Pattern Recognition 91 (2019) 232–244

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Generative adversarial framework for depth filling via Wasserstein metric, cosine transform and domain transfer

Amir Atapour-Abarghouei*, Samet Akcay, Grégoire Payen de La Garanderie, Toby P. Breckon

Department of Computer Science, Durham University, UK

ARTICLE INFO

Article history:
Received 1 March 2018
Revised 18 December 2018
Accepted 17 February 2019
Available online 27 February 2019

Keywords:
Depth image
Hole filling
Self-supervised learning
Generative model
Adversarial training
Feature distance
Domain adaptation

ABSTRACT

In this work, the issue of depth filling is addressed using a self-supervised feature learning model that predicts missing depth pixel values based on the context and structure of the scene. A fully-convolutional generative model is conditioned on the available depth information and full RGB colour information from the scene and trained in an adversarial fashion to complete scene depth. Since ground truth depth is not readily available, synthetic data is instead used with a separate model developed to predict where holes would appear in a sensed (non-synthetic) depth image based on the contents of the RGB image. The resulting synthetic data with realistic holes is utilized in training the depth filling model which makes joint use of a reconstruction loss which employs the Discrete Cosine Transform for more realistic outputs, an adversarial loss which measures the distribution distances via the Wasserstein metric and a bottleneck feature loss that aids in better contextual feature execration. Additionally, the model is adversarially adapted to perform well on naturally-obtained data with no available ground truth. Qualitative and quantitative evaluations demonstrate the efficacy of the approach compared to contemporary depth filling techniques. The strength of the feature learning capabilities of the resulting deep network model is also demonstrated by performing the task of monocular depth estimation using our pre-trained depth hole filling model as the initialization for subsequent transfer learning.

1. Introduction

The world is visually diverse, irregular and contrastingly structured at the same time. Three-dimensional scenes containing depth information are highly applicable within visual systems such as autonomous driving, augmented reality, environment modelling and alike. Moreover, recent achievements in depth capture technologies, including time-of-flight cameras, stereo correspondence and structured light devices, have made depth accessible in any scene understanding process. However, complete (hole-free) scene depth cannot be acquired facilely using commercial devices and even high-performance depth sensing solutions suffer from a range of environmental noise issues that preclude the recovery of hole-free scene depth under all conditions. This work is an exploration into whether a state-of-the-art learning based approach is capable of understanding the structures and intricacies of a scene, just as humans are, to predict the missing parts of scene depth as a standalone real-time portion of any visual system.

Image completion is considered challenging as it is inherently ill-posed. RGB completion approaches can achieve plausible results, using either local or non-local information [1–5]. However, due to the differences between depth and colour images (e.g., absence of granular texture, object separation, and in-scene transferability of varying depth sub-regions), conventional colour image inpainting is considerably less effective within the depth modality [6].

Some depth filling techniques leverage classic image inpainting approaches to complete depth [7]. There have also been attempts to fill a target region in one of a set of multi-view photographs [8], to fill depth using exemplar-based image completion [9], and a myriad of approaches utilizing filters [10], temporal-based methods [11], reconstruction-based methods [12], and others [13–15].

Deep neural networks have recently been successfully utilized for image stylization [16,17], super-resolution [18–20], and colorization [21]. In the realm of image completion, Pathak et al. [22] propose a context encoder which can predict missing regions in a colour image using an adversarial [23] and a reconstruction loss (ℓ_2). Although the model produces promising results, the absence of fine texture and the existence of visible artefacts near the boundaries of the target region point to flaws in the learning mechanism within the framework.

* Corresponding author.
E-mail addresses: amir.atapour-abarghouei@durham.ac.uk (A. Atapour-Abarghouei), toby.breckon@durham.ac.uk (T.P. Breckon).

https://doi.org/10.1016/j.patcog.2019.02.010
0031-3203/© 2019 Elsevier Ltd. All rights reserved.

In a related work, Yeh et al. [24] utilize an analogous framework with similar loss functions to map the input image with missing or corrupted regions to a latent vector, which in turn is passed through their generator that recovers the target content. Despite the large amount of corruption applied to the input, the model generates perceptually plausible outputs. Nevertheless, blurring effects and unwanted artefacts persist in spite of the low resolution of the images.

Yang et al. [25] propose a joint optimization framework composed of two separate networks, a content encoder, based on [22], which is tasked to preserve contextual structures within the image, and a texture network, which enforces similarity of the fine texture within and without the target region using neural patches [26]. The model is capable of completing higher resolution images than its two earlier counterparts [22,24] but at the cost of greater inference time since the final output is not achievable via a single forward pass.

Regarding depth images, advances have been made in monocular depth estimation [27–31] and depth super-resolution [32]. Here, we utilize a generative model trained on synthetic data [33,34] to complete depth. Since the model is expected to synthesize large portions of depth, it has to adapt to learning image structures and semantics. In existing works on learning-based colour image completion [22,24,25], training requires large datasets. The complete image is often considered as the ground truth, and the input is created by adding noise or sparse corruptions [24], removing rectangular blocks [22,24,25], or cutting random regions from the image [22]. In the realm of depth filling, however, no such large datasets exist that contain large quantities of ground truth (hole-free) depth. Consequently, synthetic data needs to be acquired from a graphically rendered virtual environment primarily designed for a gaming application [35].

Since depth holes are neither random nor manually created, they are *predictable*, in that they occur due to specific scene features or the capture device. For instance, featureless surfaces such as blank walls and roads, reflective objects, and depth discontinuities, among others can cause depth holes. As a result of this *predictability*, the location of a hole occurrence can be learned via a separate model trained to predict where holes would be in a depth image based on the features present in the scene and the assumption of a specific capture approach.

When high-quality ground truth exists, a model can be naively trained based on a simplistic reconstruction loss (ℓ_1 or ℓ_2). However, due to the multi-modality of image completion, a model trained in this way tends to generate the average of the multiple possible modes in the predictions, which results in an output containing blurring effects. This is why the techniques in [22,24,25] and other generative models [36,37] leverage adversarial training [23] as this assists with mode selection to generate realistic results. However, approaches using Generative Adversarial Networks (GAN) [22–25] suffer from certain flaws such as unstable training, difficulties in reaching an equilibrium, and vanishing gradients due to premature discriminator optimality and other issues [38–40]. Here, we utilize an improved adversarial framework [39] that avoids such issues.

Even though an adversarial loss can help diffuse blurring effects, the goal of the adversary should be generating a more realistic image across the board and blurring artefacts still occasionally make their way to the output. This is because the generator feels safer averaging than selecting values. To ease the burden of de-blurring on the adversary, we propose the addition of a loss term based on the Discrete Cosine Transform (DCT) in addition to the conventional ℓ_1 loss. The DCT preserves an accurate representation of the image structure in its spatial frequency content, which is why it has long been used in de-blurring [41], compression [42] and alike. We utilize the absolute deviations loss (ℓ_1) in the frequency domain, as this error is far more obvious when the DCT is applied to a blurry averaged image. As seen in Fig. 1, the ℓ_1 distance between the original image and the blurry image, both in the spatial domain, is not very large, but when the same images are transformed into the frequency domain using the DCT, the ℓ_1 error is much larger and therefore a better indicator of blurring effects.

The task of our generator consists of two stages: reducing the input into a compact representation of itself in the feature space (encoding) and reconstructing the image from these compact features (decoding). Up-convolutions, of any kind, are fraught with intrinsic unpredictability and can lead to bad salient edges and absence of fine texture. As a result, ensuring that the reconstruction starts from a correct and viable feature representation is paramount. We use the feature representations produced in the generator bottleneck in our loss to make sure the scene representation is correctly captured before reconstruction. While the sole use of this as a loss function is inadvisable and can lead to high-frequency artefacts, it is a helpful complement to the reconstruction and adversarial losses.

Our approach is meant to fill holes in depth images acquired via commercially and computationally inexpensive tools (a stereo camera and established stereo correspondence approaches such as [43,44]) and not in pixel-perfect synthetic depth images only. Therefore, as part of our training procedure, it is vital to guide the model toward capturing the distribution of the natural data. With this in mind, a domain transfer network is trained within the framework to rectify the model such that real-world images can be viable inputs during inference. In short, the contributions of our work are as follows:

- *Novelty* - no comparable approach utilizes a generative model using the Earth Mover's distance to complete depth via the Discrete Cosine Transform based on a synthetic training corpus with predicted holes.
- *Accuracy* - the approach is far more efficient and accurate than comparators (conventional image completion techniques) within a side-by-side comparison framework (Tables 2 and 3; Figs. 10 and 11).
- *Representation Learning* - our model is capable of learning better semantics and context as illustrated by superior sharp and artefact-free qualitative outputs when performing monocular depth estimation (Figs. 12 and 13).

In the following section, we present an overview of the literature relevant to this work. Section 3 provides a discussion on the

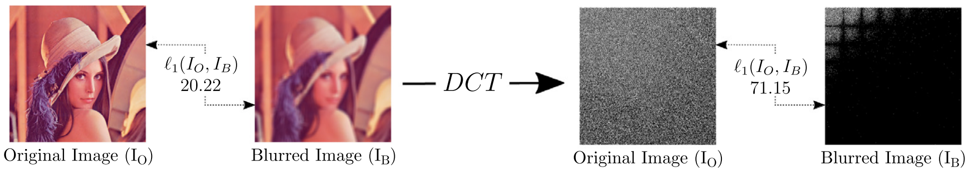

Fig. 1. A demonstration of how the DCT makes the absolute deviations loss more susceptible to blurring. Note that the ℓ_1 distance between the images in the frequency domain is higher and therefore a great tool to identify blurring.

data preparation process and Section 4 contains a detailed outline of the proposed hole filling approach. Results are evaluated in Section 5 and the work is finally concluded in Section 6.

2. Related work

There have recently been remarkable strides made in complex learning-based computer vision problems such as image classification [45–50], semantic segmentation [51–53], and image generation [23,36,38,39,54–56]. Inspired by the capabilities of recent generative models [22,23,36,38,39], we attempt to complete depth images by learning the details of a scene.

Generative Adversarial Networks (GAN) have revolutionized the field and are capable of producing semantically sound samples by creating a competition between a generator (G), which endeavours to capture the data distribution, and a discriminator (Dis), which judges the generator output and penalizes unrealistic images. Both networks are trained simultaneously to achieve an equilibrium. More formally put, this competition follows the minimax objective [23]:

$$\min_{G}\max_{Dis} \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[log(Dis(x))] + \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[log(1-Dis(\tilde{x}))], \tag{1}$$

where $\mathbb{P}_r$ is the data distribution, $\mathbb{P}_g$ is the model distribution defined by $\tilde{x} = G(z), z \sim p(z)$, with z being the random noise vector used as the generator input.

Training a GAN is rife with instability and potential issues [40], one of which is that the discriminator can rapidly reach optimality and easily distinguish between generator outputs and samples from the real distribution, and hence, will not produce meaningful gradients for training. In [38], the Earth Mover's distance (EM) or Wasserstein-1 metric is used to measure the distance between two distributions. The EM distance, $EM(p, q)$, is the minimum cost of moving distribution elements (earth mass) to transform a distribution q to distribution p (cost = mass × transport distance) and the Wasserstein GAN [38] has an aptly named "critic" (C) instead of the conventional discriminator since it is no longer a classifier. Using the EM distance, the critic will not solely judge whether a sample is fake or real as a discrete binary decision, but how real or how fake the generated sample is as a continuous regressive output. The critic will converge to a linear function with ever-present meaningful gradients and cannot saturate. The loss in the Wasserstein GAN is created via the Kantorovich-Rubinstein duality [38]:

$$\min_{G}\max_{C\in\mathcal{F}} \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[C(x)] - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[C(\tilde{x})], \tag{2}$$

where $\mathcal{F}$ is the set of 1-Lipschitz functions, $\mathbb{P}_r$ the true distribution, $\mathbb{P}_g$ the model distribution defined by $\tilde{x} = G(z), z \sim p(z)$, and z random noise. If C is optimal, minimizing the value function with respect to G minimizes $EM(\mathbb{P}_r, \mathbb{P}_g)$.

The Wasserstein GAN does not suffer from vanishing gradients and is immune to mode collapse. However, to guarantee continuity, a Lipschitz constraint must be enforced, which is achieved in [38] by clamping the weights. This creates a new clamping hyperparameter, which needs to be carefully tuned to the distribution. A gradient norm penalty with respect to the critic input is proposed in [39] to replace clamping. Since a differentiable function is 1-Lipschitz if and only if its gradient norm is no more than 1 everywhere, Gulrajani et al. [39] limits the critic gradient norm by penalizing the function on the gradient norm for samples $\hat{x} \sim \mathbb{P}_{\hat{x}}$, where $\hat{x} = \epsilon x + (1-\epsilon)\tilde{x},\ 0 < \epsilon < 1$. The new loss is therefore as follows [39]:

$$\min_{G}\max_{C} \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[C(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[C(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}}C(\hat{x})||_2 - 1)^2], \tag{3}$$

where $\mathbb{P}_g$ is the model distribution defined by $\tilde{x} = G(z), z \sim p(z)$, with z being the random noise vector, $\mathbb{P}_r$ is the true data distribution, and $\mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$ [39]. Here, we use the same critic for our adversarial loss.

In this work, our model is trained on a *synthetic* dataset of RGB-D images to perform depth filling. However, due to *dataset bias* [57], a model trained using data from a specific domain does not necessarily generalize to other data domains. In other words, a model trained on *synthetic* data may not perform well on *real-world* data. Therefore, our model may not succeed with naturally obtained depth images, which would make it utterly useless from a practical standpoint.

While the typical solution to this problem is to fine-tune the network on the novel data, 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 intractable to obtain. This is often the reason why synthetic data is used in the first place, as it is in our case. One strategy often to solve the problem is to minimize the distance between the source and target feature distributions [58,59]. Fig. 2 demonstrates how domain adaptation can aid in modelling the distribution of both the source domain (represented in blue), used for training the model, and the target domain (represented in red), which is the focus of the final objective. Using domain transfer, both distributions can be captured within the model even if the model is only trained on one of them.

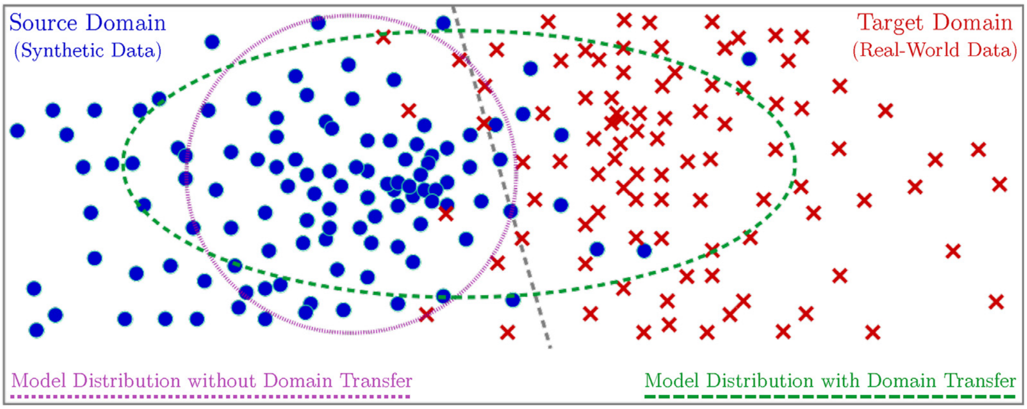
Source Domain
(Synthetic Data)
Target Domain
(Real-World Data)
Model Distribution without Domain Transfer
Model Distribution with Domain Transfer

Fig. 2. A demonstration of modelling two separate data domain distributions via domain transfer. A model only trained on samples from the source domain (blue) can capture the target data distribution (red) using domain adaptation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Some approaches have taken advantage of MMD (maximum mean discrepancy) which calculates the norm of the distance between the domains to reduce the discrepancy [60], whereas others have taken to using adversarial training which leads to a representation that minimizes the domain discrepancy while able to discriminate the source labels easily [59]. Although most of the above-mentioned techniques focus on discriminative models, research concentrating on generative tasks has also utilized domain adaptation [61].

We propose a domain critic network, which uses the Wasserstein metric to measure the distance between the source (synthetic data) and the target (real-world data) and minimizes this difference by comparing the generator outputs when synthetic and real-world images are used as the input, while the generator is simultaneously trained to fill synthetic holes using synthetic ground truth. Further details of the inner-workings of the proposed approach are explained in Section 4.

3. Data preparation

In a supervised learning approach, ground truth labels are required during training. Since the objective here is to fill depth holes, ground truth hole-free depth is required. However, obtaining complete depth from the real world is not practically possible. Consequently, we use synthetic data acquired from a graphically rendered gaming environment focusing on driving scenarios, akin to [35].

Necessary steps were taken to avoid dataset bias. Co-registered colour and depth images are captured from a camera view set in front of a virtual car as it automatically drives. An image is captured every 60 frames as the height and field of view of the camera are randomly changed after every capture. The process is carried out in numerous weather and lighting conditions at different times of day to avoid any possible model over-fitting. A total of 130,000 images were captured with 100,000 used for training and 30,000 set aside for testing.

During training, depth images are used as ground truth but corrupted depth images (with holes) of the same scenes are required as inputs. Rather than randomly cutting out sections of the image, we opt for creating realistic holes with the characteristics of those found in real-world depth images, which occur in stereo correspondence due to the existence of featureless or shiny surfaces, unclear object separation and distant objects, among others [6]. To produce these semantically meaningful holes, a separate model is needed to predict depth holes by means of pixel-wise classification, e.g., [51,62]. The objective is to produce a hole mask, which represents regions in the depth image likely to contain holes. Since within our synthetic dataset, only complete pixel-perfect depth is available, simulating corrupted depth, similar to what is naturally sensed in the real world, is important. The details of our *"hole prediction"* stage is explained in the following.

Table 1
Statistical accuracy of hole prediction over an unseen test set of 5000 examples images.

Model	Class label		Overall performance	
	Hole	Non-hole	Class average	Global average
Hole prediction	90.31	92.88	91.55	91.83

3.1. Hole prediction

Our hole prediction model is a fully convolutional encoder-decoder network inspired by [45,51] with nine convolutional layers in both the encoder and the decoder. No fully-connected layers are used to maintain a smaller number of network parameters and therefore, easier and faster training and inference. Every decoder layer corresponds to an encoder layer, with the last decoder layer connecting to a soft-max classifier. Each convolutional layer is followed by batch normalization [63] and a ReLU. Max-pooling is used in the sub-sampling to produce features that are invariant to small translational shifts in the input. In the decoder, max-unpooling [51] (which uses the recorded locations of maxima within the region of each max-pooling operation) is applied to preserve the feature structure and boundary information. The network architecture is seen in Fig. 3.

A number of stereo images (40,000) from the KITTI dataset [64] was used to train the network by estimating the disparity via Semi-Global Matching (SGM) [43] and generating a hole mask (M) which indicates which pixels are holes i.e. regions for which disparity was not recovered (with a value of zero) and which are non-holes (with a value of one). Although SGM is used, this is interchangeable with any depth via disparity or active depth capture approach. The left RGB images are used as inputs with the generated masks as ground truths. Cross-entropy is used as the loss function with the network weights randomly initialized.

Our hole prediction process is self-supervised, meaning no human annotation or intervention is necessary at any point, with the ground truth calculated using a disparity estimation approach [43]. Although this makes the ground truth for hole prediction unreliable, consequently making any accurate quantitative analyses meaningless, it suits the purposes of this endeavour. However, when tested on a set of 5000 previously-unseen images, the statistical correlation to the ground truth occurrence of holes within the image is shown to be accurate (see Table 1).

Qualitative evaluations reveal that holes are predicted where expected. From Fig. 4, we see that in regions where camera overlap is absent or featureless surfaces, sparse shrubbery, unclear object boundaries, and very distant objects are present, such pixels are correctly classified as holes. This model is subsequently used to infer where the holes would be in the hole-free ground truth synthetic RGB-D images (discussed earlier) needed for training the hole filling model.

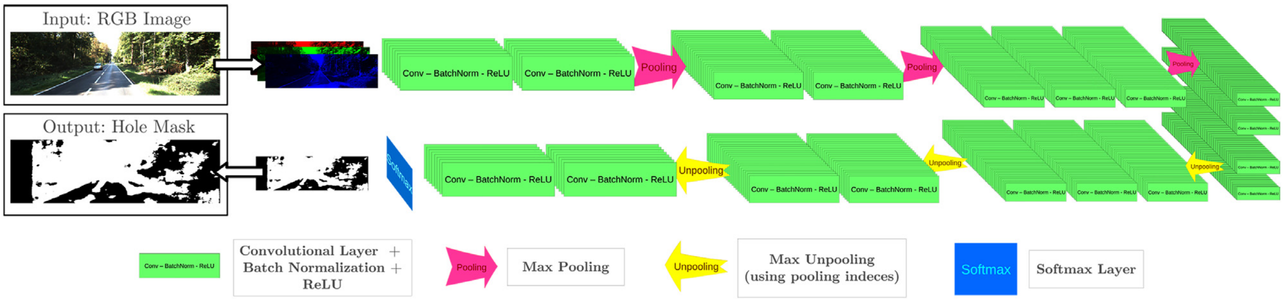

Fig. 3. An overview of the network architecture used during the hole prediction stage.

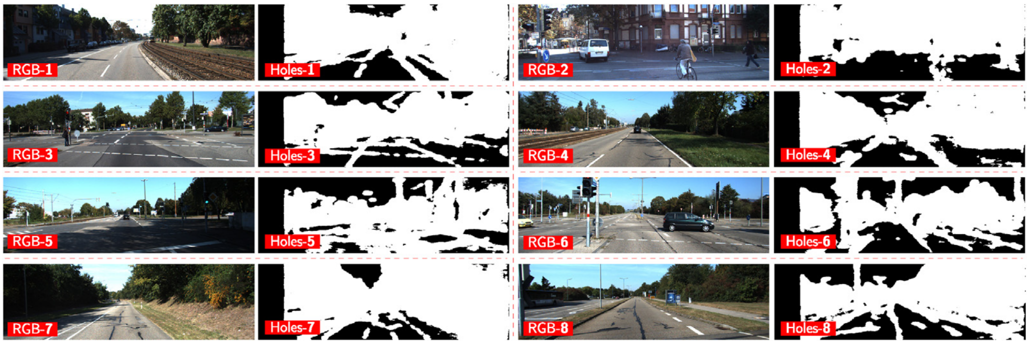

Fig. 4. Examples of the hole prediction model applied to a test set of 5000 images (*RGB*) from [64]. Note that featureless surfaces and sky are correctly identified as holes in the outputs (*Holes*).

4. Hole filling

Taking advantage of the adversarial training procedures present within the literature [38,39], our process involves three networks: a generator (*G*) which follows an encoder-decoder pipeline and is tasked with generating the completed depth, an image critic (*C*) which judges the generator output in an adversarial fashion and a domain transfer network (*D*) which provides the possibility of applying the model (trained on synthetic data) to natural images without ground truth depth. The interactions between all the networks are demonstrated in Fig. 5. In this section, details regarding the hole filling process are briefly outlined.

4.1. Missing depth prediction

Depth filling is performed by a generator with an encoder-decoder pipeline (the only network used during inference). A synthetic 4-channel RGB-D image containing holes (predicted by the model discussed in Section 3.1) is used as the encoder input, which creates a compact set of feature representations. This set of feature representations is then passed through the decoder, creating a single channel depth image with the missing regions filled if necessary (exceptions being very distant objects and sky, for which no valid depth should or does exist).

For the sake of consistency, the same architecture (Fig. 6) is used for both the hole prediction network (Section 3.1) and the generator. Since the goal is to test the learning capabilities of the model, the weights are randomly initialized and training procedure commences from scratch. The network is fully-convolutional with nine convolutional layers, batch normalization and max-pooling operations (Fig. 6). A large feature map of 78 × 24 × 256 is produced in the bottleneck. Many past works [22,24,25] advocate sub-sampling the image down to a small feature map passed through a fully-connected layer to allow for *"entire image context reasoning for each unit"* [22]. We experimented with fully-connected layers but other than a significant increase in the number of parameters, training difficulty and inference time, no noticeable difference in the quality of the results was observed. This means direct connections between different regions of a single feature map is not necessary for this task.

During inference, after the generator outputs the completed depth image, the regions filled by the network are blended using the approach in [65] into the hole regions within the original hole-ridden input image to create the final results.

4.2. Loss function

Our resulting model performs depth filling by regressing to the ground truth depth content of the unknown regions. Using

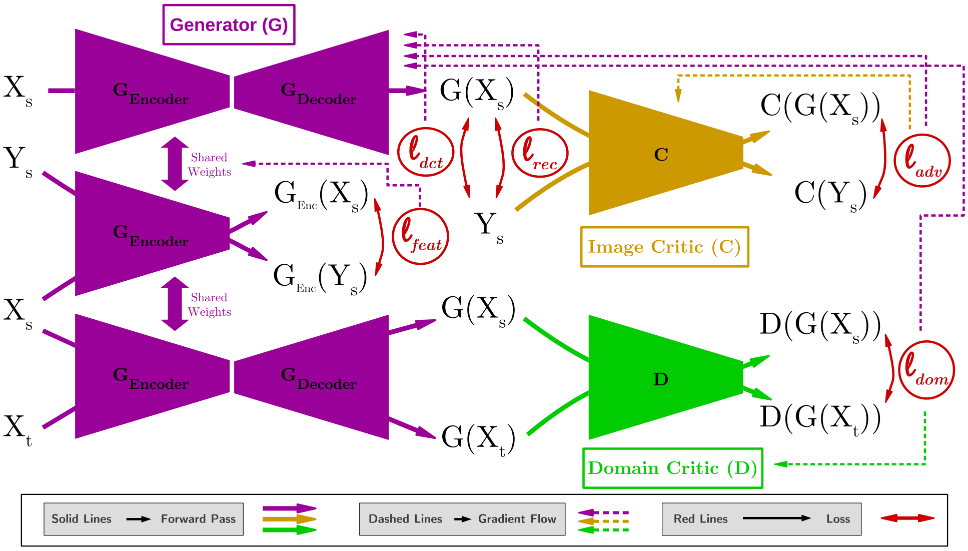

Fig. 5. The general framework of the entire model. The pipeline contains a generator (the only network used during inference, as explained in Section 4.1), an image critic to ensure the high fidelity of the generated depth (Section 4.2.2) and a domain critic to enforce generalization over real-world data (Section 4.2.4). Loss functions and gradient flows are shown for all components.

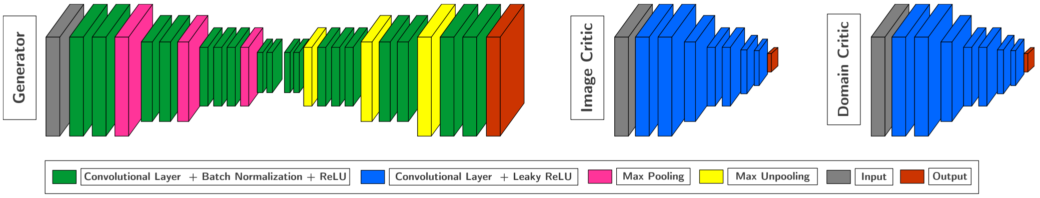
Generator
Image Critic
Domain Critic
🟩 Convolutional Layer + Batch Normalization + ReLU
🟦 Convolutional Layer + Leaky ReLU
Max Pooling
🟨 Max Unpooling
Input
🟥 Output

Fig. 6. Network architectures (generator, image critic and domain critic) used in the training.

a reconstruction loss, we ensure the filled regions are contextually sound, coherent and in accordance with the known regions. The addition of an adversarial loss [39] results in plausible outputs since the adversarial framework will enforce mode selection. To ensure robust contextual feature extraction and better encoder training, the distance is measured between the feature maps produced in the generator bottleneck when the depth channel of the input contains holes and when the ground depth is used as the input. The generator is then trained to minimize this distance. This guarantees correct and balanced training of the encoder and the decoder within the generator.

Additionally, even with perfect training, a network trained on synthetic data cannot be expected to perform equally well on naturally sensed depth images. A domain transfer loss is consequently used to ensure that our approach can complete naturally sensed depth images. A joint loss function is thus formulated consisting of four components:- reconstruction loss (Section 4.2.1), adversarial loss (Section 4.2.2), bottleneck feature loss (Section 4.2.3) and domain transfer loss (Section 4.2.4) - each of which are subsequently detailed.

4.2.1. Reconstruction loss

To maintain structural continuity and semantic coherence in the output, a reconstruction error against the ground truth is needed. However, to achieve sharper and more crisp results and to ease the burden on the adversarial image critic to enforce realism, we utilize a two-term reconstruction loss. Given a ground truth depth y, our generator (G) takes an input x, which itself is created based on y and generates $G(x)$. In this context, our hole prediction model (Section 3.1) has produced a binary hole mask, M, in which 0 denotes an unknown region (hole) and 1 a known depth region. The generator input, x, is obtained as follows:

$$x = y \odot M, \tag{4}$$

where $\odot$ is the element-wise product operation. We use a masked ℓ_1 distance as part of our reconstruction loss:

$$\mathcal{L}_{rec-\ell_1} = ||(1-M)\odot G(x) - (1-M)\odot y||_1 \tag{5}$$

Experiments with ℓ_2 loss returned the same results. With the known issues of a reconstruction loss, blurry images are often produced, which is why the use of adversarial losses is prevalent. However, here we add another term to our loss to partly alleviate the issue of blurring. Since the Discrete Cosine Transform (DCT) can be used to encode a unique embedding of spatial image structure, avoiding the limitations of ℓ_1 pixel space embedding (Fig. 1), the entire (unmasked) generated output $G(x)$ and the ground truth depth y both undergo the transform and the distance is measured within the projected DCT space:

$$\mathcal{L}_{rec-dct} = ||DCT(G(x)) - DCT(y)||_1 \tag{6}$$

The final reconstruction loss used in this work is therefore:

$$\mathcal{L}_{rec} = \mathcal{L}_{rec-\ell_1} + \mathcal{L}_{rec-dct} \tag{7}$$

Although the addition of the $\mathcal{L}_{rec-dct}$ reduces blurring, the overall quality of the output is still subject to issues due to the generality of the ℓ_1 distance, ensuring an adversarial component is subsequently needed.

4.2.2. Adversarial loss

Unlike most generative models, our network is conditioned on the known regions of the depth and the entire RGB and is tasked with generating the full depth. Our generator approximates a function which maps samples from the noisy distribution x to the true data distribution y, G: $x \mapsto y$. No noise or drop-out is used and the image critic is not conditioned like the generator, and sees the entire generator output, such that it cannot take advantage of structural discontinuities or possible differences in the overall intensities within the depth in its judgement. Therefore, it improves the whole generator output and not just the missing regions. The objective of the image critic is hence measuring the difference (using the *EM* metric) between real data samples and generated ones. Given that $\tilde{y} = G(x)$, the objective function of the critic is:

$$\min_G \max_C \mathop{\mathbb{E}}_{\tilde{y}\sim\mathbb{P}_g}[C(\tilde{y})] - \mathop{\mathbb{E}}_{y\sim\mathbb{P}_r}[C(y)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}}C(\hat{x})||_2 - 1)^2], \tag{8}$$

where $\mathbb{P}_g$ is the model distribution defined by $\tilde{y} = G(x)$, with x being the generator input sampled from the noisy distribution, $\mathbb{P}_r$ is the true data distribution, and $\mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$ [39]. The generator objective is to fool the image critic by creating increasingly more realistic outputs and getting closer to the true data distribution. The adversarial loss is thus as follows:

$$\mathcal{L}_{adv} = \max_C \; - \mathop{\mathbb{E}}_{\tilde{y}\sim\mathbb{P}_g}[C(\tilde{y})], \tag{9}$$

where once again, $\mathbb{P}_g$ is the model distribution defined by $\tilde{y} = G(x)$, with x being the generator input sampled from the noisy distribution. The generator and the image critic are trained iteratively while the critic is kept optimal at all times (in each epoch, it is trained 25 times per each generator training iteration for the first 100 generator iterations and 5 times per each generator iteration for the rest of the training process). The critic is a fully-convolutional network with no batch normalization. An overview of its architecture is seen in Fig. 6 (image critic).

4.2.3. Bottleneck feature loss

In a typical convolutional encoder-decoder pipeline [22,51], the convolutional layers in the encoder and the decoder learn independently. This can be advantageous as it provides a wide learning domain for the network. However, convergence to optimality can be slow and difficult.

Since the generator needs to predict any missing depth based on the RGB view and known depth regions, we can improve the generator training by making sure the encoder is creating the right feature representation of the entire scene, and the decoder is, in turn, starting from the best set of feature maps to produce the output.

Using the ground truth depth as the input and comparing the generated bottleneck features with the features produced from the regular input (depth with holes), we can guarantee the encoder is rightly trained to capture the full information available in the scene based on context and inferred geometry rather than local low-level scene features. As Fig. 7 demonstrates, the ground truth

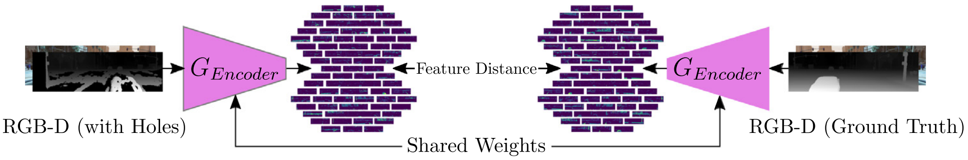

Fig. 7. A demonstration of how the bottleneck feature distance is calculated. Depth with holes (left) and ground truth depth (right) are used as inputs to the generator encoder. Minimizing the absolute difference between the feature maps extracted from the bottleneck is part of the objective.

depth is used as the input for the generator (right side of the figure) and the depth image with holes is also used as the input (left side of the figure). The distance between such features extracted from the generator bottleneck is then used as a component of the loss. Subsequently, our loss includes the distance between the generated bottleneck features from the ground truth and the noisy input:

$$\mathcal{L}_{feat} = ||G_{encoder}(x) - G_{encoder}(y)||_1 \tag{10}$$

In all previous loss terms, x as the input to the generator was a 4-channel RGB-D image with the depth channel containing holes and y a single-channel hole-free depth image. In Eq. (10), however, y is also a 4-channel RGB-D image, but the depth channel is the ground truth depth (hole-free). For the sake of consistency, the same notation is used in Eq. (10).

4.2.4. Domain transfer loss

All the training data used here are synthetic images, yet for the model to be practically viable, it has to perform on real-world images. Since no naturally-obtained ground truth is available for training, the generator is also trained to recognize natural data in an adversarial fashion (similar to Section 4.2.2).

Let all synthetic inputs (source domain) be denoted by x_s with synthetic ground truth y_s. All naturally-obtained data (target domain) are denoted by x_t. Note that there is *no* y_t since our target domain (naturally-sensed images) has no ground truth (hole-free) depth. A domain critic network (D) is used to measure the difference (in *EM* distance) between the generator output when the input is sampled from the source domain (x_s) and when the input is from the target domain x_t. The gradients will be used to train the generator and the generator is subsequently forced to model the distribution of both the source and the target domains. Given that $\tilde{y_s} = G(x_s)$ and $\tilde{y_t} = G(x_t)$, the objective function of the domain critic is:

$$\min_G \max_D \underset{\tilde{y_t}\sim\mathbb{P}_t}{\mathbb{E}}[D(\tilde{y_t})] - \underset{y_s\sim\mathbb{P}_s}{\mathbb{E}}[D(\tilde{y_s})] + \lambda \underset{\hat{x}\sim\mathbb{P}_{\hat{x}}}{\mathbb{E}}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2], \tag{11}$$

where $\mathbb{P}_t$ is the target model distribution defined by $\tilde{y_t} = G(x_t)$, with x_t being the generator input sampled from the natural data distribution, $\mathbb{P}_s$ is the source data distribution defined by $\tilde{y_s} = G(x_s)$, with x_s being the generator input sampled from the synthetic data distribution, and $\mathbb{P}_{\hat{x}}$ is implicitly defined to sample uniformly along straight lines between pairs of points from $\mathbb{P}_t$ and $\mathbb{P}_s$ [39].

The generator objective is to fool the domain critic by approximating both domain distributions. The domain transfer loss is as follows:

$$\mathcal{L}_{domTran} = \max_D - \underset{\tilde{y_t}\sim\mathbb{P}_t}{\mathbb{E}}[D(\tilde{y_t})], \tag{12}$$

where $\mathbb{P}_t$ is the natural domain distribution defined by $\tilde{y_t} = G(x_t)$, with x_t being the generator input from natural data. The generator and the domain critic are trained iteratively while the domain critic is always kept optimal, much like the critic in Section 4.2.2. The domain critic architecture is the same as the image critic, as seen in Fig. 6. Weight sharing between the image critic and the domain critic was attempted, but we could not achieve convergence with that setup.

Synthetic ground truth y_s does not come into play in domain transfer training and the model is trained on the source domain to approximate the data distribution (from which y_s is sampled). The domain transfer loss thus forces the generator to comprehend both the natural and synthetic distributions. Additionally, over-training the model using domain transfer leads to artefacts in the outputs. Thus, this term is only used in a quarter of the total number of epochs (see Section 5.1). It is important to note that this loss component was originally used in the training objective of the hole prediction network (Section 3.1) as well, but with no evidence for any significant improvement in the output.

4.2.5. Joint loss

Based on Eqs. (7) (Section 4.2.1), (9) (Section 4.2.2), (10) (Section 4.2.3) and (12) (Section 4.2.4), our overall joint loss function is finally defined as:

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{domTran}\mathcal{L}_{domTran} \tag{13}$$

The choice of the weights λ_{rec}, λ_{adv}, λ_{feat} and $\lambda_{domTran}$ is empirical.

5. Experiments

A total of 30,000 synthetic images were used as part of the test set. Moreover, a set of 5000 locally-captured images consisting of RGB and registered depth containing holes were used for training as part of domain critic training and subsequently used to test the model on real-world natural images.

5.1. Implementation details

All network implementation and training is done in *PyTorch* [66] and *Caffe* [67]. The Adam optimization method [68] is used for this problem (momentum $\beta_1 = 0.5$, $\beta_2 = 0.999$, and initial learning rate $\alpha = 0.0001$), and the coefficients in the loss function are empirically chosen to be $\lambda_{rec} = 100, \lambda_{adv} = 0.01, \lambda_{feat} = 0.01, \lambda_{domTran} = 0.01$ based on a preliminary grid search with coefficients changing an order of magnitude between 0.01 and 100. The networks used in the hole filling model are trained for 20 epochs over the entire dataset with a batch-size of 7 images. The domain transfer loss, $\mathcal{L}_{domTran}$, is used only every other epoch and only in the last 10 epochs to avoid introducing undesirable effects in the outputs.

5.2. Ablation study

A crucial part of this work was interpreting the necessity of the components of our loss function. The model was trained from random initialization each time after adding a single component of the loss function. As seen in Fig. 8, when a simple reconstruction loss (ℓ_1) is solely used, large holes are ubiquitously filled with averaged blurry content (blue boxes in Fig. 8). The addition of the

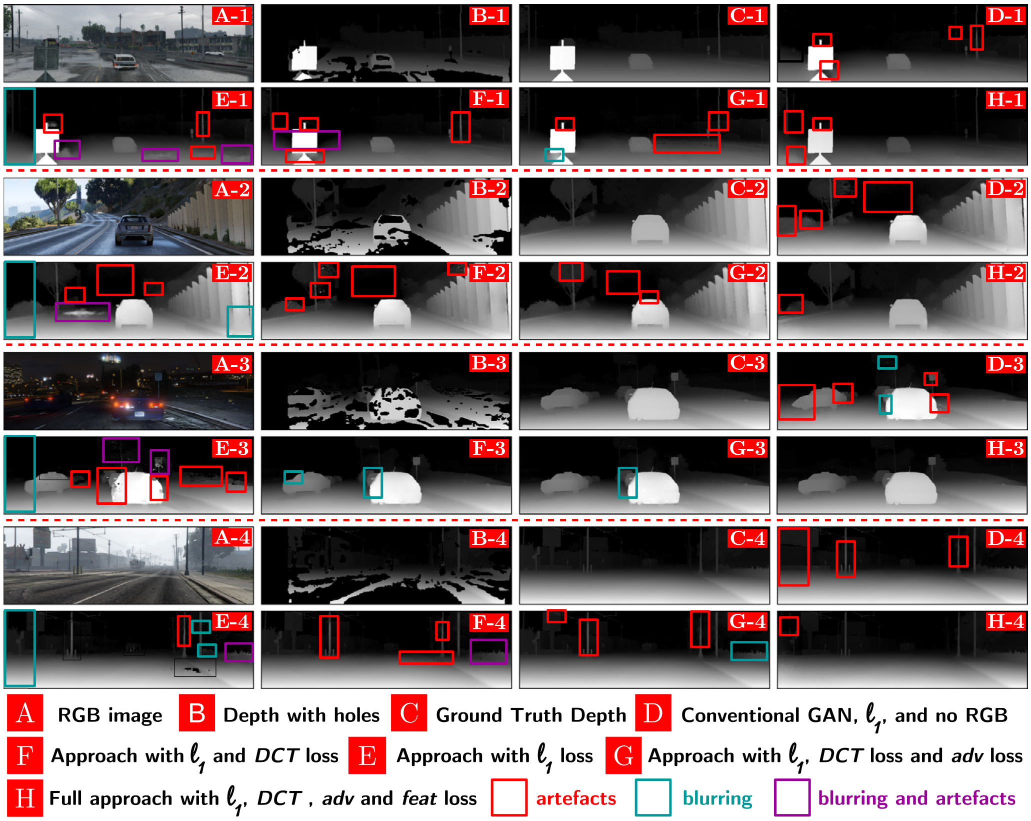

Fig. 8. Hole filling results when different components of the loss function are added to the joint loss function. Note that the results of the full proposed approach (with all four components) are substantially superior.

Table 2
Numerical comparison of our approach, our ablated method and other hole filling methods, such as Fourier-based inpainting [7] (FBI), smoothing second order inpainting [14] (SSI), exemplar-based inpainting [3], fast marching method [4] (FFM), guided inpainting and filtering technique [10] (GIF). While disparity error values are lower for more realistic images, with Peak Signal-to-Noise Ratio (PNSR) and Structural Similarity index (SSIM), higher values are better.

Method	Mean ℓ_1 error	Mean ℓ_2 error	PSNR (dB)	SSIM $(-1, 1)$
SSI [14]	5.66 ± 1.033	2.96 ± 0.512	16.04 ± 4.819	0.772 ± 0.220
FMM [4]	2.85 ± 0.491	0.89 ± 0.198	20.66 ± 2.030	0.780 ± 0.082
EBI [3]	2.39 ± 0.629	0.92 ± 0.091	20.65 ± 3.122	0.787 ± 0.139
GIF [10]	2.77 ± 0.518	0.86 ± 0.068	20.78 ± 1.910	0.764 ± 0.125
FBI [7]	2.36 ± 0.602	0.91 ± 0.105	20.67 ± 2.891	0.788 ± 0.106
ℓ_1 Loss Only	2.96 ± 0.489	0.28 ± 0.038	25.99 ± 2.890	0.819 ± 0.112
ℓ_1 + *dct* Loss	2.47 ± 0.422	0.19 ± 0.047	27.98 ± 2.019	0.872 ± 0.132
ℓ_1 + *dct* + *adv* Loss	2.33 ± 0.405	0.17 ± 0.050	28.50 ± 1.105	0.882 ± 0.096
CE [22] (ℓ_2 + *adv* Loss)	2.18 ± 0.391	0.18 ± 0.034	28.21 ± 1.359	0.877 ± 0.108
Full proposed approach	$\mathbf{1.79} \pm 0.401$	$\mathbf{0.08} \pm 0.011$	$\mathbf{31.89} \pm 2.012$	$\mathbf{0.928} \pm 0.110$

DCT helps in alleviating the issue, but blurring and unwanted artefacts still exist. The adversarial loss clears the image to a great extent but the use of full joint loss function (except of course the domain transfer portion, which is only relevant to natural images) creates a sharp and realistic image, with minimal differences with the ground truth. The significant similarities seen between the final results and the ground truth is in part because the model is conditioned on the RGB view, as seen in Fig. 8 (*D*) where the RGB is not used in the training.

Not only are the images realistic to the human eye, quantitative results in Table 2 demonstrate that our results are clearly superior to the prior works of [3,4,7,10,14]. As seen in Table 2, we use four metrics to compare the results against the ground truth (mean absolute difference, mean squared difference, peak signal-to-noise ratio and the structural similarity index). Overall, Table 2 shows a significant reduction in prediction errors of the proposed approach against ground truth with negligible standard deviation indicating consistent performance over the randomly selected test set of 30,000 synthetic images.

5.3. Evaluation using non-synthetic data

The last component of our loss fits the model to naturally-sensed images as well as the synthetic data. The effectiveness of this loss term is demonstrated in Fig. 9. Without data domain transfer, the network is incapable of producing valid and meaningful results. The domain transfer loss is only used in a quarter (1 in 4) of all training epochs to avoid over-fitting. The adversarial nature of our domain adaptation can result in the generator attempting to produce pixel-perfect depth images when a real-world

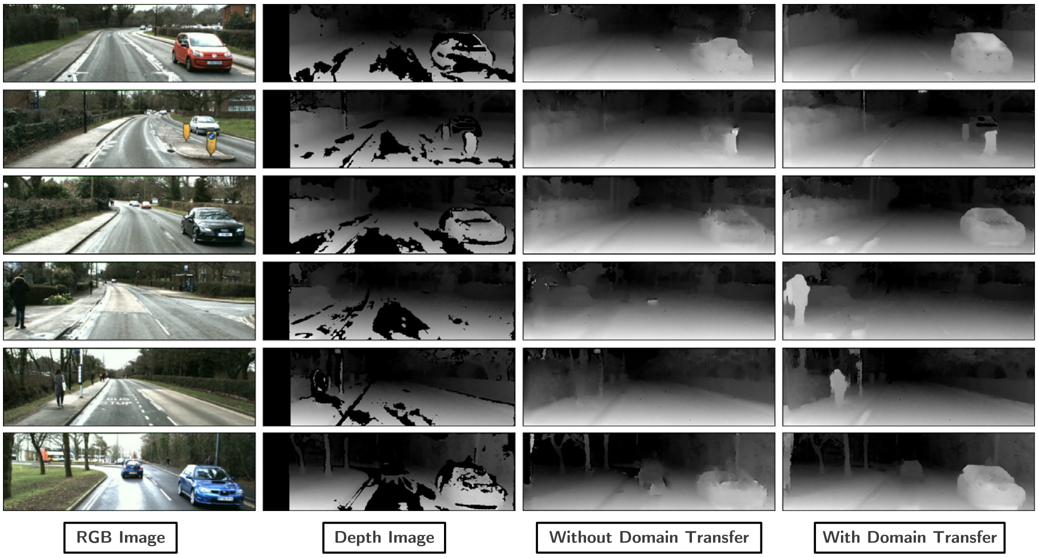

Fig. 9. Comparing the results of our approach on natural real-world data with and without domain transfer. It is clear that with domain transfer, the hole filling model (trained on synthetic data) applied to non-synthetic images outperforms the model not using any form of domain adaptation.

Table 3
Comparing the run-time of our approach with classical hole-filling techniques. Note that only requiring a single forward pass, our approach is highly efficient using modern hardware.

Method	FBI [7]	EBI [3]	SSI [14]	GIF [10]	FMM [4]	Our approach
Run-time (ms)	>36*e*5	>12*e*5	33.4*e*3	14.32*e*2	82.8*e*1	7.47

image is given as its input and therefore removes entire objects or synthesizes ones that should not be in the scene.

However, with the correct training, real-world depth images that are of far lower quality than the synthetic ones can be filled in a realistic and consistent manner. As Fig. 9 demonstrates, naturally-sensed depth images are filled in a more plausible manner with domain transfer as part of the training (Fig. 9 - fourth column) than with no domain adaptation (Fig. 9 - third column).

5.4. Comparison to contemporary approaches

The approach is also evaluated against classical hole filling techniques. We used both synthetic and natural images to test the performance, and since ground truth depth is available for the synthetic data, numerical analysis is possible in the evaluation. A Fourier-based inpainting approach [7] (FBI), a smoothing second order inpainting [14] (SSI), an exemplar-based inpainting [3], a fast marching method [4] (FFM) and a guided inpainting and filtering technique [10] (GIF) are chosen for their accuracy and their capability of handling relatively large holes.

As indicated in Table 2, our approach outperforms the comparators by a large margin, even if the loss is stripped down to a simple reconstruction loss. Since the synthetic images are of extremely high quality (pixel-perfect dense depth information with granular texture and accurate object boundaries), they should be prime candidates for traditional hole filling methods. However, since learning the semantics, structures and the context of a scene plays a vital role in predicting its contents, our approach produces more realistic results with almost no anomalies, blurring or any undesirable artefacts, as seen in Fig. 10. Based on Fig. 11, similar conclusions can be drawn when it comes to natural real-world images, where the depth is of significantly lower quality compared to synthetic data. The capabilities of our approach over real-world data are owed to the domain transfer component of our loss function (Section 5.3).

Regarding efficiency, the runtime of our approach heavily depends on the hardware. All training and inference were done using an NVIDIA GeForce GTX 1080 Ti GPU and our mean inference time (requiring a single forward pass) is 7.47 ms based on processing a 192 × 640 image (4 channel, RGB-D). Table 3 provides a comparative analysis of our approach and the comparators.

5.5. Feature learning

Our model is shown to be capable of learning scene context and content in its attempt to produce a complete hole-free depth image. Since our technique does not utilize off-the-shelf classic network architectures, quantifying the feature strength within the network weights used as a pre-training stage for tasks such as classification and detection would not be possible. However, we could evaluate our features in a task somewhat similar to depth filling, namely monocular depth estimation, despite the differences between the two problems, e.g. the different low and high level cues that need to be learned by the network. We re-purpose our model to estimate scene depth based on a single RGB view by initializing the network with the pre-trained weights from the depth completion model (excluding the depth channel of the first convolutional layer). Fine-tuning is only performed over a single epoch of the dataset without any layer freezing. The results are compared with state-of-the-art approaches [27–29]. Qualitative results based on synthetic images used as inputs are seen in Fig. 12.

No domain adaptation to our real-world set was performed during this experiment but the models are evaluated using our real-world test images, nonetheless. As seen in Fig. 13, even though our network has never seen a real-world image and data domain has not been transferred, we can see that our approach produces

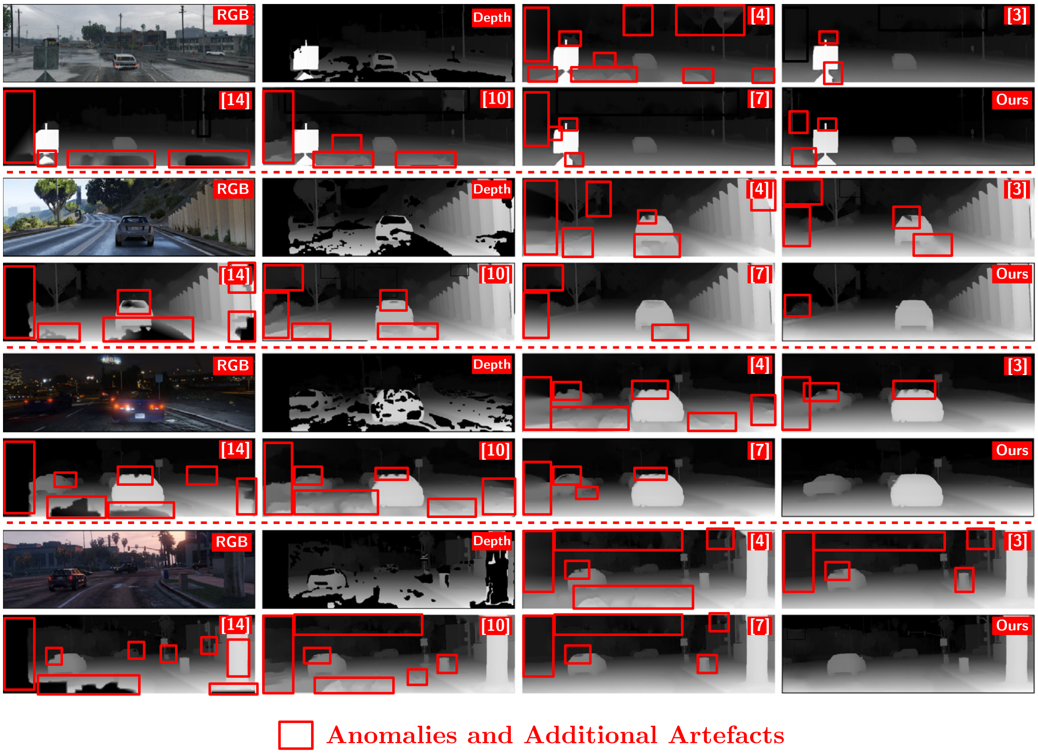

Fig. 10. Comparing our approach with hole filling methods in [3,4,7,10,14] (synthetic data).

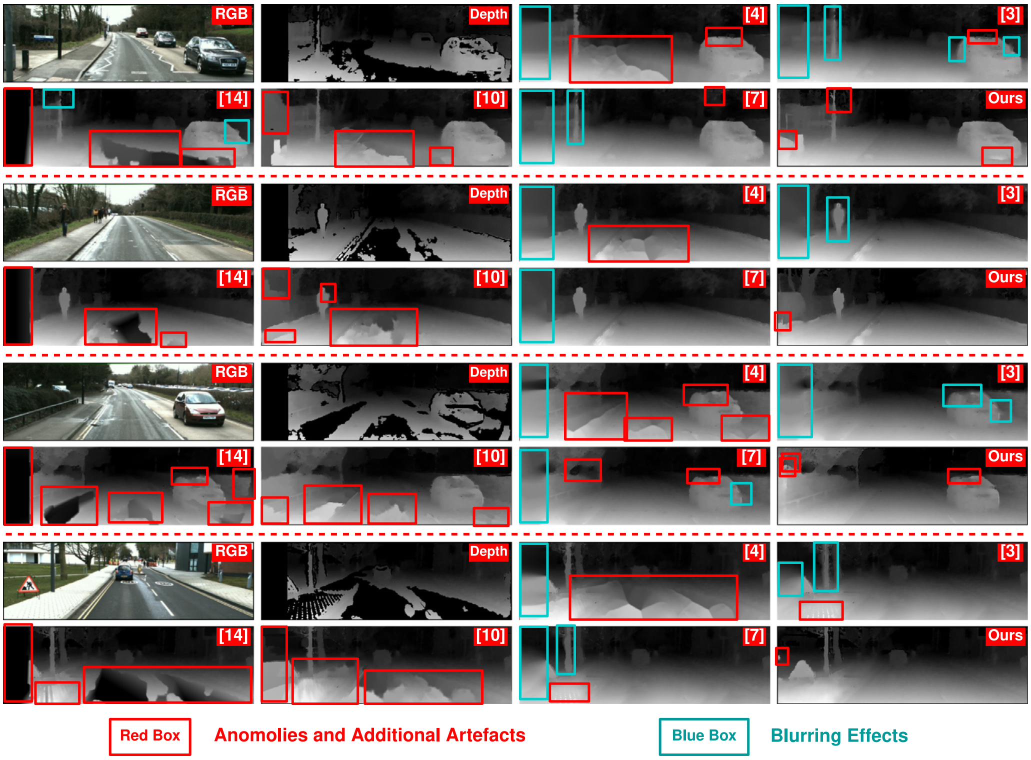

Fig. 11. Comparing the results of our approach against other hole filling methods [3,4,7,10,14] with natural real-world data.

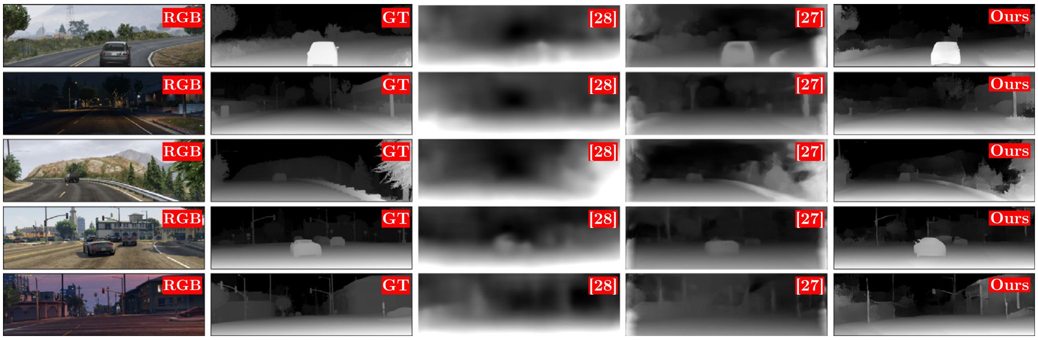

Fig. 12. The results of our approach re-purposed to estimate depth from an RGB image compared against [27] and [28] with synthetic data.

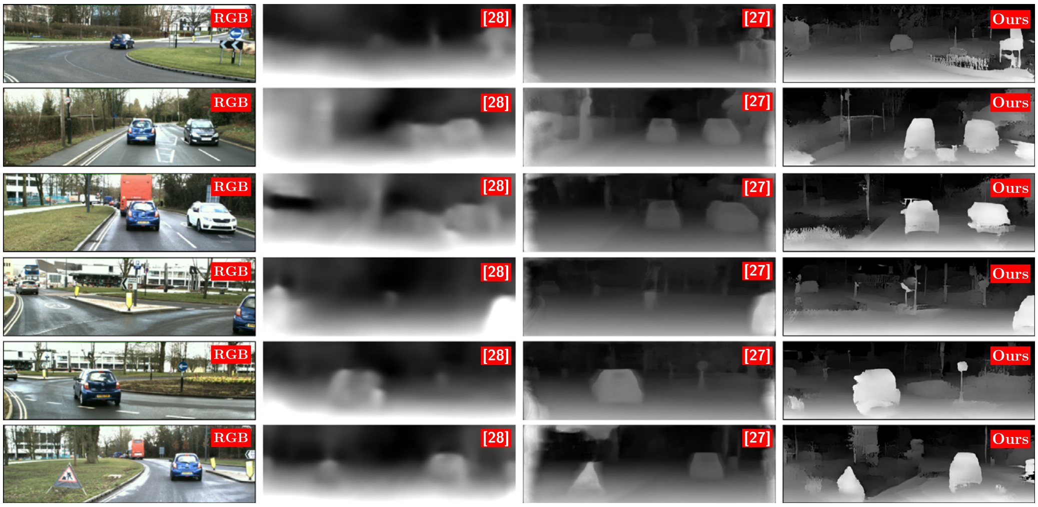

Fig. 13. The results of our approach re-purposed to estimate depth from an RGB image compared against [27] and [28] with natural real-world data.

Table 4
Our pre-trained model tasked with monocular depth estimation compared to [27–29]. More realistic images have lower error values, but with PNSR and SSIM, higher values are better.

Method	Mean ℓ_1 error	Mean ℓ_2 error	PSNR (dB)	SSIM $(-1, 1)$
Result of [28]	28.61	13.68	10.15	0.374
Result of [27]	14.46	3.93	14.22	0.565
Result of [29]	4.22	0.79	24.18	0.793
Our result	4.97	0.88	22.35	0.778

sharper and more crisp depth information despite the anomalies that persist due to domain bias issues. Quantitative analysis using synthetic ground truth images is presented in Table 4. While our approach cannot outperform directly-supervised models trained on similar synthetic data [29], we can see it has succeeded in a task it is not primarily designed for due to its strength in scene feature learning.

6. Conclusion

We have approached the problem of hole filling in depth images from a learning perspective by employ an adversarially trained self-supervised encoder/decoder architecture. It is expected that if enough is learned about the contents and semantics of a scene, missing regions of a depth image can be inferred given the known regions and the full RGB view. Our training is fully self-supervised, i.e. at no point is annotation or human intervention necessary. The ground truth depth used for training is acquired from a graphical environment developed for gaming and a separate model is trained to infer where holes would be if the data were obtained via stereo correspondence. The model objective is to minimize a loss consisting of four loss components: reconstruction, adversarial, bottleneck feature and domain transfer loss, which results in filling depth holes, not only in synthetic depth images but also in real-world data with no ground truth.

Even though the approach utilizes synthetic images for training and requires a complicated mixture of parameters with their own weighting coefficients, qualitative and quantitative evaluations demonstrate how it can outperform competing contemporary depth filling techniques. Moreover, the robust feature learning capabilities of our approach are clearly seen when it is used to estimate depth based on a single RGB image, a task it is not primarily designed or trained to perform.

Currently, the proposed approach only uses the local spatial information within the known regions of the depth and the complete RGB image to infer the missing regions of the depth. However, as part of possible future work, the use of temporal information available within a video sequence can greatly improve the quality of depth completion results since features extracted from one frame can be used to infer valuable information about the next. Additionally, using sparse or otherwise irregular convolutions, naturally-sensed depth images can be used in the training process making the model even more adaptable to real-world applications.

References

[1] T. Breckon, R. Fisher, 3D surface completion via non-parametric techniques, IEEE Trans. Pattern Anal. Mach. Intell. 30 (12) (2008) 2249–2255.

[2] T. Breckon, R. Fisher, A hierarchical extension to 3D non-parametric surface relief completion, Pattern Recognit. 45 (2012) 172–185.

[3] P. Arias, G. Facciolo, V. Caselles, G. Sapiro, A variational framework for exemplar-based image inpainting, Int. J. Comput. Vision 93 (3) (2011) 319–347.

[4] A. Telea, An image inpainting technique based on the fast marching method, J. Graph. Tools 9 (1) (2004) 23–34.

[5] D. Ding, S. Ram, J. Rodriguez, Perceptually aware image inpainting, Pattern Recognit. 83 (2018) 174–184.

[6] A. Atapour-Abarghouei, T. Breckon, A comparative review of plausible hole filling strategies in the context of scene depth image completion, Comput. Graph. 72 (2018) 39–58.

[7] A. Atapour-Abarghouei, G. Payen de La Garanderie, T. Breckon, Back to butterworth - a fourier basis for 3D surface relief hole filling within RGB-D imagery, in: Int. Conf. Pattern Recognition, IEEE, 2016, pp. 2813–2818.

[8] S.-H. Baek, I. Choi, M.H. Kim, Multiview image completion with space structure propagation, in: IEEE Conf. Computer Vision and Pattern Recognition, IEEE, 2016, pp. 488–496.

[9] A. Atapour-Abarghouei, T. Breckon, Extended patch prioritization for depth filling within constrained exemplar-based RGB-D image completion, in: Int. Conf. Image Analysis and Recognition, Springer, 2018, pp. 306–314.

[10] J. Liu, X. Gong, J. Liu, Guided inpainting and filtering for kinect depth maps, in: Int. Conf. Pattern Recognition, IEEE, 2012, pp. 2055–2058.

[11] S. Matyunin, D. Vatolin, Y. Berdnikov, M. Smirnov, Temporal filtering for depth generated by kinect camera, in: 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, IEEE, 2011, pp. 1–4.

[12] F. Qi, J. Han, P. Wang, G. Shi, F. Li, Structure guided fusion for depth map inpainting, Pattern Recognit. Lett. 34 (1) (2013) 70–76.

[13] A. Atapour-Abarghouei, T. Breckon, Depthcomp: real-time depth image completion based on prior semantic scene segmentation, in: British Machine Vision Conference, 2017, pp. 1–13.

[14] D. Herrera, J. Kannala, J. Heikkilä, et al., Depth map inpainting under a second-order smoothness prior, in: Scandinavian Conf. Image Analysis, Springer, 2013, pp. 555–566.

[15] F. Cheng, X. He, H. Zhang, Learning to refine depth for robust stereo estimation, Pattern Recognit. 74 (2018) 122–133.

[16] L.A. Gatys, A.S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 2414–2423.

[17] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: Euro. Conf. Computer Vision, Springer, 2016, pp. 694–711.

[18] L. Wang, Z. Huang, Y. Gong, C. Pan, Ensemble based deep networks for image super-resolution, Pattern Recognit. 68 (2017) 191–198.

[19] N. Kumar, R. Verma, A. Sethi, Convolutional neural networks for wavelet domain super-resolution, Pattern Recognit. Lett. 90 (2017) 65–71.

[20] K. Nguyen, C. Fookes, S. Sridharan, M. Tistarelli, M. Nixon, Super-resolution for biometrics: a comprehensive survey, Pattern Recognit. 78 (2018) 23–42.

[21] R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: Euro. Conf. Computer Vision, Springer, 2016, pp. 649–666.

[22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: feature learning by inpainting, in: IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[24] R.A. Yeh*, C. Chen*, T.Y. Lim, S.A. G, M. Hasegawa-Johnson, M.N. Do, Semantic image inpainting with deep generative models, in: IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 6882–6890.

[25] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, H. Li, High-resolution image inpainting using multi-scale neural patch synthesis, in: IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 4076–4084.

[26] C. Li, M. Wand, Combining markov random fields and convolutional neural networks for image synthesis, in: IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 2479–2486.

[27] C. Godard, O. Mac Aodha, G. J., Unsupervised monocular depth estimation with left-right consistency, in: IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 6602–6611.

[28] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth ego–motion from video, in: IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 6612–6619.

[29] A. Atapour-Abarghouei, T. Breckon, Real-time monocular depth estimation using synthetic data with domain adaptation via style transfer, in: IEEE Conf. Computer Vision and Pattern Recognition, 2018, pp. 1–8.

[30] Z. Zhang, C. Xu, J. Yang, Y. Tai, L. Chen, Deep hierarchical guidance and regularization learning for end-to-end depth estimation, Pattern Recognit. 83 (2018) 430–442.

[31] B. Li, Y. Dai, M. He, Monocular depth estimation with hierarchical fusion of dilated CNNs and soft-weighted-sum inference, Pattern Recognit. 83 (2018) 328–339.

[32] J. Lei, L. Li, H. Yue, F. Wu, N. Ling, C. Hou, Depth super-resolution considering view synthesis quality, IEEE. Trans. Image Process. 26 (2017) 1732–1745.

[33] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb, Learning from simulated and unsupervised images through adversarial training, in: IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 2242–2251.

[34] A. Gaidon, Q. Wang, Y. Cabon, E. Vig, Virtual worlds as proxy for multi-object tracking analysis, in: IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 4340–4349.

[35] R. Miralles, An open-source development environment for self-driving vehicles, in: Universitat Oberta de Catalunya, 2017, pp. 1–31.

[36] A. Dosovitskiy, T. Brox, Generating images with perceptual similarity metrics based on deep networks, in: Advances in Neural Information Processing Systems, 2016, pp. 658–666.

[37] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Int. Conf. Computer Vision, Springer, 2017, pp. 2242–2251.

[38] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Int. Conf. Machine Learning, 2017, pp. 214–223.

[39] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein GANs, in: Advances in Neural Information Processing Systems, 2017, pp. 5769–5779.

[40] M. Arjovsky, L. Bottou, Towards principles for training generative adversarial networks, in: Int. Conf. Learning Representations, 2017, pp. 1–17.

[41] R.H. Chan, T.F. Chan, C.-K. Wong, Cosine transform based preconditioners for total variation deblurring, IEEE Trans. Image Process. 8 (10) (1999) 1472–1478.

[42] A. Raid, W. Khedr, M. El-Dosuky, W. Ahmed, JPEG image compression using discrete cosine transform - a survey, Comput. Sci. Eng. Surv. 5 (2) (2014) 1–9.

[43] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 328–341.

[44] J.Y. Chang, K.M. Lee, S.U. Lee, Stereo matching using iterative reliable disparity map expansion in the color-spatial-disparity space, Pattern Recognit. 40 (12) (2007) 3705–3713.

[45] K. Simonyan, A. Zisserman, Deep convolutional networks for large-scale image recognition, in: Int. Conf. Learning Representations, 2015, pp. 1–12.

[46] C. Barat, C. Ducottet, String representations and distances in deep convolutional neural networks for classification, Pattern Recognit. 54 (2016) 104–115.

[47] X. Li, M. Fang, J.-J. Zhang, J. Wu, Learning coupled classifiers with RGB images for RGB-D object recognition, Pattern Recognit. 61 (2017) 433–446.

[48] X.-s. Tang, K. Hao, H. Wei, Y. Ding, Using line segments to train multi-stream stacked autoencoders for image classification, Pattern Recognit. Lett. 94 (2017) 55–61.

[49] H. Lei, T. Han, F. Zhou, Z. Yu, J. Qin, A. Elazab, B. Lei, A deeply supervised residual network for hep-2 cell classification via cross-modal transfer learning, Pattern Recognit. 79 (2018) 290–302.

[50] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, Q. Sun, Deep learning for image-based cancer detection and diagnosis a survey, Pattern Recognit. 83 (2018) 134–149.

[51] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.

[52] Y. Wei, X. Liang, Y. Chen, Z. Jie, Y. Xiao, S. Yan, Learning to segment with image-level annotations, Pattern Recognit. 59 (2016) 234–244.

[53] M. Hamandi, D. Asmar, E. Shammas, Ground segmentation and free space estimation in off-road terrain, Pattern Recognit. Lett. 108 (2018) 1–7.

[54] N. Wang, W. Zha, J. Li, X. Gao, Back projection: an effective postprocessing method for GAN-based face sketch synthesis, Pattern Recognit. Lett. 107 (2018) 59–65.

[55] C. Li, X. Zhao, Z. Zhang, S. Du, Generative adversarial dehaze mapping nets, Pattern Recognit. Lett. (2017) In Press.

[56] T. Yu, L. Wang, H. Gu, S. Xiang, C. Pan, Deep generative video prediction, Pattern Recognit. Lett. 110 (2018) 58–65.

[57] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, N. Lawrence, Covariate shift by kernel mean matching, Dataset Shift Mach. Learn. (2009) 131–160.

[58] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (59) (2016) 1–35.

[59] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 2962–2971.

[60] B. Sun, K. Saenko, Deep coral: correlation alignment for deep domain adaptation, in: Int. Conf. Computer Vision – Workshops, Springer, 2016, pp. 443–450.

[61] M.-Y. Liu, O. Tuzel, Coupled generative adversarial networks, in: Advances in Neural Information Processing Systems, 2016, pp. 469–477.

[62] Y. Guo, T. Chen, Semantic segmentation of RGBD images based on deep depth regression, Pattern Recognit. Lett. 109 (2018) 55–64.

[63] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Int. Conf. Machine Learning, 2015, pp. 448–456.

[64] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the KITTI dataset, Int. J. Rob. Res. (2013) 1231–1237.

[65] P. Pérez, M. Gangnet, A. Blake, Poisson image editing, in: ACM Trans. Graphics, 22, ACM, 2003, pp. 313–318.

[66] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in: Advances in Neural Information Processing Systems, 2017, pp. 1–4.

[67] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Int. Conf. Multimedia, ACM, 2014, pp. 675–678.

[68] D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Int. Conf. Learning Representations, 2014, pp. 1–15.

Amir Atapour-Abarghouei is currently a Ph.D. student in the Department of Computer Science in Durham University (UK). He received his B.Sc. degree from the Department of Computer Engineering at Shahid Bahonar University of Kerman (Iran) in 2008 and his M.Sc. degree from the Department of Computer Science at Universiti Teknologi Malaysia (Malaysia) in 2010. His research interests include 3D scene analysis, scene depth completion, monocular depth estimation, machine learning, deep neural networks and generative models.

Samet Akcay is a third year Ph.D. Student in the Department of Computer Science at Durham University (UK). He received his B.Sc. degree from the Department of Electrical and Electronics Engineering at Gazi University (Turkey) in 2011 and MSc. degree from the Department of Electrical Engineering at Penn State University (US) in 2015. His primary research interests are real-time image classification/detection, anomaly detection and unsupervised feature learning via deep/machine learning algorithms.

Grégoire Payen de La Garanderie is currently a Ph.D. student in the Department of Computer Science at Durham University (UK) sponsored by Jaguar Land Rover. Prior to this, he worked as a Graduate Research Engineer at Imagination Technologies. He received an M.Sc. from Cranfield University (UK) in 2013. His research interests include deep learning, object detection, visual question answering and computer vision applications to automotive.

Toby Breckon is currently a Professor in the Departments of Engineering and Computer Science, Durham University (UK). His key research interests lie in the domain of computer vision and image processing and he leads a range of research activity in this area. Dr. Breckon holds a Ph.D. in informatics (computer vision) from the University of Edinburgh (UK). He has been a visiting member of faculty at the Ecole Superieure des Technologies Industrielles Avancees (France), Northwestern Polytechnical University (China), Shanghai Jiao Tong University (China) and Waseda University (Japan). Dr. Breckon is a Chartered Engineer, Chartered Scientist and a Fellow of the British Computer Society. In addition, he is an Accredited Senior Imaging Scientist and Fellow of the Royal Photographic Society. He led the development of image-based automatic threat detection for the 2008 UK MoD Grand Challenge winners [R.J. Mitchell Trophy, (2008), IET Innovation Award (2009)]. His work is recognised as recipient of the Royal Photographic Society Selwyn Award for early-career contribution to imaging science (2011).