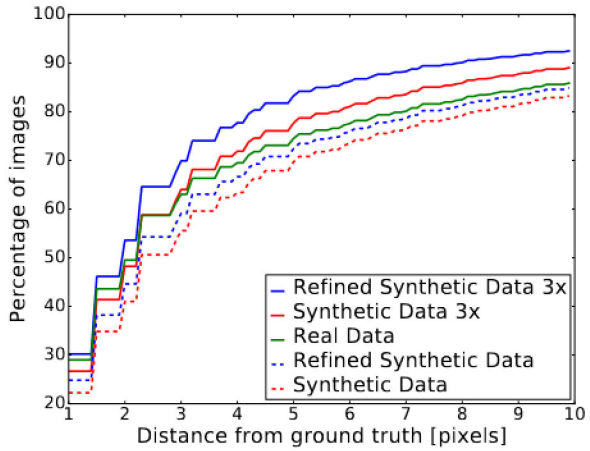

Figure 12. Quantitative results for hand pose estimation on the NYU hand pose test set of real depth images [35]. The plot shows cumulative curves as a function of distance from ground truth keypoint locations, for different numbers of training examples of synthetic and refined images. Training a pose estimator on the output of SimGAN significantly outperforms the same network trained on real images. Importantly, our refiner generative model does not require labeling for the real images.

Training data	% of images within d
Synthetic Data	69.7
Refined Synthetic Data	72.4
Real Data	74.5
Synthetic Data 3x	77.7
Refined Synthetic Data 3x	**83.3**

Table 4. Comparison of a hand pose estimator trained on synthetic data, real data, and the output of SimGAN. The results are at distance $d = 5$ pixels from ground truth. Training on the output of SimGAN outperforms training on supervised real data by 8.8%, without requiring any supervision.

stride=2, feature maps=32, (5) Conv1x1, stride=1, feature maps=32, (6) Conv1x1, stride=1, feature maps=2, (7) Softmax. We train the $R_{\boldsymbol{\theta}}$ network first with just self-regularization loss for 500 steps and $D_{\boldsymbol{\phi}}$ for 200 steps; then, for each update of $D_{\boldsymbol{\phi}}$ we update $R_{\boldsymbol{\theta}}$ twice, *i.e.* K_d is set to 1, and K_g is set to 2 in Algorithm 1.

For hand pose estimation, we use the Stacked Hourglass Net of [22] 2 hourglass blocks, and an output heatmap size 64×64. We augment at training time with random $[-20, 20]$ degree rotations and crops. All networks are trained until the validation error converges.

3.3. Analysis of Modifications to Adversarial Training

First we compare local vs global adversarial loss during training. A global adversarial loss uses a fully connected layer in the discriminator network, classifying the whole image as real vs refined. The local adversarial loss removes the artifacts and makes the generated image significantly more realistic, as seen in Figure 8.

Next, in Figure 9, we show result of using history of refined images, and compare it with standard adversarial training for gaze estimation. As shown in the figure, using the buffer of refined images prevents severe artifacts in standard training, *e.g.* around the corner of the eyes.

4. Conclusions and Future Work

We have proposed Simulated+Unsupervised learning to refine a simulator's output with unlabeled real data. S+U learning adds realism to the simulator and preserves the global structure and the annotations of the synthetic images. We described SimGAN, our method for S+U learning, that uses an adversarial network and demonstrated state-of-the-art results without any labeled real data. In future, we intend to explore modeling the noise distribution to generate more than one refined image for each synthetic image, and investigate refining videos rather than single images.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

[3] T. Darrell, P. Viola, and G. Shakhnarovich. Fast pose estimation with parameter sensitive hashing. In *Proc. CVPR*, 2015.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009.

[5] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. CVPR*, 2016.

[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

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

[8] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. *Proc. CVPR*, 2016.

[9] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proc. ECCV*, 2014.

[10] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. SceneNet: Understanding real world indoor scenes with synthetic data. In *Proc. CVPR*, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[12] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generating images with recurrent adversarial networks. *http://arxiv.org/abs/ 1602.05110*, 2016.

[13] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014.

[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016.

[15] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, A. Gupta, D. Narayanan, C. Sun, G. Chechik, and K. Murphy. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016.

[16] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. CVPR*, 2004.

[17] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. ECCV*, 2016.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, 2014.

[19] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Proc. NIPS*, 2016.

[20] W. Lotter, G. Kreiman, and D. Cox. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380*, 2015.

[21] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Adaptive linear regression for appearance-based gaze estimation. *PAMI*, 36(10):2033–2046, 2014.

[22] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *arXiv preprint arXiv:1603.06937*, 2016.

[23] D. Park and D. Ramanan. Articulated pose estimation with tiny synthetic videos. In *Proc. CVPR*, 2015.

[24] X. Peng, B. Sun, K. Ali, and K. Saenko. Learning deep object detectors from 3d models. In *Proc. ICCV*, 2015.

[25] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Proc. CVPR*, 2012.

[26] W. Qiu and A. Yuille. UnrealCV: Connecting computer vision to Unreal Engine. *arXiv preprint arXiv:1609.01326*, 2016.

[27] G. Rogez and C. Schmid. MoCap-guided data augmentation for 3d pose estimation in the wild. *arXiv preprint arXiv:1607.02046*, 2016.

[28] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. CVPR*, 2016.

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[30] T. Schneider, B. Schauerte, and R. Stiefelhagen. Manifold alignment for person independent appearance-based gaze estimation. In *Proc. ICPR*, 2014.

[31] A. Shafaei, J. Little, and M. Schmidt. Play and learn: Using video games to train computer vision models. In *Proc. BMVC*, 2016.

[32] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *PAMI*, 35(12):2821–2840, 2013.

[33] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proc. CVPR*, 2014.

[34] J. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proc. CVPR*, 2015.

[35] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graphics*, 2014.

[36] O. Tuzel, Y. Taguchi, and J. Hershey. Global-local face upsampling network. *arXiv preprint arXiv:1603.07235*, 2016.

[37] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[38] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *Proc. ECCV*, 2016.

[39] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. Huang. Deepfont: Identify your font from an image. In *Proc. ACMM*, 2015.

[40] E. Wood, T. Baltrušaitis, L. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proc. ACM Symposium on Eye Tracking Research & Applications*, 2016.

[41] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *arXiv preprint arXiv:1609.05473*, 2016.

[42] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam. Learning classifiers from synthetic data using a multichannel autoencoder. *arXiv preprint arXiv:1503.03163*, 2015.

[43] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proc. CVPR*, 2015.

[44] Y. Zhang, K. Lee, and H. Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *Proc. ICML*, 2016.

[45] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. Efros. Generative visual manipulation on the natural image manifold. In *Proc. ECCV*, 2016.

Additional Experiments

Qualitative Experiments for Appearance-based Gaze Estimation

Dataset: The gaze estimation dataset consists of 1.2M synthetic images from eye gaze synthesizer UnityEyes [40] and 214K real images from the MPIIGaze dataset [43] – samples shown in Figure 13. MPIIGaze is a very challenging eye gaze estimation dataset captured under extreme illumination conditions. For UnityEyes we use a single generic rendering environment to generate training data without any dataset-specific targeting.

Qualititative Results: In Figure 14, we show many examples of synthetic, and refined images from the eye gaze dataset. We show many pairs of synthetic and refined in multiple rows. The top row contains synthetic images, and the bottom row contains corresponding refined images. As shown, we observe a significant qualitative improvement of the synthetic images: SimGAN successfully captures the skin texture, sensor noise and the appearance of the iris region in the real images. Note that our method preserves the annotation information (gaze direction) while improving the realism.

Qualitative Experiments for Hand Pose Estimation

Dataset: Next, we evaluate our method for hand pose estimation in depth images. We use the NYU hand pose dataset [35] that contains $72,757$ training frames and $8,251$ testing frames. Each depth frame is labeled with hand pose information that has been used to create a synthetic depth image. We pre-process the data by cropping the pixels from real images using the synthetic images. Figure 15 shows example real depth images from the dataset. The images are resized to 224×224 before passing them to the refiner network.

Quantative Results: We show examples of synthetic and refined hand depth images in Figure 16 from the test set. We show our results in multiple pairs of rows. The top row in each pair, contains synthetic depth image, and the bottom row shows the corresponding refined image using the proposed SimGAN approach. Note the realism added to the depth boundary in the refined images, compare to the real images in Figure 15.

Convergence Experiment

To investigate the convergence of our method, we visualize intermediate results as training progresses. As shown in Figure 17, in the beginning, the refiner network learns to predict very smooth edges using only the self-regularization loss. As the adversarial loss is enabled, the network starts adding artifacts at the depth boundaries. However, as these artifacts are not the same as real images, the discriminator easily learns to differentiate between the real and refined images. Slowly the network starts adding realistic noise, and after many steps, the refiner generates very realistic-looking images. We found it helpful to train the network with a low learning rate and for a large number of steps. For NYU hand pose we used lr=0.0002 in the beginning, and reduced to 0.00005 after $600,000$ steps.

Figure 13. Example real images from MPIIGaze dataset.

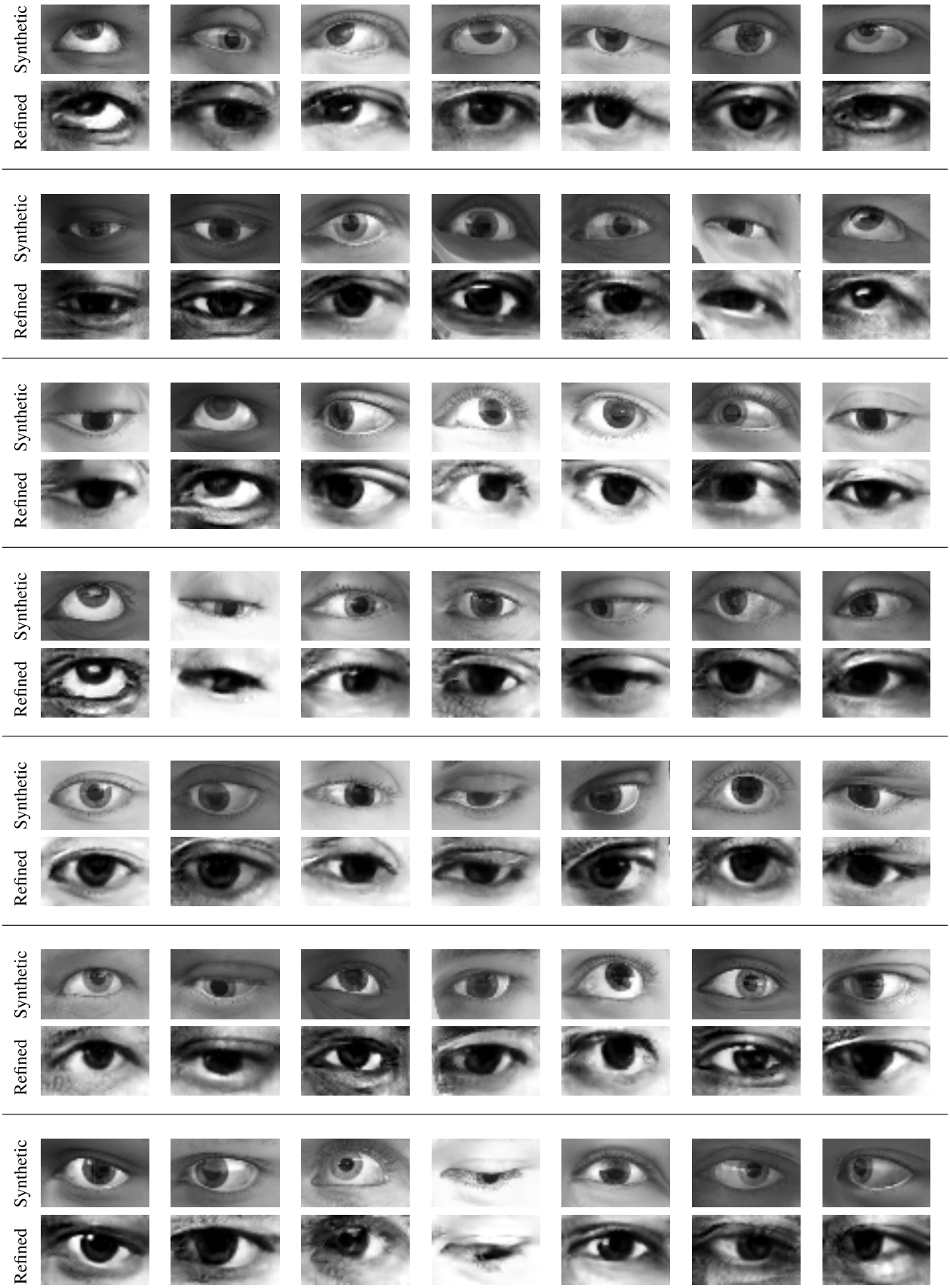

Figure 14. Qualitative results for automatic refinement of simulated eyes. The top row (in each set of two rows) shows the synthetic eye image, and the bottom row shows the corresponding refined image.

Figure 15. Example real test images in the NYU hand dataset.

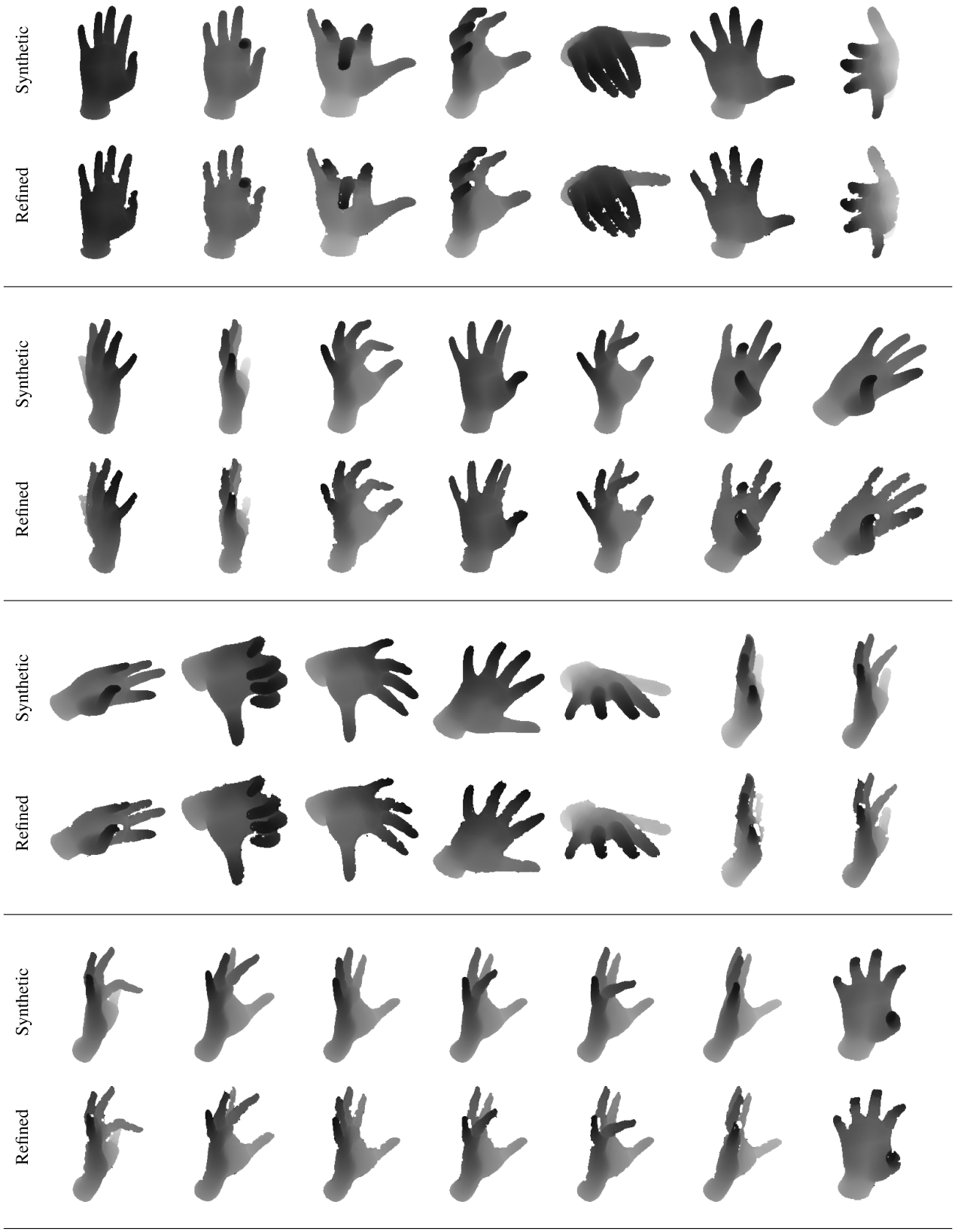

Figure 16. Qualitative results for automatic refinement of NYU hand depth images. The top row (in each set of two rows) shows the synthetic hand image, and the bottom row is the corresponding refined image. Note how realistic the depth boundaries are compared to real images in Figure 15.

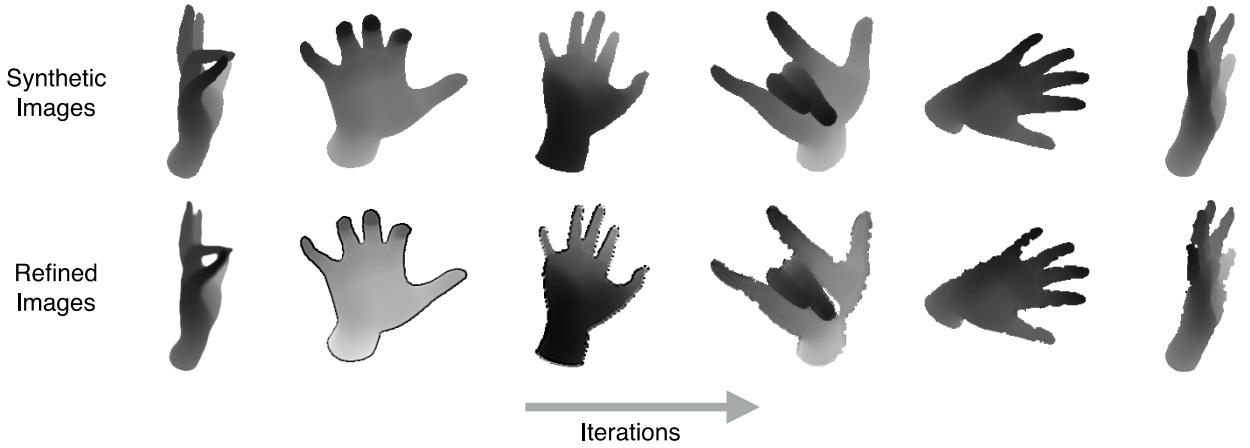

Figure 17. SimGAN output as a function of training iterations for NYU hand pose. Columns correspond to increasing training iterations. First row shows synthetic images, and the second row shows corresponding refined images. The first column is the result of training with ℓ_1 image difference for 300 steps; the later rows show the result when trained on top of this model. In the beginning the adversarial part of the cost introduces different kinds of unrealistic noise to try beat the adversarial network $D_{\boldsymbol{\phi}}$. As the dueling between $R_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\phi}}$ progresses, $R_{\boldsymbol{\theta}}$ learns to model the right kind of noise.