SharinGAN: Combining Synthetic and Real Data for Unsupervised Geometry Estimation

Koutilya PNVR  Hao Zhou*  David Jacobs
koutilya@terpmail.umd.edu  hzhou@cs.umd.edu  djacobs@umiacs.umd.edu

University of Maryland, College Park, MD, USA.

Abstract

We propose a novel method for combining synthetic and real images when training networks to determine geometric information from a single image. We suggest a method for mapping both image types into a single, shared domain. This is connected to a primary network for end-to-end training. Ideally, this results in images from two domains that present shared information to the primary network. Our experiments demonstrate significant improvements over the state-of-the-art in two important domains, surface normal estimation of human faces and monocular depth estimation for outdoor scenes, both in an unsupervised setting.

1. Introduction

Understanding geometry from images is a fundamental problem in computer vision. It has many important applications. For instance, Monocular Depth Estimation (MDE) is important for synthetic object insertion in computer graphics [18], grasping in robotics [21] and safety in self-driving cars. Face Normal Estimation can help in face image editing applications such as relighting [39, 49, 54]. However, it is extremely hard to annotate real data for these regression tasks. Synthetic data and their ground truth labels, on the other hand, are easy to generate and are often used to compensate for the lack of labels in real data. Deep models trained on synthetic data, unfortunately, usually perform poorly on real data due to the domain gap between synthetic and real distributions. To deal with this problem, several research studies [30, 53, 52, 3] have proposed unsupervised domain adaptation methods to take advantage of synthetic data by mapping it into the real domain or vice versa, either at the feature level or image level. However, mapping examples from one domain to another domain itself is a challenging problem that can limit performance.

We observe that finding such a mapping solves an unnecessary difficult problem. To train a regressor that applies to both real and synthetic domains, it is only necessary that we map both to a new representation that contains the task-relevant information present in both domains, in a common form. The mapping need not alter properties of the original domain that are irrelevant to the task since the regressor will learn to ignore them regardless.

To see this, we consider a simplified model of our problem. We suppose that real and synthetic images are formed by two components: domain agnostic (which has semantic information shared across synthetic and real, and is denoted as $I$) and domain specific. We further assume that domain specific information has two sub-components: domain specific information unrelated to the primary task (denoted as $\delta'_s$ and $\delta'_r$ for synthetic and real images respectively) and domain specific information related to the primary task ($\delta_s$, $\delta_r$). So real and synthetic images can be represented as: $x_r = f(I, \delta_r, \delta'_r)$ and $x_s = f(I, \delta_s, \delta'_s)$ respectively.

We believe the domain gap between $\{\delta_s$ and $\delta'_r\}$ can affect the training of the primary network, which learns to expect information that is not always present. The domain gap between $\{\delta'_s$ and $\delta'_r\}$, on the other hand, can be bypassed by the primary network since it does not hold information needed for the primary task. For example, in real face images, information such as the color and texture of the hair is unrelated to the task of estimating face normals but is discriminative enough to distinguish real from synthetic faces. This can be regarded as domain specific information unrelated to the primary task i.e., $\delta'_r$. On the other hand, shad-
ows in the real and synthetic images, due to the limitations of the rendering engine, may have different appearances but may contain depth cues that are related to the primary task of MDE in both domains. The simplest strategy, then, for combining real and synthetic data is to map $\delta_s$ and $\delta_r$ to a shared representation, $\delta_{sh}$, while not modifying $\delta'_s$ and $\delta'_r$, as shown in Figure 1.

Recent research studies show that a shared network for synthetic and real data can help reduce the discrepancy between images in different domains. For instance, [39] achieved state-of-the-art results in face normal estimation by training a unified network for real and synthetic data. [25] learned the joint distribution of multiple domain images by enforcing a weight-sharing constraint for different generative networks. Inspired by these research studies, we define a unified mapping function $G$, which is called SharinGAN, to reduce the domain gap between real and synthetic images.

Different from existing research studies, our $G$ is trained so that minimum domain specific information is removed. This is achieved by pre-training $G$ as an auto-encoder on real and synthetic data, i.e., initializing $G$ as an identity function. Then $G$ is trained end-to-end with reconstruction loss in an adversarial framework, along with a network that solves the primary task, further pushing $G$ to map information relevant to the task to a shared domain.

As a result, a successfully trained $G$ will learn to reduce the domain gap existing in $\delta_s$ and $\delta_r$, mapping them into a shared domain $\delta_{sh}$. $G$ will leave $I$ unchanged. $\delta'_s$ and $\delta'_r$ can be left relatively unchanged when it is difficult to map them to a common representation. Mathematically, $G(x_s) = f(I, \delta_{sh}, \delta'_s)$ and $G(x_r) = f(I, \delta_{sh}, \delta'_r)$. If successful, $G$ will map synthetic and real images to images that may look quite different to the eye, but the primary task network will extract the same information from both.

We apply our method to unsupervised monocular depth estimation using virtual KITTI (vKITTI) [1] and KITTI [29] as synthetic and real datasets respectively. Our method reduces the absolute error in the KITTI eigen test split and further trained a task network in this translated domain. GASDA [52] proposed to train the network by incorporating epipolar geometry constraints for real data along with the ground truth labels for synthetic data. All these methods try to align two domains by transferring one domain to another. Unlike these works, we propose a mapping function $G$, also called SharinGAN, to just align the domain specific information that affects the primary task, resulting in a minimum change in the images in both domains. We show that this makes learning the primary task network much easier and can help it focus on the useful information.

Self-supervised learning is another way to avoid collecting ground truth labels for monocular depth estimation. Such methods need monocular videos [55, 48, 6, 13], stereo pairs [11, 28, 31, 27], or both [13] for training. Our proposed method improves over SfSNet [39] on face normal estimation. It yields an accuracy boost of nearly 4.3% for normal prediction within $20^\circ$ (Acc < $20^\circ$) of ground truth on the Photoface dataset [51].

2. Related Work

Monocular Depth Estimation has long been an active area in computer vision. Because this problem is ill-posed, learning-based methods have predominated in recent years. Many early learning works applied Markov Random Fields (MRF) to infer the depth from a single image by modeling the relation between nearby regions [36, 37, 24]. These methods, however, are time-consuming during inference and rely on manually defined features, which have limitations in performance.

More recent studies apply deep Convolutional Neural Networks (CNNs) [8, 7, 22, 16, 50, 33, 32, 34] to monocular depth estimation. Eigen et al. [8] first proposed a multi-scale deep CNN for depth estimation. Following this work, [7] proposed to apply CNNs to estimate depth, surface normal and semantic labels together. [22] combined deep CNNs with a continuous CRF for monocular depth estimation. One major drawback of these supervised learning-based methods is the requirement for a huge amount of annotated data, which is hard to obtain in reality.

With the emergence of large scale, high-quality synthetic data [1], using synthetic data to train a depth estimator network for real data became popular [53, 52]. The biggest challenge for this task is the large domain gap between synthetic data and real data. [3] proposed to first train a depth prediction network using synthetic data. A style transfer network is then trained to map real images to synthetic images in a cycle consistent manner [56]. [30] proposed to adapt the features of real images to the features of synthetic images by applying adversarial loss on latent features. A content congruent regularization is further proposed to avoid mode collapse. T3Net [53] trained a network that translates synthetic data into real at the image level and further trained a task network in this translated domain.

Face Geometry Estimation is a sub-problem of inverse face rendering which is the key for many applications such as face image editing. Conventional face geometry estimation methods are usually based on 3D Morphable Models (3DMM) [4]. Recent studies demonstrate the effectiveness of deep CNNs for solving this problem [44, 42, 10, 39, 46, 45, 23]. Thanks to the 3DMM, generating synthetic face images with ground truth geometry is easy. [44, 42, 39] make use of synthetic face images with
ground truth shape to help train a network for predicting face shape using real images. Most of these works initially pre-train the network with synthetic data and then fine-tune it with a mix of real and synthetic data, either using no supervision or weak supervision, overlooking the domain gap between real and synthetic face images. In this work, we show that by reducing the domain gap between real and synthetic data, using our proposed method, face geometry can be better estimated.

**Domain Adaptation using GANs** There are many works [47, 5, 25, 43, 40] that use a GAN framework to perform domain adaptation by mapping one domain into another via a supervised translation. However, most of these works show performance on just toy datasets in a classification setting. We attempt to map both synthetic and real domains into a new shared domain that is learned during training and can solve complex problems of unsupervised geometry estimation. Moreover, we apply adversarial loss at the image level for our regression task, in contrast to some of the above previous works where domain invariant feature engineering sufficed for classification tasks.

### 3. Method

To compensate for the lack of annotations for real data and to train a primary task network on easily available synthetic data, we propose SharinGAN to reduce the domain gap between synthetic and real. We aim to train a primary task network on a shared domain created by SharinGAN, which learns the mapping function $G : x_r \mapsto x_{s}^{sh}$ and $G : x_s \mapsto x_{r}^{sh}$, where $x_k = f(I, \delta_k, \delta'_k) ; x_k^{sh} = f(I, \delta_{sh}, \delta'_k) ; k \in \{ r, s \}$ as shown in Figure 1. $G$ allows the primary task network to train on a shared space that holds the information needed to do the primary task, making the network more applicable to real data during testing. To achieve this, an adversarial loss is used to find the shared information, $\delta_{sh}$. This is done by minimizing the discrepancy in the distributions of $x_{s}^{sh}$ and $x_{r}^{sh}$. But at the same time, to preserve the domain agnostic information (shared semantic information $I$), we use reconstruction loss. Now, without a loss from the primary task network, $G$ might change the images so that they don’t match the labels. To prevent that, we additionally use a primary task loss for both real and synthetic examples to guide the generator. It is important to note that both the translations from synthetic to real and vice versa are equally crucial for this symmetric setup to find a shared space. To facilitate that, we use a form of weak supervision we call virtual supervision. Some possible virtual supervisions include a prior on the input data or a constraint that can narrow the solution space for the primary task network (details discussed in 3.2.2). For synthetic examples, we use the known labels.

Adversarial, Reconstruction and Primary task losses together train the generator and primary task network to align the domain specific information $\{ \delta_{s}, \delta_{r} \}$ in both the domains into a shared space $\delta_{sh}$, preserving everything else.

### 3.1. Framework

In this work, we propose to train a generative network which is called SharinGAN, to reduce the domain gap between real and synthetic data so as to help to train the primary network. Figure 2 shows the framework of our proposed method. It contains a generative network $G$, a discriminator on image-level $D$ that embodies the SharinGAN module and a task network $T$ to perform the primary task. The generative network $G$ takes either a synthetic image $x_s$ or real image $x_r$ as input and transforms it to $x_s^{sh}$ or $x_r^{sh}$ in an attempt to fool $D$. Different from existing works that transfer images in one domain to another [3, 53, 52], our
generative network G tries to map the domain specific parts \(\delta_s\) and \(\delta_r\) of synthetic and real images to a shared space \(\delta_{sh}\), leaving \(\delta'_s\) and \(\delta'_r\) unchanged. As a result, our transformed synthetic and real images \((x_s^{sh} \text{ and } x_r^{sh})\) have fewer differences from \(x_s\) and \(x_r\). Our task network \(T\) then takes the transformed images \(x_s^{sh}\) and \(x_r^{sh}\) as input and predicts the geometry. The generative network \(G\) and task network \(T\) are trained together in an end-to-end manner.

3.2. Losses

In this section, we describe the losses we use for the generative and task networks.

3.2.1 Losses for Generative Network

We design a single generative network \(G\) for synthetic and real data since sharing weights can help align distributions of different domains [25]. Moreover, existing research studies such as [42, 39] also demonstrate that a unified framework works reasonably well on synthetic and real images. In order to map \(\delta_s\) and \(\delta_r\) to a shared space \(\delta_{sh}\), we apply adversarial loss [14] at the image level. More specifically, we use the Wasserstein discriminator [2] that uses the Earth-Mover’s distance to minimize the discrepancy between the distributions for synthetic and real examples \(\{G(x_s), G(x_r)\}\), i.e.:

\[
L_W(D, G) = \mathbb{E}_{x_s}[D(G(x_s))] - \mathbb{E}_{x_r}[D(G(x_r))],
\]

(1)

\(D\) is a discriminator and \(G_r\) is the encoder part of the generator. Following [15], to overcome the problem of vanishing or exploding gradients due to the weight clipping proposed in [2], a gradient penalty term is added for training the discriminator:

\[
L_{gp}(D) = (||\nabla_h D(\hat{h})||_2 - 1)^2
\]

(2)

Our overall adversarial loss is then defined as:

\[
L_{adv} = L_W(D, G) - \lambda L_{gp}(D)
\]

(3)

where \(\lambda\) is chosen to be 10 while training the discriminator and 0 while training the generator.

Without any constraints, the adversarial loss may learn to remove all domain specific parts \(\delta\) and \(\delta'\) or even some of the domain agnostic part \(I\) in order to fool the discriminator. This may lead to loss of geometric information, which can degrade the performance of the primary task network \(T\). To avoid this, we propose to use the self-regularization loss similar to [41] to force the transformed image to keep as much information as possible:

\[
L_r = ||G(x_s) - x_s||_2^2 + ||G(x_r) - x_r||_2^2.
\]

(4)

3.2.2 Losses for the Task Network

The task network takes transformed synthetic or real images as input and predicts geometric information. Since the ground truth labels for synthetic data are available, we apply a supervised loss using these ground truth labels. For real images, domain specific losses or regularizations are applied as a form of virtual supervision for training according to the task. We apply our proposed SharinGAN to two tasks: monocular depth estimation (MDE) and face normal estimation (FNE). For MDE, we use the combination of depth smoothness and geometric consistency losses used in GASDA [52] as the virtual supervision. For FNE however, for virtual supervision we use the pseudo supervision used in SfSNet [39]. We use the term “virtual supervision” to summarize these two losses as a kind of weak supervision on the real examples.

**Monocular Depth Estimation.** To make use of ground truth labels for synthetic data, we apply \(L_1\) loss for predicted synthetic depth images:

\[
L_1 = ||\hat{y}_s - y_s^*||_1
\]

(5)

where \(\hat{y}_s\) is the predicted synthetic depth map and \(y_s^*\) is its corresponding ground truth. Following [52], we apply smoothness loss on depth \(L_{DS}\) to encourage it to be consistent with local homogeneous regions. Geometric consistency loss \(L_{GC}\) is applied so that the task network can learn the physical geometric structure through epipolar constraints. \(L_{DS}\) and \(L_{GC}\) are defined as:

\[
L_{DS} = e^{-\nabla x_r ||\nabla \hat{y}_r||}
\]

(6)

\[
L_{GC} = \eta \frac{1 - SSIM(x_r, x_r') + \mu ||x_r - x_r'||}{2}
\]

(7)

\(\hat{y}_r\) represents the predicted depth for the real image and \(\nabla\) represents the first derivative. \(x_r\) is the left image in the KITTI dataset [29]. \(x_r'\) is the inverse warped image from the right counterpart of \(x_r\) based on the predicted depth \(\hat{y}_r\). The KITTI dataset[29] provides the camera focal length and the baseline distance between the cameras. Similar to [52], we set \(\eta\) as 0.85 and \(\mu\) as 0.15 in our experiments. The overall loss for the task network is defined as:

\[
L_T = \beta_1 L_{DS} + \beta_2 L_1 + \beta_3 L_{GC},
\]

(8)

where \(\beta_1 = 0.01, \beta_2 = 0.05, \beta_3 = 100\).

**Face Normal Estimation.** SfSNet [39] currently achieves the best performance on face normal estimation. We thus follow its setup for face normal estimation and apply “SfS-supervision” for both synthetic and real images during training.

\[
L_T = \lambda_{recon} L_{recon} + \lambda_N L_N + \lambda_A L_A + \lambda_{Light} L_{Light},
\]

(9)

where \(L_{recon}, L_N\) and \(L_A\) are \(L_1\) losses on the reconstructed image, normal and albedo, whereas \(L_{Light}\) is the
L2 loss over the 27 dimensional spherical harmonic coefficients. The supervision for real images is from the "pseudo labels", obtained by applying a pre-trained task network on real images. Please refer to [39] for more details.

### 3.3. Overall loss

The overall loss used to train our geometry estimation pipeline is then defined as:

$$L = \alpha_1 L_{adv} + \alpha_2 L_r + \alpha_3 L_T.$$  

(10)

where $(\alpha_1, \alpha_2, \alpha_3) = (1, 10, 1)$ for monocular depth estimation task and $(\alpha_1, \alpha_2, \alpha_3) = (1, 10, 0.1)$ for face normal estimation task.

### 4. Experiments

We apply our proposed SharinGAN to monocular depth estimation and face normal estimation. We discuss the details of the experiments in this section.

#### 4.1. Monocular Depth Estimation

**Datasets**

Following [52], we use vKITTI [1] and KITTI [29] as synthetic and real datasets to train our network. vKITTI contains 21,260 image-depth pairs, which are all used for training. KITTI [29] provides 42,382 stereo pairs, among which, 22,600 images are used for training and 888 are used for validation as suggested by [52].

**Implementation details**

We use a generator $G$ and a primary task network $T$, whose architectures are identical to [52]. We pre-train the generative network $G$ on both synthetic and real data using reconstruction loss $L_r$. This results in an identity mapping that can help $G$ to keep as much of the input image’s geometry information as possible. Our task network is pre-trained using synthetic data with supervision. $G$ and $T$ are then trained end to end using Equation 10 for 150,000 iterations with a batch size of 2, by using an Adam optimizer with a learning rate of $1e-5$. The best model is selected based on the validation set of KITTI.

**Results**

Table 1 shows the quantitative results on the eigen test split of the KITTI dataset for different methods on the MDE task. The proposed method outperforms the previous unsupervised domain adaptation methods for MDE [52, 53] on almost all the metrics. Especially, compared with [52], we reduce the absolute error by 19.7% and 21.0% on 80m cap and 50m cap settings respectively. Moreover, the performance of our method is much closer to the methods in a supervised setting [8, 22, 19], which was trained on the real KITTI dataset with ground truth depth labels. Figure 3 visually compares the predicted depth map from the proposed method with [52]. We show three typical examples: near distance, medium distance, and far distance. It shows that our proposed method performs much better for predicting depth at details. For instance, our predicted depth map can better preserve the shape of the car (Figure 3 (a) and (c)) and the structure of the tree and the building behind it (Figure 3 (b)). This shows the advantage of our proposed SharinGAN compared with [52]. [52] learns to transfer real images to the synthetic domain and vice versa, which solves a much harder problem compared with SharinGAN, which removes a minimum of domain specific information. As a result, the quality of the transformation for [52] may not be as good as the proposed method. Moreover, the unsupervised transformation cannot guarantee to keep the geometry information unchanged.

To understand how our generative network $G$ works, we show some examples of synthetic and real images, their transformed versions, and the difference images in Figure 4. This shows that $G$ mainly operates on edges. Since depth

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervised</th>
<th>Dataset</th>
<th>Cap</th>
<th>Error Metrics, lower is better</th>
<th>Accuracy Metrics, higher is better</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Abs Rel</td>
<td>Sq Rel</td>
</tr>
<tr>
<td>Eigen et al. [8]</td>
<td>Yes</td>
<td>K</td>
<td>80m</td>
<td>0.203</td>
<td>1.548</td>
</tr>
<tr>
<td>Liu et al. [22]</td>
<td>Yes</td>
<td>K</td>
<td>80m</td>
<td>0.202</td>
<td>1.614</td>
</tr>
<tr>
<td>All synthetic (baseline)</td>
<td>No</td>
<td>S</td>
<td>80m</td>
<td>0.253</td>
<td>2.303</td>
</tr>
<tr>
<td>All real (baseline)</td>
<td>No</td>
<td>K</td>
<td>80m</td>
<td>0.158</td>
<td>1.151</td>
</tr>
<tr>
<td>GASDA [52]</td>
<td>No</td>
<td>K+S</td>
<td>80m</td>
<td>0.149</td>
<td>1.003</td>
</tr>
<tr>
<td>SharinGAN (proposed)</td>
<td>No</td>
<td>K+S</td>
<td>80m</td>
<td>0.116</td>
<td>0.939</td>
</tr>
<tr>
<td>Kuznetsov et al. [19]</td>
<td>Yes</td>
<td>K</td>
<td>50m</td>
<td>0.117</td>
<td>0.597</td>
</tr>
<tr>
<td>Garg et al. [9]</td>
<td>No</td>
<td>K</td>
<td>50m</td>
<td>0.169</td>
<td>1.080</td>
</tr>
<tr>
<td>Godard et al. [11]</td>
<td>No</td>
<td>K</td>
<td>50m</td>
<td>0.140</td>
<td>0.976</td>
</tr>
<tr>
<td>All synthetic (baseline)</td>
<td>No</td>
<td>S</td>
<td>50m</td>
<td>0.244</td>
<td>1.771</td>
</tr>
<tr>
<td>All real (baseline)</td>
<td>No</td>
<td>K</td>
<td>50m</td>
<td>0.151</td>
<td>0.856</td>
</tr>
<tr>
<td>Kundu et al. [30]</td>
<td>No</td>
<td>K+S</td>
<td>50m</td>
<td>0.203</td>
<td>1.734</td>
</tr>
<tr>
<td>T2Net [53]</td>
<td>No</td>
<td>K+S</td>
<td>50m</td>
<td>0.168</td>
<td>1.199</td>
</tr>
<tr>
<td>GASDA [52]</td>
<td>No</td>
<td>K+S</td>
<td>50m</td>
<td>0.145</td>
<td>0.756</td>
</tr>
<tr>
<td>SharinGAN (proposed)</td>
<td>No</td>
<td>K+S</td>
<td>50m</td>
<td>0.109</td>
<td>0.673</td>
</tr>
</tbody>
</table>

Table 1: MDE Results on eigen test split of KITTI dataset [8]. For the training data, K: KITTI dataset and S: vKITTI dataset. Methods highlighted in light gray, use domain adaptation techniques and the non-highlighted rows correspond to supervised methods.
Figure 3: Qualitative comparisons of SharinGAN with GASDA [52]. Ground truth (GT) has been interpolated for visualization. We mask out the top regions where ground truth depth is not available for visualization purposes. Note that in addition to various other aspects mentioned above, we are also able to remove the boundary artifacts present in the depth maps of GASDA.

Figure 4: (a), (b) and (c) show real image $x_r$, translated real image $x_{r}^{sh}$ and their difference $|x_r - x_{r}^{sh}|$ respectively. (d), (e) and (f) show synthetic image $x_s$, translated synthetic image $x_{s}^{sh}$ and their difference $|x_s - x_{s}^{sh}|$ respectively.
maps are mostly discontinuous at edges, they provide important cues for the geometry of the scene. On the other hand, due to the difference between the geometry and material of objects around the edges, the rendering algorithm may find it hard to render realistic edges compared with other parts of the scene. As a result, most of the domain specific information related to geometry lies in the edges, on which SharinGAN correctly focuses.

4.1.1 Generalization to Make3D

To demonstrate the generalization ability of the proposed method, we test our trained model on Make3D [37]. Note that we do not fine-tune our model using the data from Make3D. Table 2 shows the quantitative results of our method, which outperforms existing state-of-the-art methods by a large margin. Moreover, the performance of SharinGAN is more comparable to the supervised methods. We further visually compare the proposed method with GASDA [52] in Figure 5. It is clear that the proposed depth map captures more details in the input images, reflecting more accurate depth prediction.

**4.2. Face Normal Estimation**

**Datasets** We use the synthetic data provided by [39] and CelebA [26] as real data to train the SharinGAN for face normal estimation similar to [39]. Our trained model is then evaluated on the Photoface dataset [51].

**Implementation details** We use the RBDN network [35] as our generator and SfSNet [39] as the primary task network. Similar to before, we pre-train the Generator on both synthetic and real data using reconstruction loss and pre-train the primary task network on just synthetic data in a supervised manner. Then, we train $G$ and $T$ end-to-end using the overall loss (10) for 120,000 iterations. We use a batch size of 16 and a learning rate of $1e^{-4}$. The best model is selected based on the validation set of Photoface[51].

**Results** Table 4 shows the quantitative performance of the estimated surface normals by our method on the test split of the Photoface dataset. With the proposed SharinGAN module, we were able to significantly improve over SfSNet on all the metrics. In particular, we were able to significantly reduce the mean angular error metric by roughly $1.5^\circ$.

Additionally, Figure 6 depicts the qualitative comparison of our method with SfSNet on the test split of Photoface. Both SfSNet and our pipeline are not finetuned on this dataset, and yet we were able to generalize better compared to SfSNet. This demonstrates the generalization capacity of the proposed SharinGAN to unseen data in training.

<table>
<thead>
<tr>
<th>Method</th>
<th>Trained</th>
<th>Abs Rel</th>
<th>Sq Rel</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karsh et al. [17]</td>
<td>Yes</td>
<td>0.398</td>
<td>4.723</td>
<td>7.801</td>
</tr>
<tr>
<td>Laina et al. [20]</td>
<td>Yes</td>
<td>0.198</td>
<td>1.665</td>
<td>5.461</td>
</tr>
<tr>
<td>Kundu et al. [30]</td>
<td>No</td>
<td>0.452</td>
<td>5.71</td>
<td>9.559</td>
</tr>
<tr>
<td>Goddard et al. [12]</td>
<td>No</td>
<td>0.505</td>
<td>10.172</td>
<td>10.936</td>
</tr>
<tr>
<td>Kundu et al. [30]</td>
<td>No</td>
<td>0.647</td>
<td>12.341</td>
<td>11.567</td>
</tr>
<tr>
<td>Atapour et al. [3]</td>
<td>No</td>
<td>0.423</td>
<td>9.343</td>
<td>9.002</td>
</tr>
<tr>
<td>T2Net [53]</td>
<td>No</td>
<td>0.508</td>
<td>6.589</td>
<td>8.935</td>
</tr>
<tr>
<td>GASDA [52]</td>
<td>No</td>
<td>0.403</td>
<td>6.709</td>
<td>10.424</td>
</tr>
<tr>
<td>SharinGAN (proposed)</td>
<td>No</td>
<td>0.377</td>
<td>4.900</td>
<td>8.388</td>
</tr>
</tbody>
</table>

Table 2: MDE results on Make3D dataset [37]. Trained indicates whether the model is trained on Make3D or not. Errors are computed for depths less than 70m in a central image crop [12]. It can be concluded that our proposed method generalized better to an unseen dataset.
### Table 3: Ablation study for monocular depth estimation to understand the role of the SharinGAN module and Reconstruction loss. We need both to get the best performance for this task.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAE</th>
<th>&lt; 20°</th>
<th>&lt; 25°</th>
<th>&lt; 30°</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DMM [4]</td>
<td>26.3°</td>
<td>4.3%</td>
<td>56.1%</td>
<td>89.4%</td>
</tr>
<tr>
<td>Pix2Vertex [38]</td>
<td>33.9°</td>
<td>24.8%</td>
<td>36.1%</td>
<td>47.6%</td>
</tr>
<tr>
<td>SfSNet [39]</td>
<td>25.5°</td>
<td>43.6%</td>
<td>57.7%</td>
<td>68.7%</td>
</tr>
<tr>
<td>SharinGAN (proposed)</td>
<td>24.0°</td>
<td>47.8%</td>
<td>61.53%</td>
<td>72.1%</td>
</tr>
</tbody>
</table>

Table 4: Quantitative results for Face Normal estimation on the test split of Photoface dataset [51]. All the listed methods are not fine-tuned on Photoface. The metrics MAE: Mean Angular Error and < 20°, 25°, 30° refer to the normals prediction accuracy for different thresholds.

<table>
<thead>
<tr>
<th>Components</th>
<th>Error Metrics, lower is better</th>
<th>Accuracy Metrics, higher is better</th>
</tr>
</thead>
<tbody>
<tr>
<td>SharinGAN</td>
<td>Cap</td>
<td>Abs Rel</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>50m</td>
</tr>
<tr>
<td>✓</td>
<td>x</td>
<td>50m</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>50m</td>
</tr>
</tbody>
</table>

Table 5: Ablation study for monocular depth estimation to understand the role of the SharinGAN module and Reconstruction loss on the Make3D test dataset. We need both to get the best performance for this task.

6. Conclusion

Our primary motivation is to simplify the process of combining synthetic and real images in training. Prior approaches often pick one domain and try to map images into it from the other domain. Instead, we train a generator to map all images into a new, shared domain. In doing this, we note that in the new domain, the images need not be indistinguishable to the human eye, only to the network that performs the primary task. The primary network will learn to ignore extraneous, domain-specific information that is retained in the shared domain.

To achieve this, we propose a simple network architecture that rests on our new SharinGAN, which maps both real and synthetic images to a shared domain. The resulting images retain domain-specific details that do not prevent the primary network from effectively combining training data from both domains. We demonstrate this by achieving significant improvements over state-of-the-art approaches in two important applications, surface normal estimation for faces, and monocular depth estimation for outdoor scenes.

Finally, our ablation studies demonstrate the significance of the proposed SharinGAN in effectively combining synthetic and real data.
References


[23] Feng Liu, Luan Tran, and Xiaoming Liu. 3d face modeling from diverse raw scan data. In ICCV, 2019.


