SharinGAN: Combining Synthetic and Real Data for Unsupervised Geometry Estimation

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

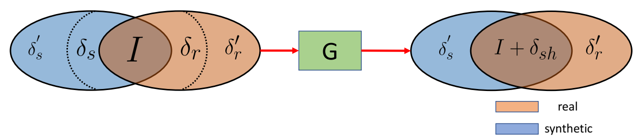


Figure 1: We propose to reduce the domain gap between synthetic and real by mapping the corresponding domain specific information related to the primary task (δ_s, δ_r) into shared information δ_{sh}, preserving everything else.

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 δ'_s and δ'_r for synthetic and real images respectively) and domain specific information related to the primary task (δ_s, δ_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., δ'_r. On the other hand, shad-

*Hao Zhou is currently at Amazon AWS.

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 δ_s and δ_r to a shared representation, δ_{sh}, while not modifying δ'_s and δ'_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 δ_s and δ_r, mapping them into a shared domain δ_{sh}. G will leave I unchanged. δ'_s and δ'_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 the test set of Make3D [37] by 23.77% and 6.45% respectively compared with the state-of-the-art method [52]. Additionally, our proposed method improves over SfSNet [39] on face normal estimation. It yields an accuracy boost of nearly 4.3% for normal prediction within $20°$ $(Acc < 20°)$ 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. T^{2}Net [53] trained a network that translates synthetic data into real at the image level 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 is complementary to these self-supervised methods, it does not require this additional data, but can use it when available.

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

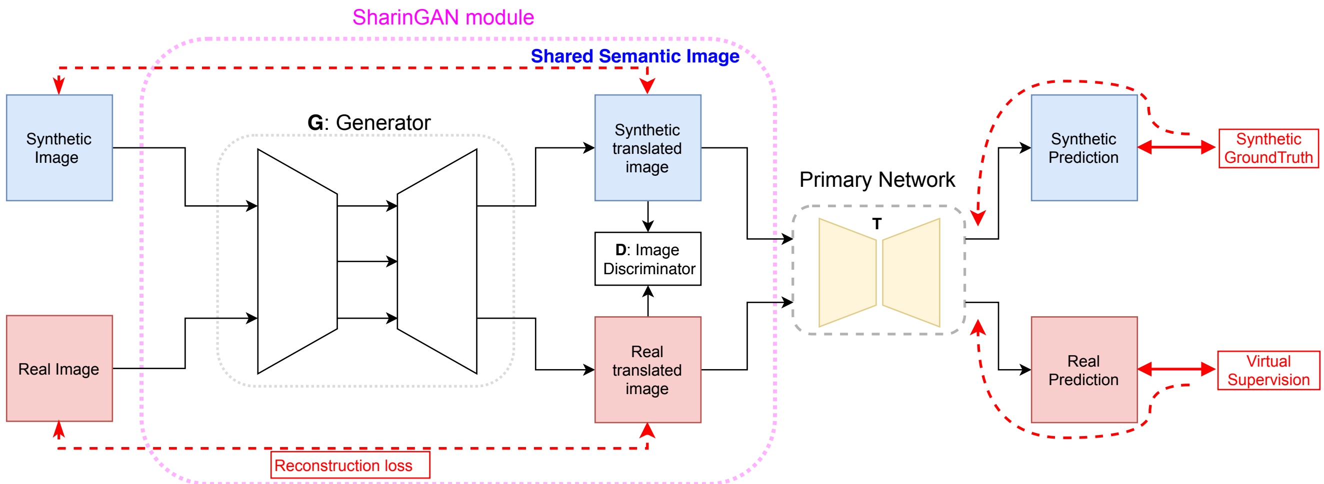


Figure 2: Overview of the model architecture. Red dashed arrows indicate the loss computations.

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 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 use this to 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 $G : x_r \mapsto x_r^{sh}$ and $G : x_s \mapsto x_s^{sh}$, 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, δ_{sh}. This is done by minimizing the discrepancy in the distributions of x_r^{sh} and x_s^{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 δ_{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 δ_s and δ_r of synthetic and real images to a shared space δ_{sh}, leaving δ'_s and δ'_r unchanged. As a result, our transformed synthetic and real images (x_s^{sh} 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 δ_s and δ_r to a shared space δ_{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.:

$$L_W(D, G) = \mathbb{E}_{x_s}[D(G(x_s))] - \mathbb{E}_{x_r}[D(G(x_r))], \tag{1}$$

D is a discriminator and G_e 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:

$$L_{gp}(D) \quad = \quad (||\nabla_{\hat{h}} D(\hat{h})||_2 - 1)^2 \tag{2}$$

Our overall adversarial loss is then defined as:

$$L_{adv} \quad = \quad L_W(D, G) - \lambda L_{gp}(D) \tag{3}$$

where λ 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 δ and δ' 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:

$$L_r = ||G(x_s) - x_s||_2^2 + ||G(x_r) - x_r||_2^2. \tag{4}$$

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:

$$L_1 = ||\hat{y}_s - y_s^*||_1 \tag{5}$$

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:

$$L_{DS} = e^{-\nabla x_r} ||\nabla \hat{y_r}|| \tag{6}$$

$$L_{GC} = \eta \frac{1 - SSIM(x_r, x'_{rr})}{2} + \mu ||x_r - x'_{rr}||, \tag{7}$$

$\hat{y}_r$ represents the predicted depth for the real image and ∇ represents the first derivative. x_r is the left image in the KITTI dataset [29]. x'_{rr} 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 η as 0.85 and μ as 0.15 in our experiments. The overall loss for the task network is defined as:

$$L_T = \beta_1 L_{DS} + \beta_2 L_1 + \beta_3 L_{GC}, \tag{8}$$

where $\beta_1 = 0.01, \beta_2 = \beta 3 = 100$.

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.

$$L_T = \lambda_{recon} L_{recon} + \lambda_N L_N + \lambda_A L_A + \lambda_{Light} L_{Light}, \tag{9}$$

where L_{recon}, L_N and L_A are L_1 losses on the reconstructed image, normal and albedo, whereas L_{light} is the

Method	Supervised	Dataset	Cap	Error Metrics, lower is better				Accuracy Metrics, higher is better		
				Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al*. [8]	Yes	K	80m	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al*. [22]	Yes	K	80m	0.202	1.614	6.523	0.275	0.678	0.895	0.965
All synthetic (baseline)	No	S	80m	0.253	2.303	6.953	0.328	0.635	0.856	0.937
All real (baseline)	No	K	80m	0.158	1.151	5.285	0.238	0.811	0.934	0.970
GASDA [52]	No	K+S	80m	0.149	1.003	**4.995**	0.227	0.824	0.941	0.973
SharinGAN (proposed)	No	K+S	80m	**0.116**	**0.939**	5.068	**0.203**	**0.850**	**0.948**	**0.978**
Kuznietsov *et al*. [19]	Yes	K	50m	0.117	0.597	3.531	0.183	0.861	0.964	0.989
Garg *et al*. [9]	No	K	50m	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard *et al*. [11]	No	K	50m	0.140	0.976	4.471	0.232	0.818	0.931	0.969
All synthetic (baseline)	No	S	50m	0.244	1.771	5.354	0.313	0.647	0.866	0.943
All real (baseline)	No	K	50m	0.151	0.856	4.043	0.227	0.824	0.940	0.973
Kundu *et al*. [30]	No	K+S	50m	0.203	1.734	6.251	0.284	0.687	0.899	0.958
T2Net [53]	No	K+S	50m	0.168	1.199	4.674	0.243	0.772	0.912	0.966
GASDA [52]	No	K+S	50m	0.143	0.756	3.846	0.217	0.836	0.946	0.976
SharinGAN (proposed)	No	K+S	50m	**0.109**	**0.673**	**3.77**	**0.190**	**0.864**	**0.954**	**0.981**

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.

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:

$$L = \alpha_1 L_{adv} + \alpha_2 L_r + \alpha_3 L_T. \tag{10}$$

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.

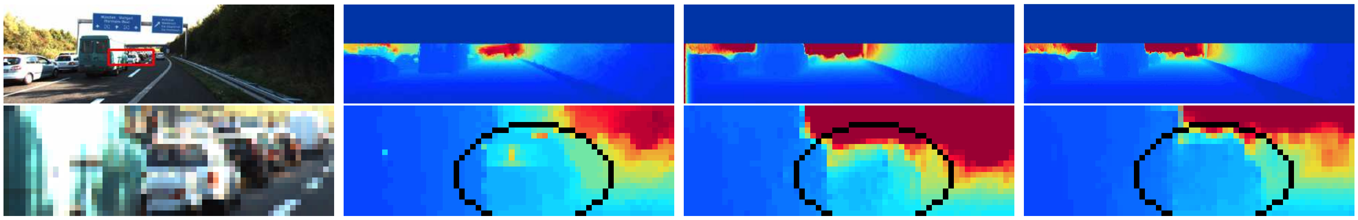
4.1. Monocular Depth Estimation

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].

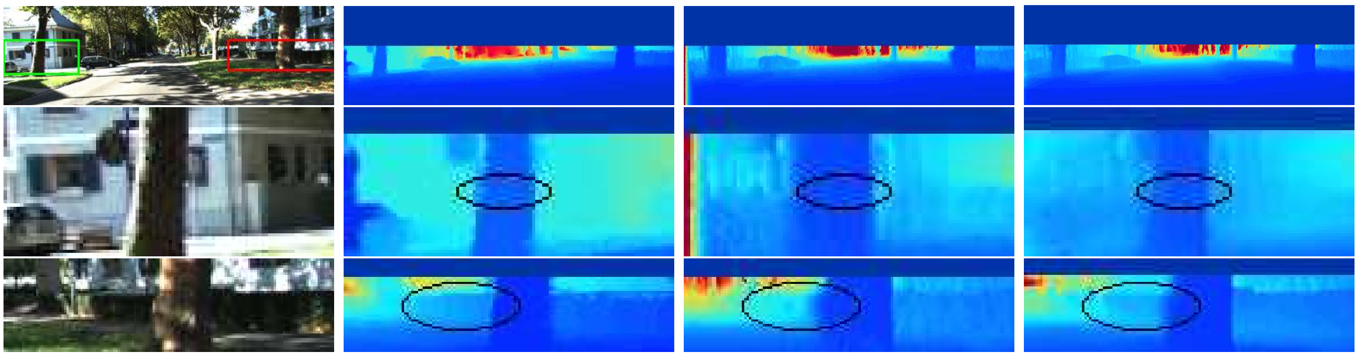
Implementation details We use a generator G and a primary task network T, whose architectures are identical to [52]. We pre-train the generative network G on both synthetic and real data using reconstruction loss L_r. This results in an identity mapping that can help G to keep as much of the input image's 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.

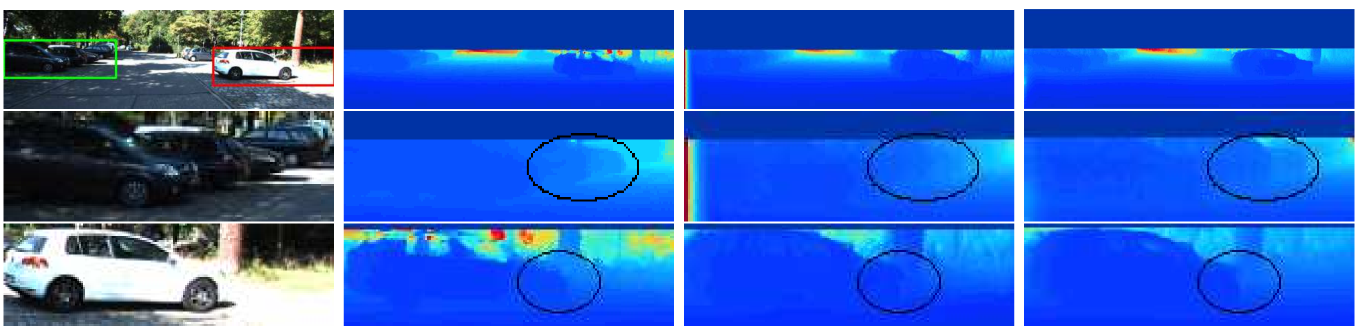
To understand how our generative network G works, we show some examples of synthetic and real images, their transformed versions, and the difference images in Figure 4. This shows that G mainly operates on edges. Since depth



(a) First row from left to right: real image, ground truth depth map, depth map by GASDA [52] and depth map by SharinGAN. The second row shows the corresponding region in the red box of the first row. The depth of the faraway car is better estimated by SharinGAN than GASDA.

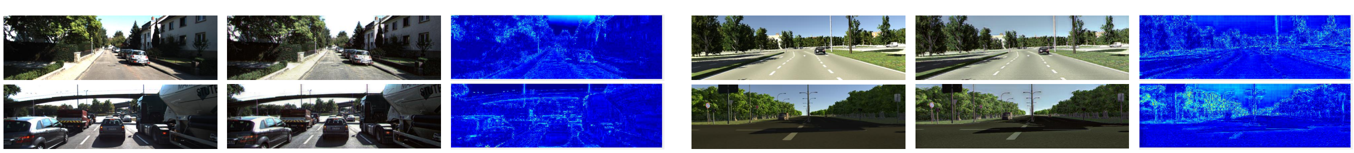


(b) First row from left to right: real image, ground truth depth map, depth map by GASDA [52] and depth map by SharinGAN. The second and third row shows the corresponding region in the green and red box of the first row. The depth of the tree to the left (green) and shrubs behind the tree in the right are better estimated by SharinGAN.



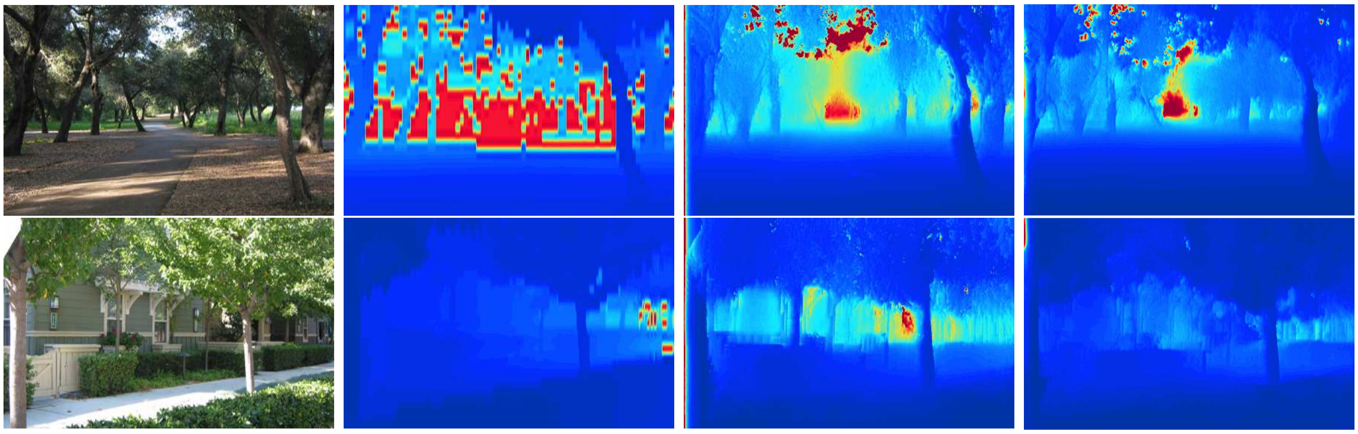
(c) First row from left to right: real image, ground truth depth map, depth map by GASDA [52] and depth map by SharinGAN. The second and third row shows the corresponding region in the green and red box of the first row. The boundaries and the depth of the cars are better estimated by SharinGAN.

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.



(a) x_r (b) $x_r^{sh} = G(x_r)$ (c) $|x_r - x_r^{sh}|$ (d) x_s (e) $x_s^{sh} = G(x_s)$ (f) $|x_s - x_s^{sh}|$

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.



(a) Input Image (b) Ground Truth (c) GASDA[52] (d) SharinGAN

Figure 5: Qualitative results on the test set of the Make3D dataset [37]. In the top row, some far tree structures that are missing in the depth map predicted by GASDA were better captured on using the SharinGAN module. For the bottom row, GASDA wrongly predicts the depth map of the houses behind the trees to be far, which is correctly captured by the SharinGAN.

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

Method	Trained	Error Metrics, lower is better		
		Abs Rel	Sq Rel	RMSE
Karsh et al. [17]	Yes	0.398	4.723	7.801
Laina et al. [20]	Yes	0.198	1.665	5.461
Kundu et al. [30]	Yes	0.452	5.71	9.559
Goddard et al. [12]	No	0.505	10.172	10.936
Kundu et al. [30]	No	0.647	12.341	11.567
Atapour et al. [3]	No	0.423	9.343	9.002
T2Net [53]	No	0.508	6.589	8.935
GASDA [52]	No	0.403	6.709	10.424
SharinGAN (proposed)	No	**0.377**	**4.900**	**8.388**

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.

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 $1.5°$.

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.

Components		Cap	Error Metrics, lower is better				Accuracy Metrics, higher is better		
SharinGAN	Reconstruction loss		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
x	x	50m	0.137	0.804	4.12	0.210	0.816	0.940	0.978
✓	x	50m	0.1113	**0.6705**	3.80	0.192	0.861	0.954	0.980
✓	✓	50m	**0.109**	0.673	**3.77**	**0.190**	**0.864**	**0.954**	**0.981**

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.

Algorithm	MAE	$< 20°$	$< 25°$	$< 30°$
3DMM [4]	26.3°	4.3%	56.1%	**89.4%**
Pix2Vertex [38]	33.9°	24.8%	36.1%	47.6%
SfSNet[39]	25.5°	43.6%	57.7%	68.7%
SharinGAN (proposed)	**24.0°**	**47.88**%	**61.53**%	72.1%

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.

(a) Input Image (b) GT (c) SfSNet[39] (d) SharinGAN

Figure 6: Qualitative comparisons of our method with SfSNet on the examples from the test set of Photoface dataset [51]. Our method generalizes much better to unseen data during training.

5. Ablation studies

We carried out our ablation study using the KITTI and Make3D datasets on monocular depth estimation. We study the role of the SharinGAN module by removing it and training a primary network on the original synthetic and real data using (8). We observe that the performance drops significantly as shown in Table 3 and Table 5. This shows the importance of the SharinGAN module that helps train the primary task network efficiently.

To demonstrate the role of reconstruction loss, we remove it and train our whole pipeline $\alpha_1 L_{adv} + \alpha_3 L_T$. We show the results on the testset of KITTI in the second row of Table 3 and on the testset of Make3D in the second row of Table 5. For both the testsets, we can see the performance drop compared to our full model. Although the drop is smaller in the case of KITTI, it can be seen that the drop is significant for Make3D dataset that is unseen during training. This signifies the importance of reconstruction loss to generalize well to a domain not seen during training.

Components		Cap	Error Metrics, lower is better		
SharinGAN	Reconstruction loss		Abs Rel	Sq Rel	RMSE
x	x	70m	0.476	8.058	9.449
✓	x	70m	0.401	5.318	**8.377**
✓	✓	70m	**0.377**	**4.900**	8.388

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

[1] Yohann Cabon Eleonora Vig Adrien Gaidon, Qiao Wang. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *NeurIPS*, 2017.

[3] Amir Atapour-Abarghouei and Toby P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *CVPR*, June 2018.

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999.

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] Vincent Casser, Sren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019.

[7] David Eigen, , and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*. 2014.

[9] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[10] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression. *CVPR*, 2018.

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[12] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[13] Clement Godard, Mac Aodha Oisin, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*. 2014.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*. 2017.

[16] Lei He, Guanghui Wang, and Zhanyi Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Trans. on Image Processing*, 27(9), 2018.

[17] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. PAMI*, 36(11):2144–2158, 2014.

[18] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ToG*, 33(3), 2014.

[19] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, July 2017.

[20] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248, 2016.

[21] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[22] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. on PAMI*, 38(10), 2016.

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

[24] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *CVPR*, June 2014.

[25] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NeurIPS*. 2016.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[27] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, 2019.

[28] Ishit Mehta, Parikshit Sakurikar, and P. Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *3DV*, 2018.

[29] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[30] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *CVPR*, 2018.

[31] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, 2018.

[32] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.

[33] Vamshi Repala and Shiv Ram Dubey. Dual cnn models for unsupervised monocular depth estimation. 04 2018.

[34] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[35] Venkataraman Santhanam, Vlad I. Morariu, and Larry S. Davis. Generalized deep image to image regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *NeurIPS*, 2006.

[37] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. PAMI*, 31(5):824–840, 2009.

[38] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *ICCV*, 2017.

[39] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *CVPR*, 2018.

[40] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation, 2017.

[41] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[42] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

[43] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[44] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *ICCV*, 2017.

[45] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphoable model. In *CVPR*, 2019.

[46] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *CVPR*, 2018.

[47] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[48] Chaoyang Wang, Jos Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[49] Y. Wang, L. Zhang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Trans. on PAMI*, 31(11):1968–1984, 2009.

[50] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018.

[51] Stefanos Zafeiriou, Mark F. Hansen, Gary A. Atkinson, Vasileios Argyriou, Maria Petrou, Melvyn L. Smith, and Lyndon N. Smith. The photoface database. In *CVPR Workshops*, 2011.

[52] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *CVPR*, 2019.

[53] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *ECCV*, 2018.

[54] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W. Jacobs. Deep single portrait image relighting. In *ICCV*, 2019.

[55] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[56] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.