Nr. 82
1. April 2021

universität innsbruck
Institut für Mathematik

Preprint-Series: Department of Mathematics - Applied Mathematics

Unpaired Single_Image Dept Synthesis with cycle-consistent Wasserstein GANs

C. Angermann, A. Moravova, M.Haltmeier,
S. Jonsson, C. Laubichler

AppliedMathematics

Technikerstraße 13 - 6020 Innsbruck - Austria
Tel.: +43 512 507 53803 Fax: +43 512 507 53898
https://applied-math.uibk.ac.at

Unpaired Single-Image Depth Synthesis with cycle-consistent Wasserstein GANs

Christoph Angermann Adéla Moravová Markus Haltmeier
Department of Mathematics, University of Innsbruck
Technikerstraße 13, 6020 Innsbruck, Austria
{christoph.angermann,adela.moravova,markus.haltmeier}@uibk.ac.at

Steinbjörn Jónsson
INNIO Jenbacher GmbH & Co OG
Achenseestrasse 1-3, 6200 Jenbach, Austria
steinbjorn.jonsson@innio.com

Christian Laubichler
LEC GmbH
Inffeldgasse 19, 8010 Graz, Austria
christian.laubichler@lec.tugraz.at

April 1, 2021

Abstract

Real-time estimation of actual environment depth is an essential module for various autonomous system tasks such as localization, obstacle detection and pose estimation. During the last decade of machine learning, extensive deployment of deep learning methods to computer vision tasks yielded successful approaches for realistic depth synthesis out of a simple RGB modality. While most of these models rest on paired depth data or availability of video sequences and stereo images, there is a lack of methods facing single-image depth synthesis in an unsupervised manner. Therefore, in this study, latest advancements in the field of generative neural networks are leveraged to fully unsupervised single-image depth synthesis. To be more exact, two cycle-consistent generators for RGB-to-depth and depth-to-RGB transfer are implemented and simultaneously optimized using the Wasserstein-1 distance. To ensure plausibility of the proposed method, we apply the models to a self acquised industrial data set as well as to the renown NYU Depth v2 data set, which allows comparison with existing approaches. The observed success in this study suggests high potential for unpaired single-image depth estimation in real world applications.

1 Introduction

Real-time depth inference of a given scene is a basic and highly relevant computer vision task, which can be applied in various robotic tasks such as simultaneous localization and mapping as well as obstacle avoidance in autonomous driving systems for instance [1]. Direct depth information of a given scene or environment can be directly obtained by depth sensors, where RGB-D cameras and LIDAR are often considered in practice and were also utilized for data generation for the well-known KITTI [2] (LIDAR) and NYU Depth v2 [3] (RGB-D camera) data sets. Since these sensors are typically bulky, heavy devices which also suffer from external influences in nature, their applicability to fast full-image depth synthesis and small on-site devices is limited. Just discussed limitations motivated depth estimation out of a much simpler modality in terms of acquisition effort, namely an RGB image. This development initiated a completely new field of research in computer vision, where the breakthrough was achieved by proposing deep convolutional neural networks (DCNNs) for monocular depth synthesis [4].

Basically, single-image depth estimation out of RGB images can be seen as a modality transfer, where observed data of a simpler modality is mapped to desired properties of a more complex modality. Although DCNNs are a promising approach for succeeding on such transfer tasks [4, 5], they are commonly based on large amounts of training data where generation and acquisition can be a demanding task. In the supervised setting, DCNNs make use of paired training data during network parameter optimization, i.e. the network is provided with monocular RGB images and corresponding per-pixel depth [4, 6, 7]. Since in many applications large scale dense depth profiles are not abundant, supervised approaches are not feasible for such environments. Therefore, monocular depth synthesis research emphasized in the past years on development of self-supervised methods, where the geometric constraints between monocular frames are regarded as the supervisory signal. The idea is to train the model on monocular RGB sequences in a self-supervised manner, where simultaneously a depth and a pose estimation network are optimized via a view-synthesis loss [1, 8, 9]. During inference, only the depth model has to be evaluated to synthesize depth profiles where no ground truth data is available during training.

In this study we consider depth profiles that are unaligned with respect to given RGB records. Basically, just discussed self-supervised method could be utilized - provided that monocular sequences of a moving scene are available. This is not the case in many industrial applications. Another, very recent, example is surface depth estimation of cylinder liners which are mounted in large internal combustion engines. A handheld microscope generates single RGB records of the cylinder's inner surface with a spatial range of $4.2 \times 4.2\,\mathrm{mm}^2$ and a depth resolution on micrometer scale (cf. Figure 1). Cylinder condition can be assessed via a depth profile of the measured area. Current standards have the disadvantage that depth sensing of microscopic cylinder surface areas is a time-consuming and resource-intensive task which consists of disassembling the liner, removing it from the engine, cutting it into segments and measuring them with a highly expensive and unportable confocal microscope [10]. Furthermore, since the records are on a quite small scale, it is quite challenging to measure depth profile aligned with the previous RGB record. Just discussed logistics chain confirms the necessity of a depth synthesizing method using a simpler sensing method such as RGB image of a motionless scene with unaligned depth data.

Several recent advances in unpaired modality and style transfer are based on generative adversarial models (GAN) [13], cycle-consistency [14] and probabilistic distance measures [15, 16]. The method proposed in this work leverages established model architectures and training strategies in this field to unpaired single image depth synthesis, and is tested on afore mentioned industrial application of surface depth estimation. Furthermore, the approach is applied to the renown NYU Depth v2 data set [3], where RGB and depth frames are not aligned during preprocessing in order to imitate the scenario of absence of paired modality pairs.

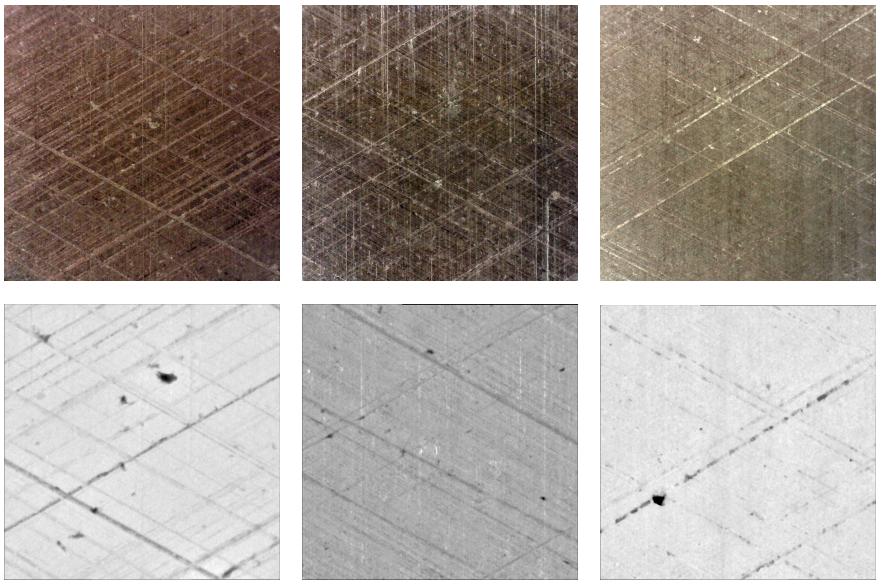

Figure 1: Top: RGB measurements of inner surfaces of 3 cylinder liners with a spatial range of $4.2 \times 4.2\,\mathrm{mm}^2$, recorded by a handheld microscope [11]. Bottom: Depth profile of the same cylinder with a spatial range of $1.9 \times 1.9\,\mathrm{mm}^2$, measured with a confocal microscope [12]. The modalities are not aligned pixel-wise.

Contributions:

- To the best of our knowledge, this is the first research work facing single-image depth prediction, where video sequences, stereo images and aligned depth data are not available or practically not feasible.
- In our implementation, a cycle-consistent GAN is optimized via the Wasserstein-1 distance. Therefore, this paper provides comprehensive results and experiments on an architecture, where latest progresses in unsupervised and adversarial learning are combined and where hardly any empirical observations exist yet.
- In this study, a solution to the industrial modality transfer problem of surface depth estimation is found and further applied to the completely different task of unsupervised indoor-scene depth synthesis, which indicates the universality of the approach.

2 Related Work

In this section, deep learning models, which take monocular input and predict a per-pixel depth profile, are reviewed as well as latest progress in designing and training GANs for unsupervised modality transfer.

2.1 Monocular Depth Estimation

Deep learning based methods achieve state-of-the-art results on depth synthesis task by training a DCNN on a large-scale and extensive data set [2, 3]. Most of RGB-based models are fully supervised, i.e. they require corresponding depth data that is pixel-wise aligned and therefore very hard to acquire. One of the first DCNN approaches by Eigen et al. [4] included sequential deployment of a coarse-scale stack and a refinement module and was benchmarked on the KITTI and the NYU Depth v2 data set. Using a encoder-decoder structure in combination with an adversarial loss term helped to increase visual quality of the dense depth estimates [17]. Later methods also considered deep residual networks [7] or deep ordinal regression networks [18] in order to significantly increase performance on these data sets, where commonly considered performance measures are the root mean squared error (RMSE) or the δ_1 accuracy [1]. Since a lot of research focused on further performance increase at the expense of model complexity and runtime, Wofk et al. [6] used a lightweight network architecture [19] and achieved comparable results.

In the absence of aligned ground truth data, depth synthesis models are trained based in a self-supervised manner, where the supervisory signal is generated via image reconstruction, either based on stereo images [20, 21] or, as discussed in the following, on monocular videos. Zhou et al. [9] proposed a depth network to predict depth of a single frame, which is simultaneously trained with a pose network that regresses transformations between subsequent frames. Geometric constraints of neighboring frames enable a view synthesis which can be used as a reconstruction loss. Later introduction of an additional motion explanation mask to the reconstruction loss in order to reduce the influence of dynamic objects increased performance on the KITTI benchmark data set [22]. While research on self-supervised monocular training explored very complex models and training methods, Godard et al. [8] achieved superiour results on the KITTI data set using a surprisingly simple model in combination with optimized design choices. Nevertheless, just discussed architectures rely on either non static scenes or a moving position of the camera, which makes them inapplicable to single-image tasks with unpaired ground truth.

2.2 Modality transfer with GANs

A GAN [13] consists of a generator network G mapping from latent space to image space $\mathcal{X}$, where parameters of the generator are adapted such that the distribution of generated examples assimilates the distribution of a given data set. To be able to assess similarity between arbitrary high-dimensional image distributions, a discriminator f is trained simultaneously to distinguish between generator distribution and real data distribution. In a two-player min-max game, generator parameters are then updated to fool a steadily improving discriminator.

While the quintessence of GANs is to draw synthetic instances following a given data distribution, cycle-consistent GANs [14] allow for one-to-one mappings between two image domains $\mathcal{X}$ and $\mathcal{Y}$. Basically, two generator networks $G_{\mathcal{Y}}, G_{\mathcal{X}}$ and corresponding discriminator networks $f_{\mathcal{Y}}, f_{\mathcal{X}}$ are trained simultaneously to enable generation of synthetic instances for both image domains (e.g. synthesizing winter landscape from summer scenes and vice versa). To ensure a one-to-one correspondence, a cycle-loss term is added to the two adversarial loss functionals. Although cycle-consistent GANs initially constructed for style transfer purposes, they were also very well received in the area of modality transfer, especially in biomedical applications [5, 23, 24]. Since optimization and parameter fine-tuning of GANs often turns out to be extremely demanding and time-intensive, a lot of research emphasized on stabilizing the training process by development of network architecture constraints [25] and improved discriminator functions [15, 16].

While style transfer emphasizes on modification of a given modality, modality transfer allows to change the domains, even when they are defined in distinct dimensional spaces. This recommends the cycle-consistency architecture for depth synthesis, as it has been the case in [26,27] for stereo images, respectively. Nevertheless, comprehensive literature study confirmed that research on cycle-consistent GANs for unpaired single-image depth estimation has not progressed far. Although Kundu et al. [28] investigated GANs for this scenario, they utilized it in a context of domain adaptation using an extra synthetic data set, which does not correspond to a fully unsupervised method. Therefore, this research work bridges this gap and introduces latest progress on GAN architectures and corresponding training techniques to the research field of depth synthesis, where proposed methods are evaluated on the industrial application of surface depth estimation (cf. Figure 1) as well as on the benchmark NYU Depth v2 data set [3], which enables qualitative and quantitative comparison to other methods.

3 Method

In this section, an approach for single-image depth synthesis with unaligned data is proposed and the introduced training strategy is discussed in detail.

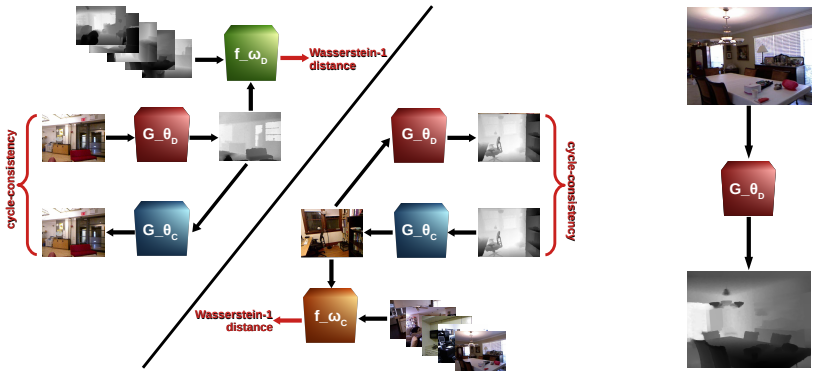

Figure 2: Illustration of a cycle-consistent Wasserstein GAN: The left plot describes, on which domain each of the four networks $G_{\theta_\mathrm{D}}, G_{\theta_\mathrm{C}}, f_{\omega_\mathrm{D}}, f_{\omega_\mathrm{C}}$ operates and where the Wasserstein-1 distance and the cycle-consistency are calculated to simultaneously update these four network functions (cf. Algorithm 1). The right plot indicates that during inference only G_{θ_D} has to be deployed to synthesize new depth profiles that are not irrelevant to the RGB input.

3.1 Cycle-Consistent GANs

The underlying structure of the proposed modality synthesis is a GAN with a cycle-consistency term [14] (cf. Figure 2). To be more exact, let $\mathcal{X} \subset [0, 255]^{d_1 \times d_2 \times 3}$ and $\mathcal{Y} \subset \mathbb{R}^{d_1 \times d_2 \times 1}$ denote the domain of RGB and depth images, respectively, where $d_1 \cdot d_2$ is the number of image pixels. Furthermore, let $X \triangleq \{x_1, \ldots, x_M\}$ be the set of M given RGB images and $Y \triangleq \{y_1, \ldots, y_N\}$ the set of N available but unaligned depth profiles. A cycle-consistent GAN (cycleGAN) includes a generator function $G_{\theta_\mathrm{D}} : \mathcal{X} \to \mathcal{Y}$, which aims to map an input RGB image to a corresponding depth counterpart in the depth domain. A generator function for image transfer is commonly approximated by a DCNN, which is parameterized by a weight vector θ_D

consisting of several convolution kernels. Note that determining G_{θ_D} is an ill-posed problem, since a single 2D image may be generated by an infinite number of distinct depth profiles [29]. By adjusting θ_D, one can bring the distribution of generator outputs P_{θ_D} closer to the real data distribution on the depth domain P_D. We have to be aware of that we do not know how P_{θ_D} and P_D actually look like, we only have access to unpaired training samples $G_{\theta_\mathrm{D}}(x) \sim P_{\theta_\mathrm{D}},\ x \in X$ and $y \sim P_\mathrm{D},\ y \in Y$. To ensure assimilation of both high-dimensional distributions in the cycleGAN setting, an adversarial approach is deployed. The distance between generator and real distribution is estimated by an additional DCNN $f_{\omega_\mathrm{D}} : \mathcal{Y} \to \mathbb{R}$, which is parameterized by weight vector ω_D and has to be trained simultaneously with the generator network since P_{θ_D} changes after each update to the generator weights θ_D. This ensures that G_{θ_D} can be pitted against a steadily improving loss network f_{ω_D}.

In the context of depth learning, it is not sufficient to only ensure that the output samples lie in the depth domain, one must also take care that synthetic depth profiles do not become irrelevant to the input. A cycle-consistency constraint forces generator input and output to share same spatial structure by taking care of the distance between the input and the reconstruction of the synthesized depth profile. Obviously, calculation of a reconstruction error requires a opposite generator function $G_{\theta_\mathrm{C}} : \mathcal{Y} \to \mathcal{X}$ as well as the corresponding distance network $f_{\omega_\mathrm{C}} : \mathcal{X} \to \mathbb{R}$. Both have to be optimized analogously to the RGB-to-depth direction. The cycle-consistency error is commonly assessed by applying a distance function to the pairs $(x, G_{\theta_\mathrm{C}}(G_{\theta_\mathrm{D}}(x)))$ and $(y, G_{\theta_\mathrm{D}}(G_{\theta_\mathrm{C}}(y)))$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Commonly considered distance functions are the mean absolute error (MAE) as well as the mean squared error (MSE). In subsequent implementations, the influence of the cycle-consistency term during generator updates is triggered via a loss weight $\lambda_\mathrm{cyc} > 0$.

3.2 Generator Distribution Distance

For both modality transfer directions, RGB-to-depth and depth-to-RGB, a generator G_θ has to be optimized against a distance network f_ω, which simultaneously has to be updated after each change in the generator output distribution. When GANs were introduced the first time, Goodfellow et al. [13] used discriminative networks to assess distribution distance, i.e. the generator was pitted against a function assigning high probability to real samples of the target domain drawn by P_r and small probability to generated instances following P_{G_θ}. This yields the following adversarial updates for the two parameterized networks f_ω and G_θ:

$$\begin{aligned} &\min_\omega \left[-\mathbb{E}_{y \sim P_r} \left[\log f_\omega(y) \right] - \mathbb{E}_{x \sim P_{G_\theta}} \left[\log(1 - f_\omega(x)) \right] \right] \\ &\min_\theta \left[-\mathbb{E}_{x \sim P_{G_\theta}} \left[\log(f_\omega(x)) \right] \right], \end{aligned} \tag{1}$$

where iteration between k discriminator updates and 1 generator update is suggested. Obviously, it is desired that f_ω gives a good estimation on the actual distance between P_r and P_{G_θ} to ensure reliable generator updates. A closer look on minimization problem (1) confirms, that a nearly perfect discriminator (i.e. $f_\omega(y) \approx 1,\ y \sim P_r$ and $f_\omega(x) \approx 0,\ x \sim P_{G_\theta}$) leads to vanishing gradients considering the generator update. Therefore, to ensure non-zero gradients for generator updates, the discriminator must not be trained close to optimality which on the other side leads to worse distance estimations. The consequences are less reliable generator updates and non-interpretability of the discriminator loss. Non-interpretability is referred to as the value of the discriminator output does not correspond to the image quality of the generator.

The dilemma is discussed very thoroughly in [15], where this problem is bypassed by replacement of the discriminator with a network that approximates the Wasserstein-1 distance [15, 30, 31] between real and generator distribution. The Wasserstein-1 distance

(earth mover distance) between two distribution P_1 and P_2 is defined as $\mathcal{W}_1(P_1, P_2) \triangleq \inf_{J \in \mathcal{J}(P_1,P_2)} \mathbb{E}_{(x,y)\sim J} \|x - y\|$, where the infimum is taken over the set of all joint probability distributions that have marginal distributions P_1 and P_2. The Wasserstein-1 distance can be interpreted in the setting of optimal mass transport. In this setting, one aims to find an optimal transfer plan to transport one mass distribution into another as cheaply as possible in reference to a given cost function. Taking a metric as cost function results into the Wasserstein-1 distance. Since the exact computation of the infimum is highly intractable, the Kantorovich-Rubinstein duality [30] is commonly used:

$$\mathcal{W}_1(P_r, P_{G_\theta}) = \sup_{\|f_\omega\|_L \leq 1} \left[\mathbb{E}_{x\sim P_r} f_\omega(x) - \mathbb{E}_{x\sim P_{G_\theta}} f_\omega(x) \right], \tag{2}$$

where $\|\cdot\|_L$ denotes the Lipschitz norm of a function. Replacing $\|f_\omega\|_L \leq 1$ with $\|f_\omega\|_L \leq K$ in the supremum for some constant K yields $K \cdot \mathcal{W}_1(P_r, P_{G_\theta})$ on the left side in (2). In contrast to discriminator learning, the idea now is to find a neural network critic $f_\omega : X \to \mathbb{R}$ that has parameters in the set $\Omega_K \triangleq \{\omega \mid f_\omega \ K\text{-Lipschitz}\}$ for constant $K > 0$ and that maximizes the difference in (2). It has been shown that indeed a solution to this maximization problem exists [15].

There exist several ways to ensure the Lipschitz property for DCNNs. One can ensure that ω lies in a compact space after each update by clipping the parameters to a fixed box, i.e. $\Omega = [-c, c]^l$, where the Lipschitz constant depends on the clipping parameter $c > 0$ and l denotes the number of network parameters. This approach likely causes instabilities if an already powerful critic is further maximized. The weights saturate around the clipping boundaries and turn as a consequence the critic into a naive function which does not fully exploit the potential of its complexity (cf. Figure 4, right). In contrast to clipping the weight space, a more plausible solution was proposed by Gulrajani et al. [16], where during maximization the property is used that a differentiable function is 1-Lipschitz if and only if the norm of the first derivative is bounded by 1. To recap this section, the proposed method is summarized in Algorithm 1.

3.3 Network Architectures

As crucial as loss function design of an unsupervised methods is the choice of an appropriate architecture for the critic and the generator network. For the critic a decoder is built following the DCGAN constraints in [25] with nearly 7×10^6 parameters. DCGANs are empirically proven to perform quite stable over a variation of different generative tasks and were also considered when the idea of Wasserstein GANs [15, 16] was introduced. For the generator network, two differing approaches were implemented in this study. The first generator is a ResNet18 [32] with a depth specific upsampling part taken from [8] (15.7×10^6 parameters) , where Godard et al. achieved state-of-the-art depth synthesis results in an unsupervised manner based on monocular videos and view synthesis. The second architecture is a U-net [14] with 14.2×10^6 parameters, which was initially proposed in [33] and is often considered in biomedical applications [34–36] Detailed information on the critic and the two generator networks is given in the supplementary material.

4 Experiments & Discussion

We implemented the steps proposed in Algorithm 1 with the publicly available TensorFlow framework [37] and its built-in Keras API [38]. The model is trained and evaluated on two strongly differing tasks. The applications are inner surface depth estimation of cylinder liners as well as depth synthesis of indoor environments based on the NYU Depth v2 data set [3], where

Algorithm 1 Cycle-consistent Wasserstein GAN, our proposed method

Require: α_D RGB-to-depth learning rate; α_C depth-to-RGB learning rate; c clipping parameter; n_f number of critic iterations; n_G number of generator updates; b minibatch size; λ_{cyc};
Require: $\omega_{\text{D}}, \omega_{\text{C}}$ initial critic weights, $\theta_{\text{D}}, \theta_{\text{C}}$ initial generator weights
  for $k = 1, \ldots, n_G$ **do**
    for $i = 1, \ldots, n_f$ **do**
      Sample $\{x_m\}_{m=1}^{b} \subset X$ and $\{y_n\}_{n=1}^{b} \subset Y$
      $\partial_D \leftarrow \nabla_{\omega_{\text{D}}} \frac{1}{b} \left[\sum_{m=1}^{b} f_{\omega_{\text{D}}}(G_{\theta_{\text{D}}}(x_m)) - \sum_{n=1}^{b} f_{\omega_{\text{D}}}(y_n) \right]$
      $\partial_C \leftarrow \nabla_{\omega_{\text{C}}} \frac{1}{b} \left[\sum_{n=1}^{b} f_{\omega_{\text{C}}}(G_{\theta_{\text{C}}}(y_n)) - \sum_{m=1}^{b} f_{\omega_{\text{C}}}(x_m) \right]$
      $\omega_{\text{D}} \leftarrow \text{clip}\left[\omega_{\text{D}} - \alpha_D \cdot \text{RMSprop}(\omega_{\text{D}}, \partial_D), -c, c\right]$
      $\omega_{\text{C}} \leftarrow \text{clip}\left[\omega_{\text{C}} - \alpha_C \cdot \text{RMSprop}(\omega_{\text{C}}, \partial_C), -c, c\right]$
    end for
    Sample $\{x_m\}_{m=1}^{b} \subset X$ and $\{y_n\}_{n=1}^{b} \subset Y$
    $\partial_D \leftarrow \nabla_{\theta_{\text{D}}} \frac{1}{b} \big[- \sum_{m=1}^{b} f_{\omega_{\text{D}}}(G_{\theta_{\text{D}}}(x_m))$
      $+\lambda_{\text{cyc}} \sum_{m=1}^{b} \| x_m - G_{\theta_{\text{C}}}(G_{\theta_{\text{D}}}(x_m)) \|_1$
      $+\lambda_{\text{cyc}} \sum_{n=1}^{b} \| y_n - G_{\theta_{\text{D}}}(G_{\theta_{\text{C}}}(y_n)) \|_1 \big]$
    $\partial_C \leftarrow$ analogous
    $\theta_{\text{D}} \leftarrow \theta_{\text{D}} - \alpha_D \cdot \text{RMSprop}(\theta_{\text{D}}, \partial_D)$
    $\theta_{\text{C}} \leftarrow \theta_{\text{C}} - \alpha_C \cdot \text{RMSprop}(\theta_{\text{C}}, \partial_C)$
  end for

no aligned data pairs were used for both settings. In the following, both data sets are briefly described and the results are presented separately for each. At the end of this section, a discussion on the general observed training characteristics and on the influence of hyperparamters in Algorithm 1 is supplemented.

4.1 Surface Depth

In this study, the same large engine database as initially proposed in [10] is used for depth estimation of cylinder inner surfaces. The examined objects are two type 6 gas engines from INNIO Jenbacher GmbH & Co OG. Each engine consists of 24 cylinders with a displacement of $6\,\text{dm}^3$. Depth measurements face a spatial region of $1.9 \times 1.9\,\text{mm}^2$, have a dimension of nearly 3100×3100 pixels and are acquired with a resource-intensive logistic chain as discussed in the Introduction and in [10]. The profiles denote (in contrast to the NYU Depth v2 data set) relative depth with respect to the core area of the surface. Utilizing a simple handheld microscope, the RGB data is taken from the same cylinder surfaces, where the measurements cover a significantly larger spatial region of $4.2 \times 4.2\,\text{mm}^2$ and have a resolution of nearly 1024×1024 pixels. Measurement positions are not aligned with the depth data. 243 samples are obtained for each modality, where RGB and depth data gets shuffled and augmented separately to nearly 3000 samples via random cropping and flipping. Furthermore, to make computation feasible by our *NVIDIA GeForce RTX 2080* GPU, each sample is resized to dimension 256×256.

In order to asses visual quality between two completely unaligned domains, we further generated depth profiles of some additional cylinders and tried at highly effort to record RGB data of nearly the same area on point millimeter range with the handheld device. These modality samples are not included in the training database. As we have no access to pixel-wise aligned depth profiles, performance of the model can only be assessed visually on these additional samples. We also report a metric which is commonly used in engine liner surfacing based on *bearing load*

curves. The metric computes the MAE between the pixel distributions of two depth maps and is further discussed in [10] and in the supplementary material. Experiments indicated, that this metric correlates with image quality in the targeted application.

Various experiments have been performed to find appropriate settings for the hyper-parameters in Algorithm 1. During evaluation we set the number of generator updates n_G to 16×10^3 and the minibatch size b equals 6. The number of critic iterations n_f is initially set to 20 to ensure a good approximation of the Wasserstein-1 distance in the beginning and gets halfed after 2000 generator updates to speed up convergence. Convergence is referred to as when both critic losses, which indicate the approximated Wasserstein-1 distance between generator outputs and real domain distributions, trend towards zero. Furthermore, we observed that it is important to include normalization techniques into the critic network. Without normalization, the critic suffered from vanishing or exploding gradients, depending if the choice for the clipping value c was too low or too high, respectively. The considered normalization layers apply Instance normalization [39] between convolution and activation function. We also considered the use of normalization in the generator networks but noticed that this disadvantages the image quality crucially.

A huge impact on convergence have also the learning rates for the generator and the critic on both domains. In our first attempts we chose for the critics a learning rate of 5×10^{-5} and for the generators a rate of 7×10^{-5} but we realized quickly that is not possible to obtain simultaneous convergence for both directions using joint learning rates (cf. Figure 3). Therefore we decided to choose separate learning rates α_D, α_C for each direction, with $\alpha_D = 5 \times 10^{-5}$ and $\alpha_C = 9 \times 10^{-5}$. Basically, learning rates for discriminator and generator within one direction should also be fine-tuned separately. If the common rate is too high, the critic might not be trained to optimality, and if the rate is too small, generator might converge only slowly. In our experiments we decided to choose the best compromise between these two scenarios to avoid additional complexity during hyper-parameter search.

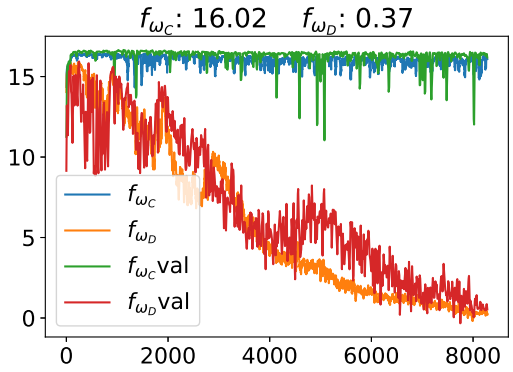

Figure 3: Visualization of the scores of the critic networks (i.e. the approximated Wasserstein-1 distance) for both domains. The scores are additionally evaluated on the unseen evaluation data to detect critic overfitting. Although a small learning rate leads to stable convergence of f_{ω_D} with slight overfitting, f_{ω_C} hardly converges.

Convergence of both generators is also sensitive to the choice of λ_{cyc} which triggers the influence of the cycle-consistency term in Algorithm 1. A small choice for this parameter results into less structural correspondence between input and output. A high choice seduces the generators to neglect the adversarial term in the beginning. To address this conflict, we set λ_{cyc} to

0 at training start and linearly increase the value to 20 with respect to n_G. Results for varying generator architectures (ResNet18, U-net) and different clipping parameters c are presented in Table 1. The critic weight vectors $\omega_{\mathrm{D}}, \omega_{\mathrm{C}}$ are initialized via normal distribution with mean 0 and standard deviation $c/2$.

Table 1: Results for selected generator architectures and different clipping values. The reported value is the mean absolute distance between pixel-distributions of the ground truth and the synthesized depth, evaluated on unseen data (smaller is better).

Generator	$c = 0.002$	$c = 0.003$	$c = 0.005$
w.o. Dropout			
ResNet18	0.178	0.171	0.173
U-Net	0.170	0.140	0.171
Dropout rate 0.25			
ResNet18	0.195	0.152	0.155
U-Net	0.160	0.176	**0.123**

As already discussed in Section 3.2, weight clipping can cause concentration of the critic weights at the boundaries and therefore turn the critic into a naive function [16]. We also observed this behavior. Experiments showed, that the insertion of Dropout layers [40] after each convolution in the critic significantly reduced the saturation of the weights at the clipping bounds and therefore helped the critic to exploit its complexity more (cf. Figure 4). The Dropout rate in the experiments equals 0.25.

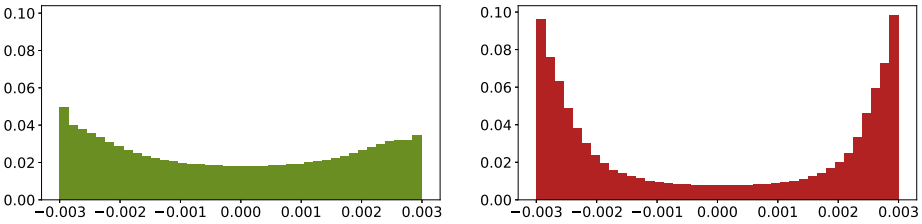

Figure 4: Influence of Dropout layers on the distribution of the clipped critic weights. The weights of the critic with Dropout (left) are less concentrated at the clipping boundaries compared to the critic without Dropout (right).

4.2 NYU Depth

The NYU Depth v2 data set is one of the largest RGB-D data sets for indoor scene depth synthesis. The raw data set consist of 464 scenes, where we sample a subset of equally-spaced frames out of each sequence and center-crop each image to dimension 448×576. This results into approximately 22k images per modality. The RGB and depth frames are basically not aligned due to different frame rates. Due to the reason we test here a unsupervised approach, no further processing on the sequences is necessary. Depth profiles contain information on absolute depth, i.e. each pixel corresponds to the distance between camera and object. Similar to Section

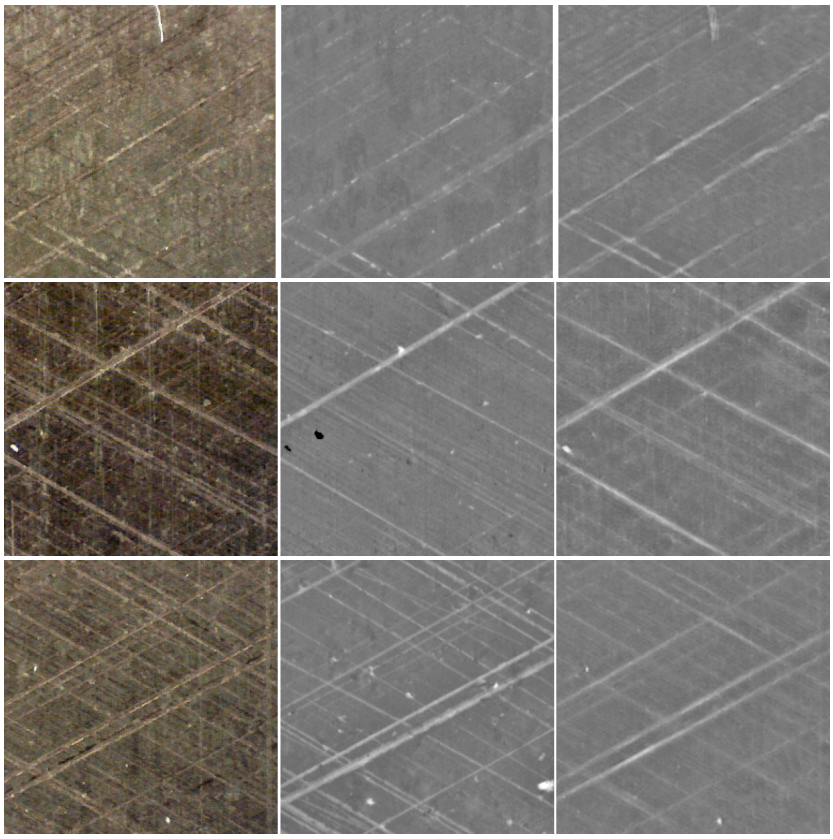

Figure 5: From left to right: RGB input, ground truth profiles with relative depth (of nearly the same position), and synthesized depth profile, generated with a U-net and trained with clipping parameter 0.005. More images can be found in the supplementary material.

4.1, RGB and depth data is shuffled separately and resized to 224×288 to fit computational capacities. For qualitative and quantitative evaluation of our proposed model we make use of the official test subset of 654 pixel-wise aligned modality pairs, which were not included during training.

Many observations from the previous task on hyper-parameter and regularization settings were decisive for fine tuning the model in Algorithm 1 on this data set. The number of generator updates n_G equals again 16×10^3 and the minibatch size b is set to 6. The number of critic iterations n_f is also set to 20 in the beginning gets halfed after 2000 generator updates. Instance normalization is applied after each convolution layer in the critic networks as well as in the generators. The learning rates for the RGB-to-depth and the depth-to-RGB direction are defined by $\alpha_D = 6 \times 10^{-5}$ and $\alpha_C = 9 \times 10^{-5}$, respectively. Again, we observed that the use of Dropout layers with Dropout rate 0.15 helped the critic to further exploit its complexity despite of weight clipping. We adopt the technique to linearly increase λ_{cyc} from previous task, where λ_{cyc} equals 2 at the end of the training phase.

We also implemented the cycle-consistent Wasserstein GAN with gradient penalty, where in Algorithm 1 the clipping operation is omitted and the critic gradient is defined as follows:

$$\partial \leftarrow \nabla_\omega \frac{1}{b}\Bigg[\sum_{i=1}^{b} f_\omega(G_\theta(x_i)) - \sum_{i=1}^{b} f_\omega(y_i) + p\sum_{i=1}^{b}\max\{0, \|\nabla_{\tilde{x}_i} f_\omega(\tilde{x}_i)\|_2 - 1\}^2\Bigg],$$

where $\tilde{x}_i \triangleq \epsilon \cdot G_\theta(x_i) + (1-\epsilon)\cdot y_i$ for $\epsilon \sim \mathcal{U}(0,1)$ and p denotes the influence of the gradient penalty [16]. We only implemented the gradient penalty for the U-net architecture, were we observed only little improvement compared to weight clipping (cf. Table 2).

Table 2: Results for selected generator architectures and different clipping values. Also optimization via gradient penalty (GP) is applied to the U-net architecture. Reported values are the root mean squared error and the accuracy $\delta < 1.25$ (RMSE/δ_1). Results are compared to two well-known supervised approaches (there hardly exist unsupervised results for the NYU v2 Depth data set).

	ResNet18	U-net	Benchmark
$c = 0.002$	**0.931/0.434**	1.102/0.346	
$c = 0.003$	0.934/0.419	1.179/0.291	
$c = 0.005$	1.285/0.245	1.347/0.325	
GP $p = .1$		1.072/0.375	
GP $p = 1$		1.082/0.404	
Eigen [4]			0.907/0.611
Laina [7]			0.573/0.811

As we observe in Table 2, the results of proposed model (with ResNet18 architecture) on unseen data are promising, but not better than established supervised models. The experiments should show that it is possible to achieve sufficient depth synthesis of real scenes even when aligned data, stereo images and video sequences are not available. We are quite confident that further fine-tuning on the generator architectures as well as variation of the learning rates and clipping parameters for each direction separately improves performance. More visualizations and also plots from the depth-to-RGB direction can be found in the supplementary material.

4.3 Discussion

The cycle-consistent Wasserstein GAN proposed in Algorithm 1 was trained on two very different applications taken from the area of unsupervised single-image depth synthesis. Various experiments and research on different architectures, regularization and hyper-parameter settings showed important aspects in both applications that definitely should be considered during training of a cycle-consistent Wasserstein GAN:

- To be able to really use the advantage of the cycle-consistency constraint, convergence must be ensured for both directions. Therefore, variation of the learning rates for both tasks separately is of great benefit.
- Clipping critic weights after each update significantly restricts potential of the critic network. Although the approach of gradient penalty [16] also enforces the Lipschitz property

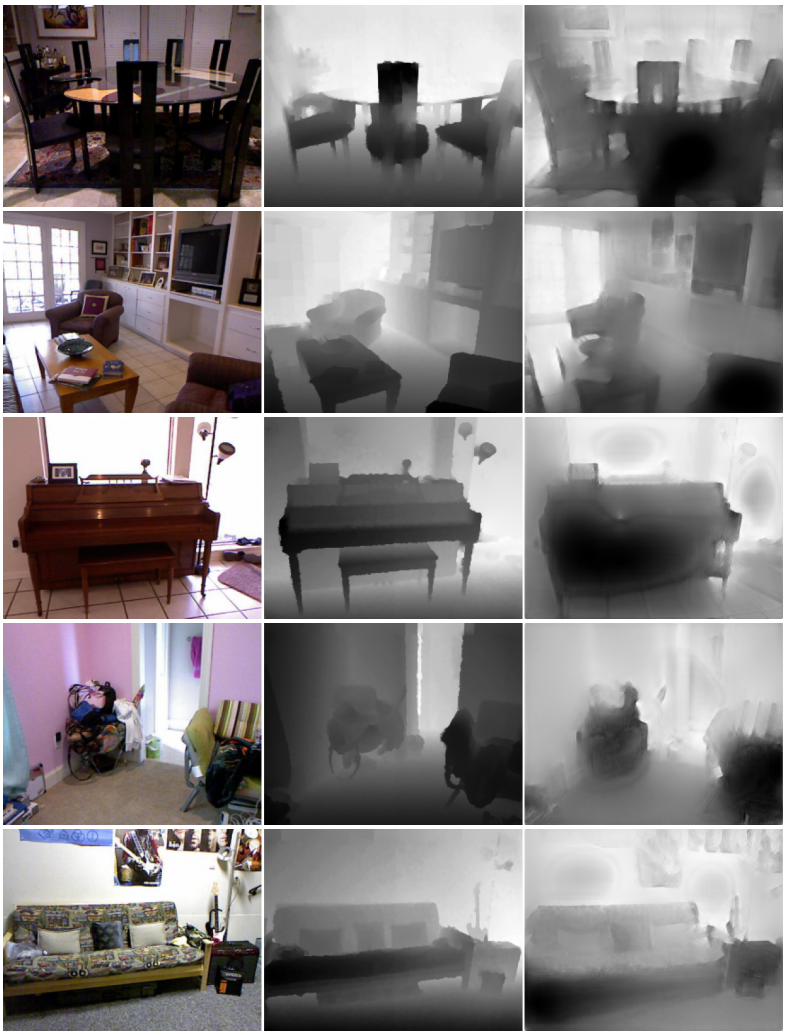

Figure 6: Left to right: RGB input, aligned ground truth profiles with absolute depth, and synthesized depth profile, generated with a ResNet18 and trained with clipping parameter 0.002. More images can be found in the supplementary material.

of critic networks, one has to be aware that the gradient penalty causes further fine-tuning and increasing computational complexity. Our suggestion is to make use of Dropout layers in the critic network, which significantly helps to decrease the concentrations at the clipping boundaries.

- Since the critic networks do not learn a probability like discriminators but an unbounded score, it is very important to make use of normalization in the critic. Otherwise, fine-tuning on the clipping parameter might always lead to vanishing or exploding gradients. For a small minibatch-size (as in our experiments) we suggest the usage of Instance normalization [39] or Group normalization [41], which are techniques that operate on each

input instance separately but not on the whole minibatch as in Batch normalization [42].

- Learning depth in an adversarial manner is a quite demanding task and needs during training a lot of time to develop a confident critic, especially in the beginning. Using a high penalty λ_{cyc} for the cycle-consistency term seduces the generators to ignore the adversarial part and the generators conduct a simple mapping to preserve structural properties (e.g. RGB-to-depth generator outputs grayscaled versions of the input). Therefore, we suggest to choose a very small penalty λ_{cyc} for the cycle-consistency term at the beginning which forces generators to map into the correct domain. The penalty then should be steadily increased during training to exploit the cycle-consistency constraint in a later phase of training.

5 Conclusion

In this work, the architecture of a cycle-consistent GAN with Wasserstein loss is proposed and applied to two different depth synthesis tasks. We observe that the approach succeeds on estimation of relative depth of cylinder liner surfaces. Moreover, the proposed algorithm also shows promising results when applied to the task of absolute depth estimation of indoor spaces. Through a comprehensive evaluation process, major complications during training cycle-consistent Wasserstein GANs were identified and workable solutions were found. One disadvantage, however, is that two generators must be fitted in parallel, although only one synthesis network is needed in the application. Future work will therefore include the development of one-sided depth synthesis models and also the application of our approach to other modality transfer tasks.

Funding

The authors would like to acknowledge the financial support of the ”COMET - Competence Centres for Excellent Technologies” Programme of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and the Federal Ministry for Digital and Economic Affairs (BMDW) and the Provinces of Styria, Tyrol and Vienna for the COMET Centre (K1) LEC EvoLET. The COMET Programme is managed by the Austrian Research Promotion Agency (FFG) [grant number 865843].

References

[1] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, “Monocular depth estimation based on deep learning: an overview,” *Science china technological sciences*, pp. 1–16, 2020.

[2] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving the KITTI vision benchmark suite,” in *Conference on computer vision and pattern recognition (CVPR)*, 2012.

[3] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.

[4] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” *Advances in neural information processing systems*, vol. 27, pp. 2366–2374, 2014.

[5] Y. Lei, J. Harms, T. Wang, Y. Liu, H.-K. Shu, A. B. Jani, W. J. Curran, H. Mao, T. Liu, and X. Yang, "MRI-only based synthetic CT generation using dense cycle consistent generative adversarial networks," *Medical physics*, vol. 46, no. 8, pp. 3565–3581, 2019.

[6] Wofk, Diana and Ma, Fangchang and Yang, Tien-Ju and Karaman, Sertac and Sze, Vivienne, "FastDepth: Fast Monocular Depth Estimation on Embedded Systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[7] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 fourth international conference on 3D vision (3DV)*, pp. 239–248, IEEE, 2016.

[8] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF international conference on computer cision*, pp. 3828–3838, 2019.

[9] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1851–1858, 2017.

[10] C. Angermann, S. Jónsson, M. Haltmeier, A. Moravová, C. Laubichler, C. Kiesling, M. Kober, and W. Fimml, "Machine learning for nondestructive wear assessment in large internal combustion engines," 2021. under submission.

[11] "Mic-fi digitial wi-fi microscope." `https://mic-fi.it/en/home`. Accessed: 2020-12-15.

[12] "Alicona infinitefocus confocal microscope." `https://www.alicona.com/en/products/infinitefocus/`. Accessed: 2020-12-15.

[13] 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* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[14] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE international conference on computer vision (ICCV)*, pp. 2242–2251, 2017.

[15] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th international conference on machine learning*, vol. 70, pp. 214–223, PMLR, 2017.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Advances in neural information processing systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[17] H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn, "Depth prediction from a single image with conditional adversarial networks," in *2017 IEEE international conference on image processing, ICIP 2017 - proceedings*, pp. 1717–1721, IEEE Computer Society, 2018.

[18] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 2002–2011, IEEE, 2018.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017.

[20] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279, 2017.

[21] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: geometry to the rescue," in *European conference on computer vision*, pp. 740–756, Springer, 2016.

[22] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: learning of structure and motion from video," 2017.

[23] X. Han, "MR-based synthetic CT generation using a deep convolutional neural network method," *Medical physics*, vol. 44, no. 4, pp. 1408–1419, 2017.

[24] Y. Hiasa, Y. Otake, M. Takao, T. Matsuoka, K. Takashima, A. Carass, J. L. Prince, N. Sugano, and Y. Sato, "Cross-modality image synthesis from unpaired data using cyclegan," in *International workshop on simulation and synthesis in medical imaging*, pp. 31–41, Springer, 2018.

[25] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015.

[26] S. Zhao, H. Fu, M. Gong, and D. Tao, "Geometry-aware symmetric domain adaptation for monocular depth estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9788–9798, 2019.

[27] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *2018 international conference on 3D vision (3DV)*, pp. 587–595, IEEE, 2018.

[28] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, "Adadepth: unsupervised content congruent adaptation for depth estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2656–2665, 2018.

[29] A. Bhoi, "Monocular depth estimation: a survey," 2019.

[30] C. Villani, *Optimal transport: old and new*, vol. 338. Springer science & business media, 2008.

[31] J. Adler, A. Ringh, O. Öktem, and J. Karlsson, "Learning to solve inverse problems using wasserstein loss," 2017.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention – MICCAI 2015*, (Cham), pp. 234–241, Springer International Publishing, 2015.

[34] C. Angermann, M. Haltmeier, R. Steiger, S. Pereverzyev, and E. Gizewski, "Projection-based 2.5D U-net architecture for fast volumetric segmentation," in *2019 13th international conference on sampling theory and applications (SampTA)*, pp. 1–5, IEEE, 2019.

[35] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Medical Image Analysis*, vol. 63, p. 101693, 2020.

[36] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo, "Deep learning for undersampled MRI reconstruction," *Physics in Medicine & Biology*, vol. 63, p. 135007, jun 2018.

[37] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

[38] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[39] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2017.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of machine learning research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[41] Y. Wu and K. He, "Group normalization," in *Proceedings of the european conference on computer vision (ECCV)*, pp. 3–19, 2018.

[42] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd international conference on machine learning*, vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, PMLR, 07–09 Jul 2015.

[43] R. B. Obara and A. Sinatora, "Quantification of cylinder bores almost zero-wear," *Wear*, vol. 364, pp. 224–232, 2016.

[44] F. Cabanettes and B. G. Rosén, "Topography changes observation during running-in of rolling contacts," *Wear*, vol. 315, no. 1-2, pp. 78–86, 2014.

[45] W. P. Dong, E. J. Davis, D. L. Butler, and K. J. Stout, "Topographic features of cylinder liners — an application of three-dimensional characterization techniques," *Tribology international*, vol. 28, no. 7, pp. 453–463, 1995.

Supplementary Material

The following sections provide further information on definitions and results that were only briefly addressed or presented in the manuscript. To be more precise, engine liner surface characterization via bearing load curves is discussed and detailed descriptions of the architectures of the critic and generator networks are given. The third part contains visualizations for both targeted applications (surface depth and NYU Depth v2), with synthesized images of the contrarian generator also added.

Bearing Load Curve

The bearing load curve (BLC) - also known as Abbott-Firestone curve - is an important tool for assessing surface structure conditions [10, 43, 44]. As illustrated in Figure S1, the BLC plots the height value against the percentage of measurement points above that height. In terms of cylinder liner functionality, the BLC describes two important parts of the surface structure: the valley part, which specifies the oil retention capacity, and the peak part, which specifies debris or asperities of the surface structure [45].

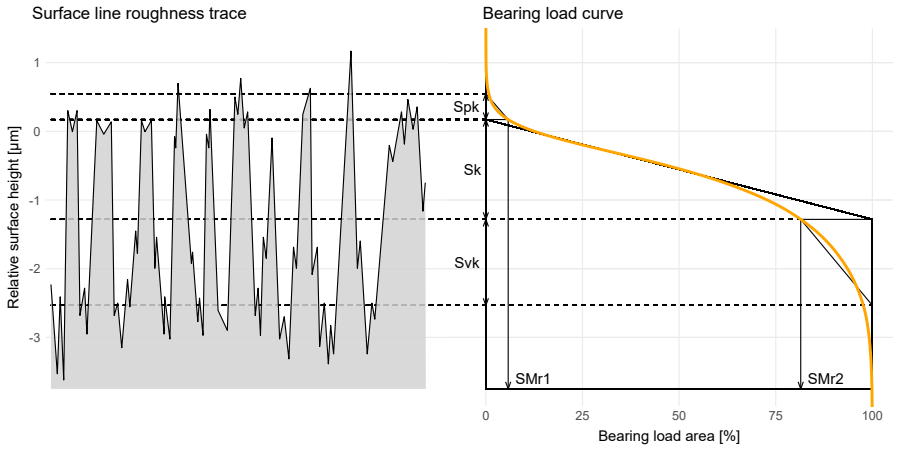

Figure S1: Visualization of a surface line roughness trace and the corresponding bearing load curve.

Let $A \in \mathbb{R}^{d_1 \times d_2 \times 1}$ denote a depth profile. The BLC is defined as the reversed empirical quantile function $B\colon (0,1) \to \mathbb{R}$ of pixel values of the depth image:

$$B(x) \triangleq \inf \left\{ t \in \mathbb{R} \;\middle|\; 1 - x \leq \frac{1}{d_1 \cdot d_2} \sum_{A} \mathbb{1}\{a_i \leq t\} \right\} .$$

where the sum runs over all pixels a_i in depth image A. For calculation purposes, the discretized BLC $b \in \mathbb{R}^K$ is considered, obtained after sampling the left-continuous BLC B at K equidistant sampling positions. For bearing load curves b and $\hat{b}$ of a measured and a synthesized depth profile, respectively, the following distance can be calculated:

$$\ell(b, \hat{b}) = \frac{1}{K} \sum_{k=1}^{K} \left| b_k - \hat{b}_k \right| . \tag{S1}$$

As thoroughly discussed in [10], the component-wise absolute distance between both curves is a proper approximation to the Wasserstein-1 distance between the underlying two depth images. Therefore, the distance in (S1) works well as metric for comparing surface depth profiles with synthesized instances that are not pixelwise aligned.

Network Details

In the following, **k** is the kernel size, **s** the stride, and **chns** the number of layer output channels. **Input** corresponds to the input of each layer. Network input and output are denoted by $\mathcal{I}$ and $\mathcal{O}$, respectively.

Table S1: **ResNet18 generator.** The encoder corresponds exactly to the illustrated architecture in [32, Table 1]. The table only describes the decoder architecture which is a slightly modified version of [8]. The output activation function depends on the task (e.g. tangens hyperbolicus for RGB/relative depth, ReLU for absolute depth). The number of weights equals 15.7×10^6. For upsampling (upsam.), nearest neighbor method is used.

name	**type**	**k**	**s**	**chns**	**input**	**activ.**
c61	conv.	4	1	256	conv5_4	ELU
up1	upsam.		2	256	c61	
con	concat			512	up1,conv4_4	
c62	conv.	3	1	256	con	ELU
c71	conv.	4	1	128	c62	ELU
up2	upsam.		2	128	c71	
con	concat			256	up2,conv3_4	
c72	conv.	3	1	128	con	ELU
c81	conv.	4	1	64	c72	ELU
up3	upsam.		2	64	c81	
con	concat			128	up3,conv2_4	
c82	conv.	3	1	64	con	ELU
c91	conv.	4	1	64	c82	ELU
up4	upsam.		2	64	c91	
con	concat			128	up4,conv1	
c92	conv.	3	1	64	con	ELU
c01	conv.	4	1	32	c92	ELU
up5	upsam.		2	32	c01	
c02	conv.	3	1	32	up5	ELU
$\mathcal{O}$	conv.	3	1	$3/1$	c02	task

Table S2: **U-net generator.** The output activation function depends on the task. For upsampling, transposed convolution (transp.) is used. The number of weights equals 14.2×10^6.

name	**type**	**k**	**s**	**chns**	**input**	**activation**
c11	conv.	3	1	48	$\mathcal{I}$	ReLU
c12	conv.	3	1	48	c11	ReLU
c13	conv.	3	2	48	c12	ReLU
c21	conv.	3	1	96	c13	ReLU
c22	conv.	3	1	96	c21	ReLU
c23	conv.	3	2	96	c22	ReLU
c31	conv.	3	1	192	c23	ReLU
c32	conv.	3	1	192	c31	ReLU
c33	conv.	3	2	192	c32	ReLU
c41	conv.	3	1	384	c33	ReLU
c42	conv.	3	1	384	c41	ReLU
c43	conv.	3	2	384	c42	ReLU
c51	conv.	3	1	768	c43	ReLU
c52	conv.	3	1	768	c51	ReLU
up1	transp.	4	2	384	c52	ReLU
con	concat			768	up1,c42	
c61	conv.	3	1	384	con	ReLU
up2	transp.	4	2	192	c61	ReLU
con	concat			384	up2,c32	
c71	conv.	3	1	192	con	ReLU
up3	transp.	4	2	96	c71	ReLU
con	concat			192	up3,c22	
c81	conv.	3	1	96	con	ReLU
up4	transp.	4	2	48	c81	ReLU
con	concat			96	up4,c12	
c91	conv.	3	1	48	con	ReLU
$\mathcal{O}$	conv.	3	1	3/1	c91	task

Table S3: **Critic network.** Every convolution layer applies Instance normalization [39] before activation function. LReLU denotes the Leaky ReLU activation function with slope parameter 0.2. The number of weights equals 7×10^6.

name	**type**	**k**	**s**	**chns**	**input**	**activation**
c1	conv.	4	2	64	$\mathcal{I}$	LReLU
d1	dropout			64	c1	
c2	conv.	4	2	128	d1	LReLU
d2	dropout			128	c2	
c3	conv.	4	2	256	d2	LReLU
d3	dropout			256	c3	
c4	conv.	4	2	512	d3	LReLU
d4	dropout			512	c4	
c5	conv.	4	2	512	d4	LReLU
d5	dropout			512	c5	
$\mathcal{O}$	conv.	4	1	1	d5	linear

Further Visualizations

Figures S2 and S4 visualize further results of the proposed cycle-consistent Wasserstein GAN, evaluated on unseen RGB frames of the surface depth database and the NYU Depth v2 data set. Colorized instances of the contrary depth-to-RGB generators, which ensure that synthesized depth does not become irrelevant to the input, are also added. Especially with the NYU Depth v2 data set we recognize that the contrary generator is far away from synthesizing realistic RGB images (cf. Figure S5, which suggests that for this dataset further research on the generator architectures is necessary.

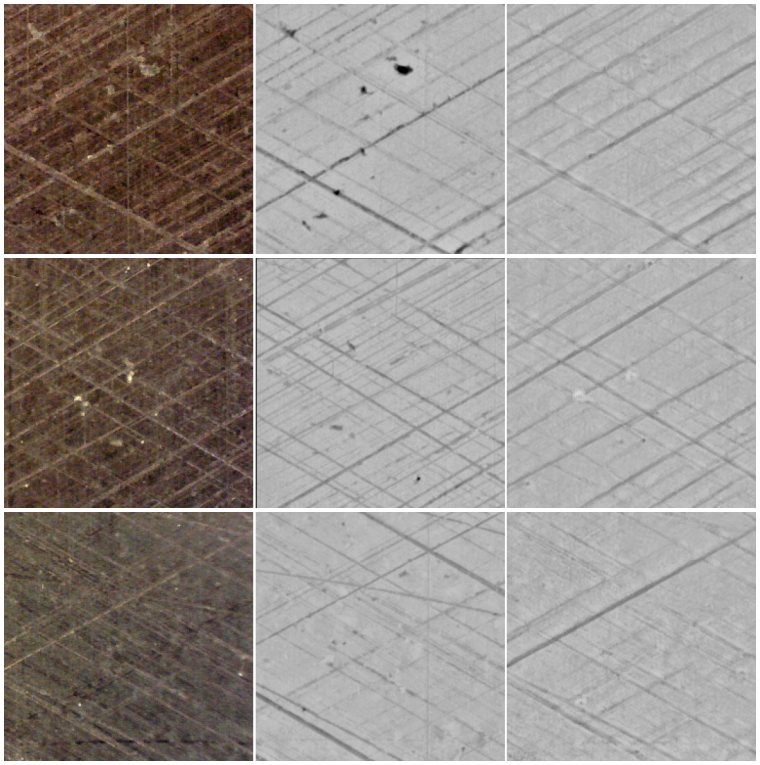

Figure S2: **Surface depth**
Left to right: RGB input, unaligned ground truth profiles with relative depth (of nearly the same position), and synthesized depth profile.

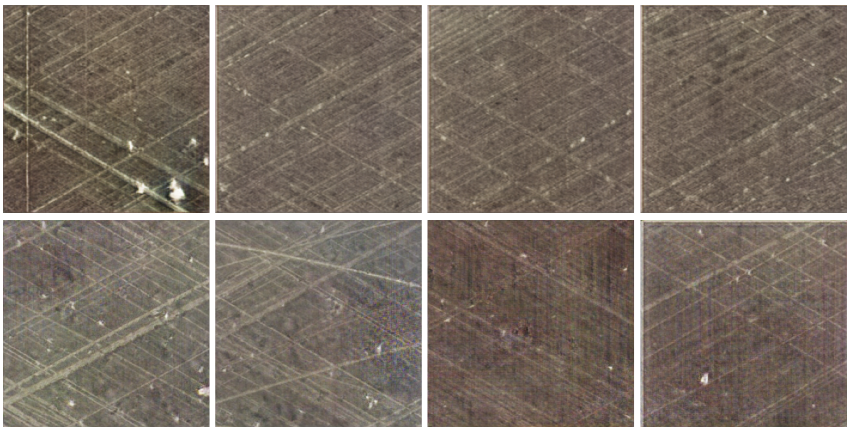

Figure S3: **Surface depth**
Colorized instances of the contrary depth-to-RGB generator, which synthesizes RGB images out of surface depth profiles.

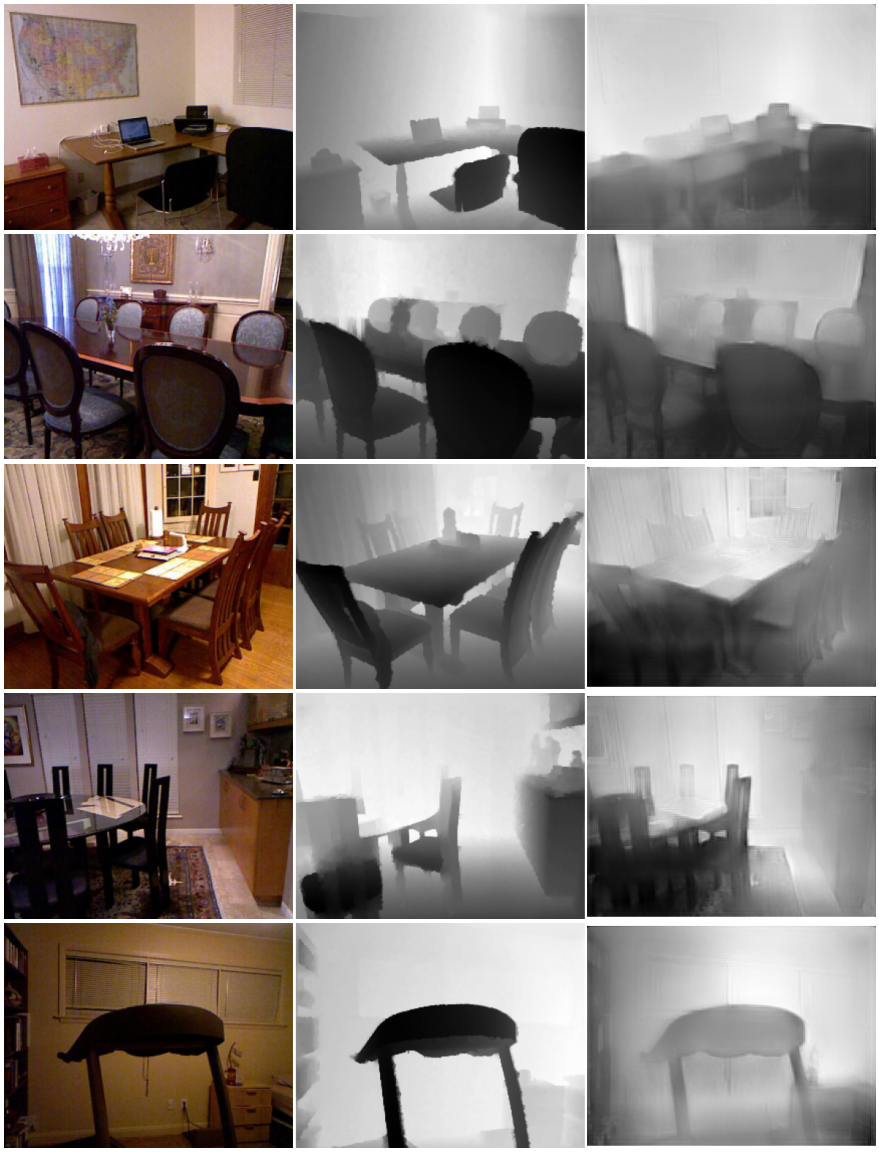

Figure S4: **NYU Depth v2**
Left to right: RGB input, aligned ground truth profiles with absolute depth, and synthesized depth profile.

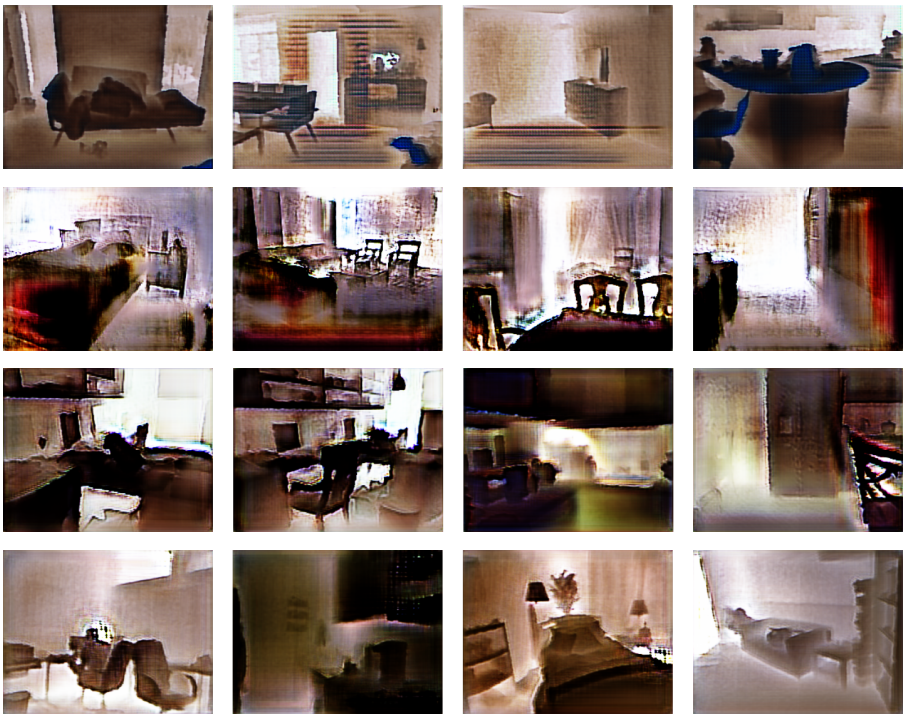

Figure S5: **NYU Depth v2**
Colorized instances of the contrary depth-to-RGB generator, which synthesizes RGB images out of indoor depth scenes.