

# Real-Time Monocular Depth Estimation using Synthetic Data with Domain Adaptation via Image Style Transfer

Amir Atapour-Abarghouei<sup>1</sup> Toby P. Breckon<sup>1,2</sup>

<sup>1</sup>Department of Computer Science – <sup>2</sup>Department of Engineering  
Durham University, UK

{amir.atapour-abarghouei,toby.breckon}@durham.ac.uk

## Abstract

Monocular depth estimation using learning-based approaches has become promising in recent years. However, most monocular depth estimators either need to rely on large quantities of ground truth depth data, which is extremely expensive and difficult to obtain, or predict disparity as an intermediary step using a secondary supervisory signal leading to blurring and other artefacts. Training a depth estimation model using pixel-perfect synthetic data can resolve most of these issues but introduces the problem of domain bias. This is the inability to apply a model trained on synthetic data to real-world scenarios. With advances in image style transfer and its connections with domain adaptation (Maximum Mean Discrepancy), we take advantage of style transfer and adversarial training to predict pixel perfect depth from a single real-world color image based on training over a large corpus of synthetic environment data. Experimental results indicate the efficacy of our approach compared to contemporary state-of-the-art techniques.

## 1. Introduction

As 3D imagery has become the staple requirement within many computer vision applications, accurate and efficient depth estimation is now one of its core foundations. Conventional depth estimation methods have relied on numerous strategies such as stereo correspondence [67, 28], structure from motion [14, 9], depth from shading and light diffusion [73, 82, 1] and alike. However, these approaches are often rife with issues such as depth inhomogeneity, missing depth (holes), computationally intensive requirements and more importantly, careful calibration and setup demanding expert knowledge which often requires special post-processing [4, 2, 49, 58].

A solution to many of these challenges is monocular depth estimation. Over the past few years, research into predicting depth from a single image has significantly escalated [39, 48, 17, 26, 22, 83]. A number of supervised



Figure 1: Our monocular depth estimation (KITTI [55]).

learning approaches have recently emerged that take advantage of off-line training on ground truth depth data to make monocular depth prediction possible [39, 48, 18, 17, 42, 91]. However, since ground truth depth is extremely difficult and expensive to acquire in the real world, when it is obtained it is often sparse and flawed, constraining the practical use of many of these approaches.

Other monocular approaches, sometimes referred to as *unsupervised*, do not require direct ground truth depth, but instead utilize a secondary supervisory signal during training which indirectly results in producing the desired depth [26, 22, 83, 12]. Training data for these approaches is abundant and easily obtainable but they suffer from undesirable artefacts, such as blurring and incoherent content, due to the nature of their secondary supervision.

However, an often overlooked fact is that the same technology that facilitates training large-scale deep neural networks can also assist in acquiring synthetic data for these neural networks [64, 69]. Nearly photorealistic graphically rendered environments primarily used for gaming can be used to capture homogeneous synthetic depth maps which are then utilized in training a depth estimating model.

While the use of synthetic data is not novel [41, 61, 19, 69], domain adaptation has always been the greatest challenge in this area. Stated precisely, the problem is that: *A model trained on data from one domain is often incapable of performing well on data from another domain due to distinctions in the intrinsic nature of these two domains.*

Here, we explore the possibility of training a depth estimation model on synthetic data using the new findings re-

garding the connection between style transfer and domain adaptation [47]. Our contributions are thus as follows:

- *synthetic depth prediction* - a directly supervised model using a light-weight architecture with skip connections that can predict depth based on high quality synthetic depth training data (Section 3.1).
- *domain adaptation via style transfer* - a solution to the issue of domain bias via style transfer (Section 3.2).
- *efficacy* - an efficient and novel approach to monocular depth estimation that produces pixel-perfect depth.
- *reproducibility* - simple and effective algorithm relying on data that is easily and openly obtained.

## 2. Related Work

We consider prior work within three distinct domains: monocular depth estimation (Section 2.1), domain adaptation (Section 2.2), and image style transfer (Section 2.3).

### 2.1. Monocular Depth Estimation

There have been great strides made in the field of monocular depth estimation based on directly supervised training, and many existing approaches produce promising results.

The work in [65] utilizes a Markov Random Field (MRF) and linear regression to estimate depth, which is later extended into Make3D [66] with the MRF combining planes predicted by the linear model to describe the 3D position and orientation of segmented patches within RGB images. Since depth is predicted locally, the combined output lacks global coherence. Additionally, the model is manually tuned which is a detriment against achieving a true learning system. The tuning is subsequently performed using a convolutional neural network (CNN) in [48]. Later on, [39] utilizes semantic labels to train classifiers at chosen depths, which are subsequently used to predict depth.

Global scene depth prediction has also seen significant progress. The method in [6] employs sparse coding to estimate entire scene depth. Similarly, [18, 17] uses a two-scale network trained on RGB and depth to produce depth. Since then, numerous improvements have been made to achieve better directly supervised training for monocular depth estimation [43, 80, 40, 8]. However, due to the scarcity of high quality ground truth depth, these approaches have to make do with either smaller number of images or lower quality data, and as such any supervised learning approach cannot produce results better than the limits of its training data.

More recently, a new class of monocular depth estimators have emerged that do not require ground truth depth and calculate disparity by reconstructing the corresponding view within a stereo correspondence framework. The work in [83] proposes the Deep3D network, which learns to generate the right view from the left image used as the input, and in the process produces an intermediary disparity map. While results are promising, the method is very memory intensive. The approach in [22] follows a similar framework

with a model that is not fully differentiable. On the other hand, [26] uses bilinear sampling [33] and a left/right consistency check incorporated into training for better results.

While these approaches produce better and more consistent results than the directly supervised methods, there are shortcomings. Firstly, the training data must consist of temporally aligned and rectified stereo images, and more importantly, in the presence of occluded regions (*i.e.* groups of pixels that are seen in one image but not the other), disparity calculations fail and meaningless values are generated.

The work in [88] estimates depth and camera motion from video by training depth and pose prediction networks, indirectly supervised via view synthesis. The results are favorable especially since they include ego-motion but the depth outputs are blurry, do not consider occlusions and are dependent on camera parameters. The training in the work of [38] is supervised by sparse ground truth depth and the model is then enforced within a stereo framework via an image alignment loss to output dense depth.

Since our model is trained on synthetic images, there is an abundance of training data, and as there is no need for a secondary supervisory signal, complete depth is obtainable free from any unwanted artefacts. As a result, our approach does not suffer from the aforementioned limitations

### 2.2. Domain Adaptation

In this work, our depth estimation model is trained on a *synthetically generated* dataset of corresponding RGB and depth images to learn the context and content of the scene and predict depth. However, due to *dataset bias* [59], a typical model trained on a specific set of data does not necessarily generalize well to other datasets. In other words, a model trained on *synthetic* data may not perform well on *real-world* data. Therefore, while our depth estimation model may successfully predict the depth for synthetic data, it will not be able to do the same for naturally obtained images, which would make the model utterly useless from a practical visual sensing perspective.

While the typical solution to this data domain variation problem is to fine-tune the network on the target data (in our case, real-world images), fitting the large number of parameters in a deep network to a new dataset requires a large amount of data, which can be very time-consuming, expensive, or even practically intractable to collate in our case giving rise to the use of synthetic data instead. Given that the objective is to employ a model trained on the source dataset to successfully perform on a target dataset, one strategy is to minimize the distance between the source and target feature distributions [52, 25, 20, 74, 15, 21, 75].

Some approaches have taken advantage of Maximum Mean Discrepancy (MMD) which calculates the norm of the distance between the domains to reduce the discrepancy [52, 76, 72], while others have taken to using an adversarial

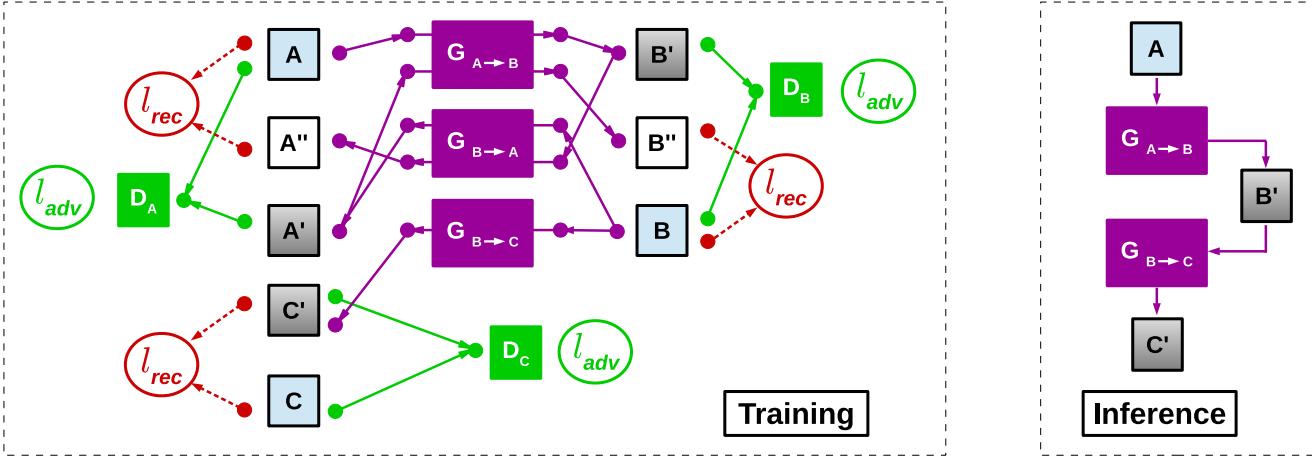


Figure 2: Our approach using [90]. Domain  $A$  (real-world RGB) is transformed into  $B$  (synthetic RGB) and then to  $C$  (pixel-perfect depth).  $A, B, C$  denote ground truth,  $A', B', C'$  generated images, and  $A'', B''$  cyclically regenerated images.

loss which leads to a representation that minimizes the domain discrepancy while able to discriminate the source labels easily [25, 20, 74, 75]. Although most of the these techniques focus on discriminative models, research on generative tasks has also utilized domain adaptation [15, 51].

Recently [47] proposed that matching the Gram matrices [68] of feature maps, often performed within neural style transfer of images, is theoretically equivalent to minimizing the maximum mean discrepancy with the second order polynomial kernel. In the following section, we briefly review neural style transfer and its relevance to this work.

### 2.3. Image Style Transfer

Image style transfer by means of convolutional neural networks has recently become noted via [23]. Since then, numerous improved and novel approaches have been proposed that can transfer the style of one image onto another.

Some methods transfer style by directly updating the pixels in the output image (often initialized with random noise) [24, 87, 13, 44]. Others improve efficiency by avoiding the direct manipulation of the image and pre-training a model using large amounts of training data for a specific image style [35, 77, 45, 11, 90]. Most approaches utilize Gram matrices to capture the style of a image [23, 24, 87, 13], while some utilize an MRF framework to manipulate image patches in order to match the desired style [44, 10].

As demonstrated in [47], style transfer can be considered as a distribution alignment process from the content image to the style image [34]. In other words, transferring the style of one image (from the source domain) to another image (from the target domain) is essentially the same as minimizing the distance between the source and target distributions (for a more in-depth theoretical analysis, readers are referred to [47]). In this work, we take advantage of this idea to adapt our data distribution (*i.e.* real-world images) to our depth estimation model trained on data from a different

distribution (*i.e.* synthetic images). In the next section, this proposed approach is outlined in greater depth.

## 3. Proposed Approach

Our approach consists of two stages, the operations of which are carried out by two separate models, trained at the same time. The first stage includes training a depth estimation model over synthetic data captured from a graphically rendered environment used for gaming applications [64] (Section 3.1). However, as the eventual goal involves real-world images, we attempt to reduce the domain discrepancy between the synthetic data distribution and the real-world data distribution using a model trained to transfer the style of synthetic images to real-world images in the second stage (Section 3.2).

### 3.1. Stage 1 - Monocular Depth Estimation Model

We treat monocular depth estimation as an image-to-image mapping problem, with the RGB image used as the input to our mapping function, which produces depth as its output. With the advent of convolutional neural networks, image-to-image translation and prediction problems have become significantly more tractable. A naive solution would be utilizing a network that minimizes a reconstruction loss (Euclidean distance) between the pixel values of the network output and the ground truth. However, due to the inherent multi-modality of the monocular depth estimation problem (several plausible depth maps can correspond with a single RGB view), any model trained to predict depth based on a sole reconstruction loss ( $\ell_1$  or  $\ell_2$ ) tends to generate values that are the average of all the possible modes in the predictions. This results in blurry outputs.

For this reason, many prediction-based approaches [57, 85, 84, 32, 46, 90] and other generative models [16, 78] leverage adversarial training [27] since the use of an ad-

Method	Training Data	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Train Set Mean	K	0.403	0.530	8.709	0.403	0.593	0.776	0.878
Eigen <i>et al.</i> Coarse	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen <i>et al.</i> Fine	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu <i>et al.</i>	K	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou <i>et al.</i>	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou <i>et al.</i>	K+CS	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Godard <i>et al.</i>	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Godard <i>et al.</i>	K+CS	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Our Approach	K+S*	<b>0.110</b>	<b>0.929</b>	<b>4.726</b>	<b>0.194</b>	<b>0.923</b>	<b>0.967</b>	<b>0.984</b>

Table 1: Comparing the results of our approach against other approaches over the KITTI dataset using the data split in Eigen *et al.*, 2014. For the training data, K represents KITTI, CS is Cityscapes, and S\* is our captured synthetic data.

versarial loss helps the model select a single mode from the distribution and generate more realistic results without blurring.

A Generative Adversarial Network (GAN) [27] is capable of producing semantically sound samples by creating a competition between a generator, which endeavors to capture the data distribution, and a discriminator, which judges the output of the generator and penalizes unrealistic images. Both networks are trained simultaneously to achieve an equilibrium. While most generative models generate images from a latent noise vector as the input to the generator, our model is conditioned on an input image (RGB).

More formally, our generative model learns a mapping from the input image  $x$  (RGB view) to the output image  $y$  (scene depth)  $G : x \rightarrow y$ . The generator ( $G$ ) attempts to produce fake samples  $G(x) = \hat{y}$  that cannot be distinguished from real ground truth samples  $y$  by the discriminator ( $D$ ) that is adversarially trained to detect the fake samples produced by the generator.

Many other approaches following a similar framework incorporate a random noise vector  $z$  or drop-outs into the generator training to prevent deterministic mapping and induce stochasticity [32, 57, 54, 81]. While we experimented with both random noise as part of the generator input and drop-outs in different layers of the generator, no significant difference in the output distribution could be achieved.

### 3.1.1 Loss Function

Our objective is achieved using a loss function consisting of two components: a reconstruction loss, which incentivizes the generator to produce images that are structurally and contextually as close as possible to the ground truth. We utilize the  $\ell_1$  loss:

$$\mathcal{L}_{rec} = \|G(x) - y\|_1 \quad (1)$$

However, with the sole use of a reconstruction loss, the generator optimizes towards averaging all possible values (blurring) rather than selecting one (sharpness). To remedy this,

an adversarial loss is introduced:

$$\mathcal{L}_{adv} = \min_G \max_D \mathbb{E}_{x,y \sim \mathbb{P}_d(x,y)} [\log D(x, y)] + \mathbb{E}_{x \sim \mathbb{P}_d(x)} [\log(1 - D(x, G(x)))] \quad (2)$$

Minimum maximum distribution cross entropy?

where  $\mathbb{P}_d$  is our data distribution defined by  $\tilde{y} = G(x)$ , with  $x$  being the generator input and  $y$  the ground truth. Subsequently, the joint loss function is as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{rec} + (1 - \lambda) \mathcal{L}_{adv} \quad (3)$$

with  $\lambda$  selected empirically. This forces optimization towards explicit value selection and content preservation.

### 3.1.2 Implementation Details

Since synthetic data is needed to train the model, color and disparity images are captured from a camera view set in front of a virtual car as it automatically drives around the virtual environment and images are captured every 60 frames with randomly varying height, field of view, weather and lighting conditions at different times of day to avoid over-fitting. 80,000 images were captured with 70,000 used for training and 10,000 set aside for testing. Our model trained using this synthetic data outputs a disparity image which is converted to depth using focal length and scaled to the depth range of the KITTI image frame [55].

An important aspect of the monocular depth estimation problem is that overall structures within the RGB image (input) and the depth map (output) are aligned as they provide types of information for the exact same scene. As a result, much information (e.g. structure, geometry, object boundaries and alike) is shared between the input and output.

In this sense, we utilize skip connections in the generator rather than using a classic encoder-decoder pipeline with no skip connections [30, 35, 5, 57, 81]. By taking advantage of these skip connections, the generator has the opportunity to directly pass geometric information between corresponding

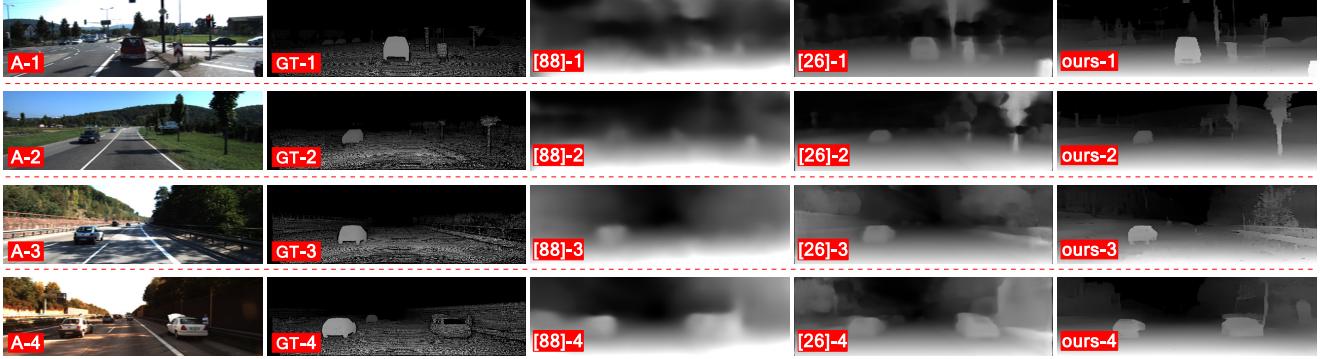


Figure 3: Qualitative comparison of our results against the state-of-the-art methods in [88, 26] over the KITTI split. *GT* denotes ground truth. Our approach produces sharp and crisp results with no blurring or additional artefacts.

layers in the encoder and the decoder without having to go through every single layer in between.

Following the success of U-net [62] which contains an efficient light-weight architecture, our generator consists of a similar pipeline, with the exception that skip connections exist between every pair of corresponding layers in the encoder and decoder. For our discriminator, we deploy the basic architecture used in [60]. Both generator and discriminator use the convolution-BatchNorm-ReLu module [31] with the discriminator using leaky ReLUs (*slope* = 0.2).

All implementation and training is done in PyTorch [56], with the ADAM [37] providing experimentally superior optimization (momentum  $\beta_1$  = 0.5,  $\beta_2$  = 0.999, initial learning rate  $\alpha$  = 0.0002). The coefficient in the joint loss function was empirically chosen to be  $\lambda$  = 0.99.

### 3.2. Stage 2 - Domain Adaptation via Style Transfer

Assuming the monocular depth estimation procedure presented in the Section 3.1 performs well (Figure 4), since the model is trained on synthetic images, the idea of estimating depth from RGB images captured in the real-world is still far fetched as the synthetic and real-world images are from widely different domains.

Our goal is thus to learn a mapping function  $\mathcal{D} : X \rightarrow Y$  from the source domain  $X$  (real-world images) to the target domain  $Y$  (synthetic images) in a way that the distributions  $\mathcal{D}(X)$  and  $Y$  are identical. When images from  $X$  are mapped into  $Y$ , their depth can be inferred using our monocular depth estimator (Section 3.1) that is specifically trained on images from  $Y$ .

While the notion of transforming images from one domain to the other is not new [90, 51, 63, 7, 70], we utilize image style transfer using generative adversarial networks, as proposed in [90], to reduce the discrepancy between our source domain (real-world data) and our target domain (synthetic data on which our depth estimator in Section 3.2 functions). This approach uses adversarial training [27] and cycle-consistency [26, 79, 89, 86] to translate between two sets of unaligned images from different domains.

Formally put, the objective is to map images between the two domains  $X, Y$  with distributions  $x \sim \mathbb{P}_d(x)$  and  $y \sim \mathbb{P}_d(y)$ . The mapping functions are approximated using two separate generators,  $G_{X \rightarrow Y}$  and  $G_{Y \rightarrow X}$  and two discriminators  $D_X$  (discriminating between  $x \in X$  and  $G_{Y \rightarrow X}(y)$ ) and  $D_Y$  (discriminating between  $y \in Y$  and  $G_{X \rightarrow Y}(x)$ ). The loss contains two components: an adversarial loss [27] and a cycle consistency loss [90]. The general pipeline of the approach (along with the depth estimation model 3.1) is seen in Figure 2, with three generators  $G_{A \rightarrow B}$ ,  $G_{B \rightarrow A}$  and  $G_{B \rightarrow C}$ , and three discriminators  $D_A$ ,  $D_B$  and  $D_C$ .

#### 3.2.1 Loss Function

Since there are two generators to constrain the content of the images, there are two mapping functions, each with its own loss but with similar formulations. The use of an adversarial loss guarantees the style of one domain is transferred to the other. The loss for  $G_{X \rightarrow Y}$  with  $D_Y$  is as follows:

$$\begin{aligned} \mathcal{L}_{adv}(X \rightarrow Y) = & \min_{G_{X \rightarrow Y}} \max_{D_Y} \mathbb{E}_{y \sim \mathbb{P}_d(y)} [\log D_Y(y)] + \\ & \mathbb{E}_{x \sim \mathbb{P}_d(x)} [\log(1 - D_Y(G_{X \rightarrow Y}(x)))] \end{aligned} \quad (4)$$

where  $\mathbb{P}_d$  is the data distribution,  $X$  the source domain with samples  $x$  and  $Y$  the target domain with samples  $y$ . Similarly, for  $G_{Y \rightarrow X}$  and  $D_X$ , the adversarial loss is:

$$\begin{aligned} \mathcal{L}_{adv}(Y \rightarrow X) = & \min_{G_{Y \rightarrow X}} \max_{D_X} \mathbb{E}_{x \sim \mathbb{P}_d(x)} [\log D_X(x)] + \\ & \mathbb{E}_{y \sim \mathbb{P}_d(y)} [\log(1 - D_X(G_{Y \rightarrow X}(y)))] \end{aligned} \quad (5)$$

The original work in [90] replaces the log likelihood by a least square loss to improve training stability [53]. We experimented with that setup, but noticed no significant improvement in training stability or the quality of the results. Therefore the original adversarial loss is used.

In order to constrain the adversarial loss of the generators to encourage the model to produce desirable contextually



Figure 4: Comparison of the results with different components of the loss in the depth estimation model (Section 3.1).

Method	Training Data	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Ours w/o domain adaptation	K+S*	0.498	6.533	9.382	0.609	0.712	0.823	0.883
Ours w/ the approach of Johnson <i>et al.</i>	K+S*	0.154	1.338	6.470	0.296	0.874	0.962	0.981
Ours w/ the cycleGAN approach	K+S*	<b>0.101</b>	<b>1.048</b>	<b>5.308</b>	<b>0.184</b>	<b>0.903</b>	<b>0.988</b>	<b>0.992</b>

Table 2: Ablation study over the KITTI dataset using the KITTI split. our approach is trained using, KITTI (K) and synthetic data (S\*). The approach with domain adaptation using cycleGAN [90] provides the best results.

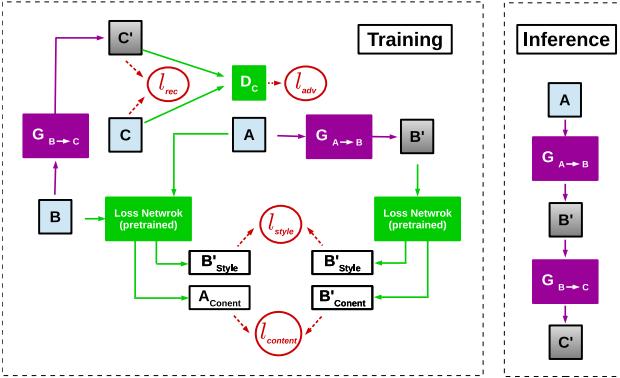


Figure 5: Our approach using [35]. Images from domain  $A$  (real-world) are transformed into  $B$  (synthetic) and then to  $C$  (pixel-perfect depth maps).  $A, B, C$  represent ground truth images and  $A', B', C'$  denote generated images.

coherent images rather than random images with the target domain, a cycle-consistency loss is added that prompts the model to become capable of bringing an image  $x$  that is translated into the target domain  $Y$  using  $G_{X \rightarrow Y}$  back into the source domain  $X$  using  $G_{Y \rightarrow X}$ . Essentially, after a full cycle, we should have:  $G_{Y \rightarrow X}(G_{X \rightarrow Y}(x)) = x$  and vice versa. As a result, the cycle-consistency loss is:

$$\begin{aligned} \mathcal{L}_{cyc} = & ||G_{Y \rightarrow X}(G_{X \rightarrow Y}(x)) - x||_1 \\ & + ||G_{X \rightarrow Y}(G_{Y \rightarrow X}(y)) - y||_1 \end{aligned} \quad (6)$$

Subsequently, the joint loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{adv}(X \rightarrow Y) + \mathcal{L}_{adv}(Y \rightarrow X) + \lambda \mathcal{L}_{cyc} \quad (7)$$

with  $\lambda$  selected empirically.

### 3.2.2 Implementation Details

The style generator architectures are based on the work in [35] with two convolutional layers followed by nine residual blocks [29] and two up-convolutions that bring the image back to its original input size.

As for the discriminators, the same architecture is used as was in Section 2.1. Additionally, the discriminators are updated based on the last 50 generator outputs and not just the last generated image [90, 70].

All implementation and training was done in PyTorch [56], and ADAM [37] was used to perform the optimization for the task (momentum  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$ , and initial learning rate  $\alpha = 0.0001$ ). The coefficient in the joint loss function in Eqn. 7 was empirically chosen to be  $\lambda = 10$ .

## 4. Experimental Results

In this section, we evaluate our approach using ablation studies and both qualitative and quantitative comparisons with state-of-the-art monocular depth estimation methods via publicly available datasets. We use KITTI [55] for our comparisons and Make3D [66] in addition to data captured locally to test how our approach generalizes over unseen data domains. It is worth noting that using a GeForce GTX 1080 Ti, the entire two passes take an average of 22.7 milliseconds, making the approach real-time ( $\sim 44$  fps).

### 4.1. KITTI

To facilitate better numerical comparisons against existing approaches within the literature, we test our model using the 697 images from the data split suggested in [18]. As seen in Table 1, our approach performs better than the



Figure 6: Results demonstrating the importance of style transfer. Left column shows results with no domain adaptation. Middle column contains results with [35] as domain transfer and the right column indicates results with [90].

current state-of-the-art [18, 48, 88, 26] with lower error and higher accuracy. Measurement metrics are based on [18]. Some of the comparators [88, 26] use a combination of different datasets for training and fine-tuning to boost performance, while we only use the synthetic data and KITTI [55]. Additionally, following the conventions of the literature [18, 48, 88, 26], the error measurements are all performed in depth space, while our approach produces disparities, and as a result, small precision issues are expected.

We also used the data split of 200 images in KITTI [55] to provide better *qualitative* evaluation, since the ground truth disparity images within this split are of considerably higher quality than velodyne laser data and provide CAD models as replacements for moving cars. As is clearly shown in Figure 3, compared to the state-of-the-art approaches [88, 26] trained on similar data domains, our approach generates sharper and more crisp outputs in which object boundaries and thin structures are well preserved.

## 4.2. Ablation Studies

A crucial part of this work was interpreting the necessity of the components of our approach. Our monocular depth estimation model (Section 3.1) utilizes a combination of reconstruction and adversarial losses (Eqn. 3). We trained our model using the reconstruction loss only and the adversarial loss only to test their importance. Figure 4 demonstrates the effects of removing parts of the training objective. The model based only on the reconstruction loss ( $\ell_1$ ) produces contextually sound but blurry results, while the adversarial loss generates sharp outputs that contain artefacts. The full approach creates more accurate results without unwanted effects. Further evidence of the efficacy of a combination of a reconstruction and adversarial loss is found in [32].

Another important aspect of our ablation study entails evaluating the importance of domain transfer (Section 3.2)



Figure 7: Qualitative results of our approach on urban driving scenes captured locally without further training.

within our framework. Due to the differences in the domains of the synthetic and natural data, our depth estimator directly applied to real-world data does not produce qualitatively or quantitatively desirable results, which makes the domain adaptation step necessary (Table 2 and Figure 6).

While our approach requires an adversarial discriminator [90] to carry out the style transfer needed for our domain adaptation, [47] has suggested that more conventional style transfer, which involves matching the Gram matrices [68] of feature maps, is theoretically equivalent to minimizing the Maximum Mean Discrepancy with the second order polynomial kernel, and leads to domain adaptation.

As evidence for the notion that a discriminator can perform the task even better, we also experiment with the style transfer approach of [35], which improves on the original work [23] by training a generator that can transfer a specific style (that of our synthetic domain in our work) onto a set of images of a specific domain (real-world images). A loss network (pre-trained VGG [71]) is used to extract the image style and content (as in [23]). This network calculates the loss values for content (based on the  $\ell_2$  difference between feature representations extracted from the loss network) and style (from the squared Frobenius norm of the distance between the *Gram matrices* of the input and main style images) that are used to train the generator. An overview of the entire pipeline using [35] (along with the depth estimation model - Section 3.1) can be seen in Figure 5.

Whilst [90] transfers the style between two *sets* of unaligned images from different domains, [35] requires *one* specific image to be used as the style image. We have access to tens of thousands of images representing the same style. This is remedied by collecting a number of synthetic images that contain a variety of objects, textures and colors that represent their domain, and pooling their features to create a single image that holds our desired style.

The data split of 200 images in KITTI [55] was used to evaluate our approach regarding the effects of domain adaptation via style transfer. We experimented with both [90] and [35], in addition to feeding real-world images to our depth estimator without domain adaptation. As seen in the results presented in Table 2, not using style transfer ends in

Method	Error Metrics (lower, better)			
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log
Train Set Mean	0.814	12.992	11.411	0.285
Karsch <i>et al.</i> [36]	0.398	4.723	7.801	0.138
Liu <i>et al.</i> [50]	0.441	6.102	9.346	0.153
Godard <i>et al.</i> [26]	0.505	10.172	10.936	0.179
Zhou <i>et al.</i> [88]	0.356	4.948	9.737	0.443
Laina <i>et al.</i> [40]	0.189	1.711	5.285	0.079
Our Approach	0.423	9.343	9.002	0.122

Table 3: Comparative results on Make3D [66], on which the approach is not trained. [36, 50, 40] are specifically trained on Make3D. Following [88, 26], errors are only calculated where depth is less than 70 meters in a central image crop.

considerable amount of anomalies in the output while translating images into synthetic space using [90] before depth estimation generates significantly better outputs. Figure 6 provides qualitative results leading to the same conclusion.

#### 4.3. Model Generalization

Images used in our training procedure come from the synthetic environment [64] and the KITTI dataset [55], but we evaluate our approach on additional data to test the model generalization capabilities. Using data captured locally in an urban environment we generated visually convincing depth without any training on our data which are sharp, coherent and plausible as seen in Figure 7.

Furthermore, we tested our model on the Make3D dataset [66], which contains paired RGB and depth images from a different domain, and compared our results against supervised methods trained on said dataset and state-of-the-art monocular depth estimation methods. Even though our approach does not numerically beat comparators that are trained on Make3D [36, 50, 40], as seen in Table 3, our results are promising despite no training over this data, and outputs are highly plausible qualitatively, even compared to the ground truth. Some results are seen in Figure 8.

#### 4.4. Limitations

Even though the proposed approach is capable of generating high quality depth by taking advantage of domain adaptation through image style transfer, the very idea of style transfer brings forth certain shortcomings. The biggest issue is that the approach is incapable of adapting to sudden lighting changes and saturation during style transfer. When the two domains significantly vary in intensity differences between lit areas and shadows (as is the case with our approach), shadows can be recognized as elevated surfaces or foreground objects post style transfer. Figure 9 contains some examples of how these issues arise.

Moreover, despite the fact that depth holes are generally considered undesirable [3, 4, 2, 49, 58], certain areas within the scene depth should remain without depth val-

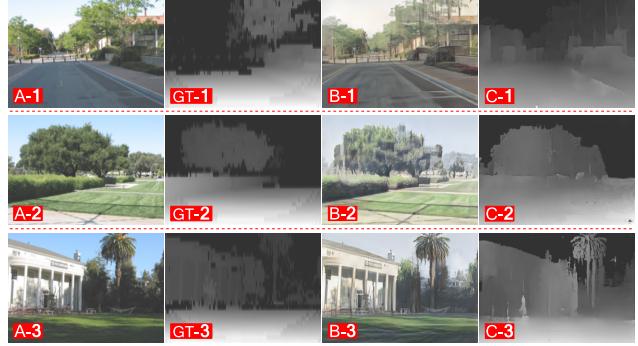


Figure 8: Results on the Make3D test set [66]. Note the quality of our outputs despite the vast differences between this dataset and the images used in our training.



Figure 9: Examples of failures, mainly due to light saturation and shadows. Issues are marked with red boxes.

ues (*e.g.* very distant objects and sky). However, a supervised monocular depth estimation approach such as ours (even with style transfer) is incapable of distinguishing the sky from other extremely saturated objects within the scene, which can lead to small holes within the scene. This issue can be tackled in any future work by adding a weighted loss component that can penalize the generator when holes are misplaced based on the approximate location of the sky and other distant background objects.

## 5. Conclusion

We have proposed a learning-based monocular depth estimation approach. Using synthetic data captured from a graphically rendered urban environment designed for gaming applications, an effective depth estimation model can be trained in a supervised manner. However, this model cannot perform well on real-world data as the domain distributions to which these two sets of data belong are widely different. Relying on new theoretical studies connecting style transfer and distances between distributions, we propose the use of a GAN-based style transfer approach to adapt our real-world data to fit into the distribution approximated by the generator in our depth estimation model. Although some isolated issues remain, experimental results prove the superiority of our approach compared to contemporary state-of-the-art methods tackling the same problem.

Supplementary video: <https://vimeo.com/260393753> (larger, higher quality results).

## References

- [1] A. Abrams, C. Hawley, and R. Pless. Heliometric stereo: Shape from sun position. *Proc. Euro. Conf. Computer Vision*, pages 357–370, 2012. 1
- [2] A. Atapour-Abarghouei and T. Breckon. Depthcomp: Real-time depth image completion based on prior semantic scene segmentation. In *Proc. British Machine Vision Conference*, 2017. 1, 8
- [3] A. Atapour-Abarghouei and T. Breckon. A comparative review of plausible hole filling strategies in the context of scene depth image completion. *J. Computers and Graphics*, 72:39–58, 2018. 8
- [4] A. Atapour-Abarghouei, G. Payen de La Garanderie, and T. P. Breckon. Back to butterworth - a fourier basis for 3d surface relief hole filling within rgb-d imagery. In *Proc. Int. Conf. Pattern Recognition*, pages 2813–2818. IEEE, 2016. 1, 8
- [5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 4
- [6] M. H. Baig, V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan. Im2depth: Scalable exemplar based depth transfer. In *Proc. Winter Conf. Applications of Computer Vision*, pages 145–152. IEEE, 2014. 2
- [7] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016. 5
- [8] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Trans. Circuits and Systems for Video Technology*, 2017. 2
- [9] P. Cavestany, A. Rodriguez, H. Martinez-Barbera, and T. Breckon. Improved 3d sparse maps for high-performance structure from motion with low-cost omnidirectional robots. In *Proc. Int. Conf. Image Processing*, pages 4927–4931, 2015. 1
- [10] A. J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016. 3
- [11] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016. 3
- [12] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 1
- [13] Y.-L. Chen and C.-T. Hsu. Towards deep style transfer: A content-aware perspective. In *Proc. British Machine Vision Conference*, 2016. 3
- [14] L. Ding and G. Sharma. Fusing structure from motion and lidar for dense accurate depth map estimation. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, pages 1283–1287. IEEE, 2017. 1
- [15] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2, 3
- [16] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016. 3
- [17] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. Int. Conf. Computer Vision*, pages 2650–2658, 2015. 1, 2
- [18] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014. 1, 2, 6, 7
- [19] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 4340–4349, 2016. 1
- [20] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. Int. Conf. Machine Learning*, pages 1180–1189, 2015. 2, 3
- [21] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Machine Learning Research*, 17(59):1–35, 2016. 2
- [22] R. Garg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proc. Euro. Conf. Computer Vision*, pages 740–756. Springer, 2016. 1, 2
- [23] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 3, 7
- [24] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 3
- [25] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proc. Euro. Conf. Computer Vision*, pages 597–613. Springer, 2016. 2, 3
- [26] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 6602 – 6611, 2017. 1, 2, 5, 7, 8
- [27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3, 4, 5
- [28] O. Hamilton and T. Breckon. Generalized dynamic object removal for dense stereo vision based scene mapping using synthesised optical flow. In *Proc. Int. Conf. Image Processing*, pages 3439–3443, 2016. 1
- [29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6
- [30] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 4

- [31] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Machine Learning*, pages 448–456, 2015. 5
- [32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 5967–5976, 2017. 3, 4, 7
- [33] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 2
- [34] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song. Neural style transfer: A review. *arXiv preprint arXiv:1705.04058*, 2017. 3
- [35] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. Euro. Conf. Computer Vision*, pages 694–711, 2016. 3, 4, 6, 7
- [36] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *Proc. Euro. Conf. Computer Vision*, pages 775–788. Springer, 2012. 8
- [37] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations*, 2014. 5, 6
- [38] Y. Kuznetsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 6647–6655, 2017. 2
- [39] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 89–96, 2014. 1, 2
- [40] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. Int. Conf. 3D Vision*, pages 239–248. IEEE, 2016. 2, 8
- [41] T. A. Le, A. G. Baydin, R. Zinkov, and F. Wood. Using synthetic data to train neural networks is model-based reasoning. *arXiv preprint arXiv:1703.00868*, 2017. 1
- [42] B. Li, Y. Dai, H. Chen, and M. He. Single image depth estimation by dilated deep residual convolutional neural network and soft-weight-sum inference. *arXiv preprint arXiv:1705.00534*, 2017. 1
- [43] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1119–1127, 2015. 2
- [44] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 2479–2486, 2016. 3
- [45] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. Euro. Conf. Computer Vision*, pages 702–716. Springer, 2016. 3
- [46] C. Li, X. Zhao, Z. Zhang, and S. Du. Generative adversarial dehaze mapping nets. *Pattern Recognition Letters*, 2017. 3
- [47] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 2, 3, 7
- [48] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016. 1, 2, 7
- [49] J. Liu, X. Gong, and J. Liu. Guided inpainting and filtering for kinect depth maps. In *Proc. Int. Conf. Pattern Recognition*, pages 2055–2058. IEEE, 2012. 1, 8
- [50] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 716–723, 2014. 8
- [51] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 469–477, 2016. 3, 5
- [52] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *Proc. Int. Conf. Machine Learning*, pages 97–105, 2015. 2
- [53] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 2016. 5
- [54] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *Proc. Int. Conf. Learning Representations*, 2016. 4
- [55] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 1, 4, 6, 7, 8
- [56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 5, 6
- [57] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 3, 4
- [58] F. Qi, J. Han, P. Wang, G. Shi, and F. Li. Structure guided fusion for depth map inpainting. *Pattern Recognition Letters*, 34(1):70–76, 2013. 1, 8
- [59] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning*, pages 131–160, 2009. 2
- [60] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 5
- [61] P. S. Rajpura, R. S. Hegde, and H. Bojinov. Object detection using deep cnns trained on synthetic images. *arXiv preprint arXiv:1706.06782*, 2017. 1
- [62] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 5
- [63] R. Rosales, K. Acham, and B. J. Frey. Unsupervised image translation. In *Proc. Int. Conf. Computer Vision*, pages 472–478, 2003. 5
- [64] A. Ruano Miralles. An open-source development environment for self-driving vehicles. 2017. 1, 3, 8
- [65] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pages 1161–1168, 2006. 2

- [66] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 2, 6, 8
- [67] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47(1-3):7–42, 2002. 1
- [68] H. Schwerdtfeger. *Introduction to Linear Algebra and the Theory of Matrices*. P. Noordhoff Groningen, 1950. 3, 7
- [69] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635, 2017. 1
- [70] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 2242–2251, 2017. 5, 6
- [71] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Proc. Int. Conf. Learning Representations*, 2015. 7
- [72] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Proc. Int. Conf. Computer Vision – Workshops*, pages 443–450. Springer, 2016. 2
- [73] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1940–1948, 2015. 1
- [74] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proc. Int. Conf. Computer Vision*, pages 4068–4076, 2015. 2, 3
- [75] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017. 2, 3
- [76] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2
- [77] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proc. Int. Conf. Machine Learning*, pages 1349–1357, 2016. 3
- [78] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *Proc. Int. Conf. Computer Vision*, pages 3352–3361. IEEE, 2017. 3
- [79] F. Wang, Q. Huang, and L. J. Guibas. Image co-segmentation via consistent functional maps. In *Proc. Int. Conf. Computer Vision*, pages 849–856, 2013. 5
- [80] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 539–547, 2015. 2
- [81] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *Proc. Euro. Conf. Computer Vision*, pages 318–335. Springer, 2016. 4
- [82] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):191139, 1980. 1
- [83] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *Proc. Euro. Conf. Computer Vision*, pages 842–857. Springer, 2016. 1, 2
- [84] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 4076 – 4084, 2017. 3
- [85] R. A. Yeh\*, C. Chen\*, T. Y. Lim, S. A. G., M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 6882–6890, 2017. \* equal contribution. 3
- [86] Z. Yi, H. Zhang, P. T. Gong, et al. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proc. Int. Conf. Computer Vision*, pages 2868–2876, 2017. 5
- [87] R. Yin. Content aware neural style transfer. *arXiv preprint arXiv:1601.04568*, 2016. 3
- [88] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 6612–6619, 2017. 2, 5, 7, 8
- [89] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3D-guided cycle consistency. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 117–126, 2016. 5
- [90] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proc. Int. Conf. Computer Vision*, pages 2242 – 2251, 2017. 3, 5, 6, 7, 8
- [91] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 614–622, 2015. 1