Cast Shadow Generation Using Generative Adversarial Networks

Khasrouf Taif, Hassan Ugail(✉), and Irfan Mehmood

Centre for Visual Computing, University of Bradford, Bradford, UK
h.ugail@bradford.ac.uk

Abstract. We propose a computer graphics pipeline for 3D rendered cast shadow generation using generative adversarial networks (GANs). This work is inspired by the existing regression models as well as other convolutional neural networks such as the U-Net architectures which can be geared to produce believable global illumination effects. Here, we use a semi-supervised GANs model comprising of a PatchGAN and a conditional GAN which is then complemented by a U-Net structure. We have adopted this structure because of its training ability and the quality of the results that come forth. Unlike other forms of GANs, the chosen implementation utilises colour labels to generate believable visual coherence. We carried forth a series of experiments, through laboratory generated image sets, to explore the extent at which colour can create the correct shadows for a variety of 3D shadowed and un-shadowed images. Once an optimised model is achieved, we then apply high resolution image mappings to enhance the quality of the final render. As a result, we have established that the chosen GANs model can produce believable outputs with the correct cast shadows with plausible scores on PSNR and SSIM similarity index metrices.

Keywords: Cast shadows · Generative Adversarial Networks · Visualisation · Semi-supervised learning

1 Introduction

Shadow generation is a popular computer graphics topic. Depending on the level of realism required, the algorithms can be real-time such as Shadow mapping [23] and Shadow projection [3], or precomputed such as ray marching techniques, which are an expensive way to generate realistic shadows. Thus, pre-computing often become a compelling route to take, such as [28], [2] and [6]. Generative Adversarial Networks have been implemented widely to perform graphical tasks, as it requires minimum to no human interaction, which gives GANs a great advantage over conventional deep learning methods, such as image-to-image translation with single D, G semi-supervised model [7] or unsupervised dual learning [26].

We apply image-to-image translation to our own image set to generate correct cast shadows for 3D rendered images in a semi-supervised manner using

V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12141, pp. 481–495, 2020.
https://doi.org/10.1007/978-3-030-50426-7_36

colour labels. We then augment a high-resolution image to enhance the overall quality. This approach can be useful in real-time scenarios, such as in games and Augmented Reality applications, since recalling a pre-trained model is less costly in time and quality compared to 3D real-time rendering, which often sacrifices realism to enhance performance. Our approach eliminates the need to constantly render shadows within a 3D environment and only recalls a trained model at the image plane using colour maps, which could easily be generated in any 3D software. The model we use utilises a combination of PatchGAN and Conditional GAN, because of their ability to compensate for missing training data, and tailor the output image to the desired task.

There are many benefits of applying GANs to perform computer graphic tasks. GANs can interpret predictions from missing training data, which means a smaller training data set compared to classical deep learning models. GANs can operate in multi modal scenarios, and a single input can generalise to multiple correct answers that are acceptable. Also, the output images are sharper due to how GANs learn the cost function, which is based on real-fake basis rather than traditional deep learning models as they minimise the Euclidean distance by averaging all plausible outputs, which usually produces blurry results. Finally, GAN models do not require label annotations nor classifications.

The rest of this paper is structured as follows. Section 1 reviews related work in terms of traditional shadow algorithms, machine learning and GANs, then in Sect. 2 we explain the construction of the generative model we used. In Sect. 3 we present our experiments from general GAN to cGAN and DCGAN and ending with Pix2pix. In Sect. 4 we discuss our results. Then, in Sects. 5 and 6 we present the conclusion and future work, respectively.

1.1 Generative Networks

Recent advancements in machine learning has benefited computer graphics applications immensely. In terms of 3D representation, modelling chairs of different styles from large public domain 3D CAD models was proposed by [1]. [25] applied Deep Belief Network to create representation of volumetric shapes. Similarly, supervised learning can be used to generate chairs, tables, and cars using up-convolutional networks [5],[4]. Furthermore, the application of CNN extended into rendering techniques, such as enabling global illumination fast rendering in a single scene [19], and Image based relighting from small number of images [18]. Another application filters Monte Carlo noise using a non-linear regression model [8].

Deep convolutional inverse graphics network (DC-IGN) enabled producing variations of lighting and pose of the same object from a single image [10]. Such algorithms can provide full deep shading, by training end to end to produce dense per-pixel output [16]. One of the recent methods applies multiple algorithms to achieve real-time outputs, which is based on a recommendation system that learns the user's preference with the help of a CNN, and then allow the user to make adjustments using a latent space variant system [30].

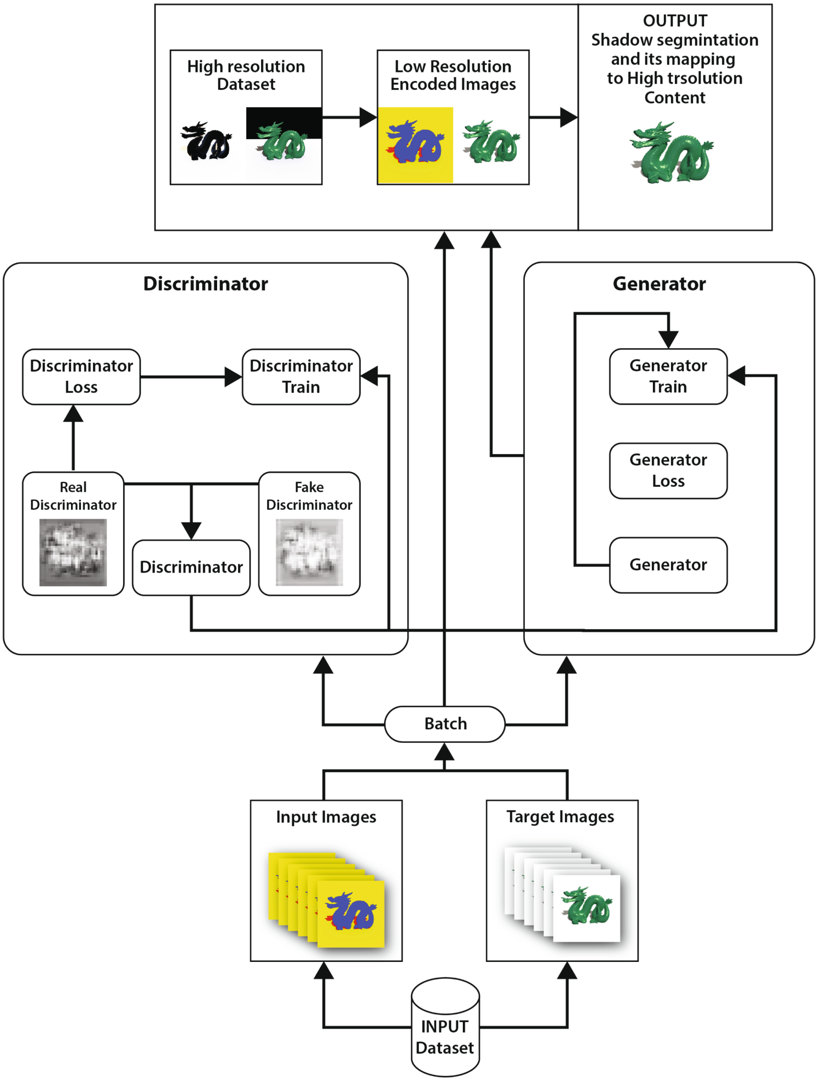


Fig. 1. The proposed framework emphasises on the combination of image pairs tailored to a specific function using pix2pix, producing high resolution content.

GANs have been implemented widely to perform graphical tasks. In cGANs, for instance, feeding a condition into both D, G networks is essential to control output images [14]. Training the domain-discriminator to maintain relevancy between input image and generated image, to transfer input domain into a target domain in semantic level and generate target images at pixel level [27]. Semi-supervised and unsupervised models has been approached in various ways such as trading mutual information between observed and predicted categorical class information [20]. This can be done by enabling image translators to be trained from two unlabelled images from two domains [26], or by translating both the image and its corresponding attributes while maintaining the permutation invariance property of the instance [15], or even by training a generator along with a mask generator that models the missing data distribution [11]. [12] proposed an image-to-image translation framework based on Coupled GANs that learns a joint distribution of images in different domains by using images from the marginal distributions in individual domains. Also, [13] uses an unsupervised model that performs image-to-image translation from few images. [21] made use of other factors such as structure and style. Another method eliminated the need for image pairs, by training the distribution of $G : X \to Y$ until $G(X)$ is indistinguishable from the Y distribution which is demonstrated with [29]. Another example is by discovering cross-domain relations [9]. Another way is forcing the discriminator to produce class labels by predicting which of $N + 1$ class the input belongs to during training [17]. Another model can generate 3D objects from probabilistic space with volumetric CNN and GANs [24].

2 Proposed Network Structure

The model we use in this paper is a Tensorflow port of the Pytorch Image-to-Image translation presented by Isola et al. [7]. This approach can be generalised to any semi-supervised model. However, this model serves us better for its two network types; PatchGAN, which allows better learning for interpreting missing data and partial data generation. And conditional GAN, which allows semi-supervised learning to facilitate control over desired output images by using colour labels. It also replaces the traditional discriminator with a U-Net structure with a step, this serves two purposes. First, it solves the known drop model issue in traditional GAN structure. Second, it helps transfer more features across the bottle neck which reduces blurriness and outputs larger and higher quality images.

The objective of GAN is to train two networks (see Fig. 1) to learn the correct mapping function through gradient descent to produce outputs believable to the human eye y. The conditional GAN here learns from observed image x and the random noise z, such that,

$$y, G : \{x, z\} \to y. \tag{1}$$

Where x is a random noise vector, z is an observed image, y is the output image. The Generator G, and a Discriminator D operate on "real" or "fake" basis. This is achieved by training both networks simultaneously with different objectives, G is trained to produce as realistic images as possible, while D is trained to distinguish which are fake, thus conditional GAN ($cGAN$) can be expressed as,

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x, z))]. \tag{2}$$

The aim here for G to minimise the objective against the discriminator D which aims to maximise it such that,

$$G^* = arg \min_G \max_D \mathcal{L}_{cGAN}(G, D). \tag{3}$$

By comparing it to an unconditional variant where the discriminator does not observe x it becomes

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_y[logD(y)] + \mathbb{E}_{x,z}[log(1 - D(G(x, z))]. \tag{4}$$

Here the distance of $\mathcal{L}_{L1}$ is used instead of $\mathcal{L}_{L2}$ to reduce blurring such that,

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||1]. \tag{5}$$

The final objective becomes,

$$G^* = arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G). \tag{6}$$

Both networks follow the convolution-BatchNorm-ReLu structure. However, the generator differs by following the general U-Net structure, and the discriminator is based on Markov random fields. The application of a U-Net model allows better information flow across the network than the encoder-decoder model by adding skip connections over bottle necks between layer i and layer $n - i$, where n is the total number of layers, by concatenating all channels at layer i with the ones in $n - i$, thus, producing sharper images.

For discriminator D, a PatchGAN of $N \times N$ is applied to minimize blurry results by treating the image in small patches that are classified as real-fake across the image, then averaged to produce the accumulative results of D, such that the image is modelled as a Markov random field, Isola et al. [7] refers to this PatchGAN as a form of texture/style loss.

The optimisation process alternates descent steps between D and G, by training the model to maximize $\log D(x, G(x, z))$ and dividing the objective by 2 to slow the learning rate of D, minibatch Stochastic Gradient Descend with Adam solver is applied at the rate of $0 : 0002$ for learning, and its momentum parameters are set to $\beta_1 = 0.5, \beta_2 = 0 : 999$. This allows the discriminator to compare minibatch samples of both generated and real samples. The G network runs at the same setting as the training phase at inference time. Dropout and batch normalization to the test batch is applied at test time with batch size of 1. Finally, random jitter is applied by extending the 256×256 input image size to 286×286 and then crop back to its original size of 256×256. Further processing using Photoshop, is applied manually to enhance the quality of output image, by mapping a higher resolution render of the model over the output model image, thus delivering a more realistic final image.

3 Experiments

Here we report our initial experiments for shadow generation as well as minor shading functions to support it. Our approach is data driven; it focuses on adjusting the image set in every iteration to achieve the correct output. For that we manually created the conditions we intended to test. Also, our image set is created using Maya with Arnold renderer. All of our experiments are conducted on an HP Pavilion laptop with 2.60 GHz Intel core i7 processor, 8 Ghz of RAM and Nvidia Geforce GTX 960M graphics card.

3.1 Image-to-Image Approach

We start with the assumption that GANs can generate both soft and hard shadows on demand, using colour labels and given a relatively small training image set. Our evaluation is based on both real-fake basis as well as similarity index matrices. Real-fake implies that the images can be clearly evaluated visually, for the network itself does not allow poor quality images by design. The similarity index matrices applied here are proposed by [22], namely, PSNR which measure the peak signal to noise ratio, which is scored between 1 and 100, and SSIM which computes the ratio between the strength of the maximum achievable power of the reconstructed signal and the strength of the corrupted noisy signal, which is scored between 0 and 1.

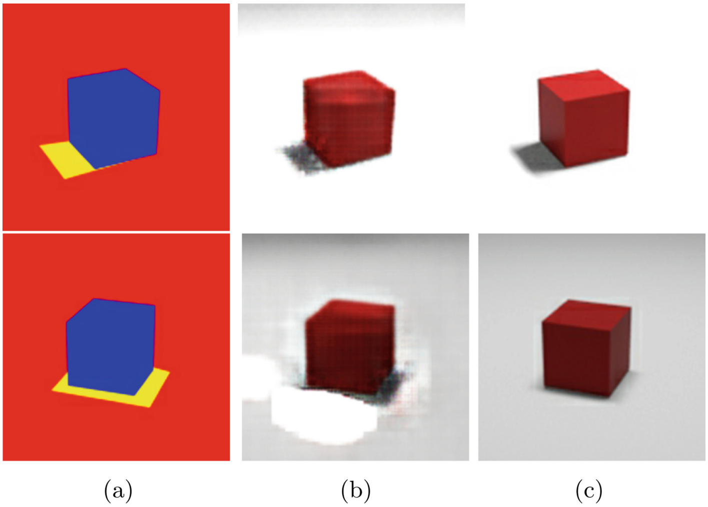


Fig. 2. Problematic output samples from the initial Image-to-image approach given a limited training image set, showing colour map (a), generated output (b) and target (c). Top (b) shows correct translation with some tiling effects due to the small training set, while bottom (b) is more problematic with incorrect environment translation such as large white patches. (Color figure online)

For our image-to-image translation approach, we started with a small image set of 100 images: 80 for training, 20 for testing and validation of random cube renders, with different lighting intensities and views. The results showed correct translations as shown in Fig. 2 (b) top section. However, the output colours of the background sometimes differed from the target images. Also, some of the environments contained patches and tiling artefacts as shown in Fig. 2 bottom section. Which is understandable given the small number of training images.

Next, we trained the model with Stonehenge images. It is also lit with a single light minus the variable intensity. The camera rotates 360 degrees around the Y axis. The total image number is 500 images, 300 for training and 200 for testing and validation.

We started with the question; can we generate shadows for non-shadowed images that are not seen during training? We worked around it by designing the colour label to our specific need. In the validation step we fed images with no shadows as in Fig. 3 (c), paired with colour labels that contains the correct shadows (a). As the results show in Fig. 3 (b), the model translated the correct shadow with the appropriate amount of drop off to the input image.

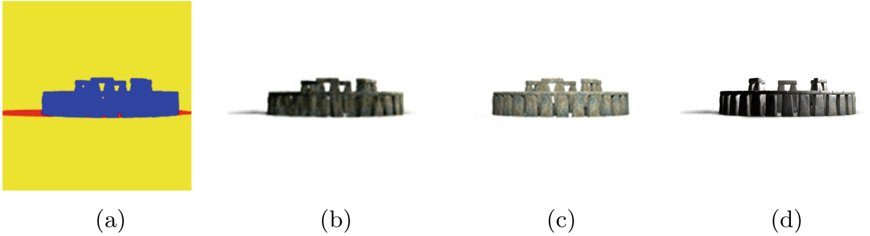


Fig. 3. Stonehenge samples showing the colour map (a), and correct translation of shadows (b) from non-shadowed images (c) and high resolution mapping (d). (Color figure online)

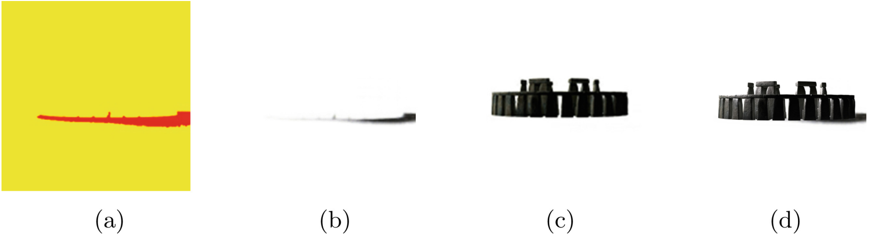


Fig. 4. Generating shadows with correct translation (b) for non-shadowed image (c) using only a shadow colour map (a) of the input image (c), then the output (b) is mapped to a high resolution image (d). (Color figure online)

Next we explore generating only accurate shadows Fig. 4 (b) for non-shadowed images (c), which is accomplished by constructing the colour map to only contain shadows (a), while training the network with full shadowed images as previously, paired with shadowed labels. The results show accurate translation of shadow direction and intensity (see Fig. 4 (b) and (d)).

For the third and final set of experiments, we used two render setups of the Stanford Dragon, one for training, the second for testing and validation. The camera setup that rotate 360 degrees around the dragon's Y axis with different step angle from training and testing. Also, the camera elevates 30 degrees across the X axis to show a more complex and elevated view of the dragon model rather than an orthographic one. The image set is composed of 1600 training images, 600 testing images and 800 for validation.

Table 1. Showing all the various categories applied in the final set of experiments.

Categories	Object		Colour map			Shadow			Position	
1	●	●	●	●			●	●	●	
2		●		●			●	●	●	
3	●	●	●			●	●			●
4		●	●				●		●	
5	●		●			●			●	
6	●			●			●	●	●	
7	●		●				●	●	●	
8		●		●			●		●	
9	●	●		●			●			●
10	●	●		●		●				●
11	●	●		●		●			●	
12	●	●	●	●			●		●	
13	●	●	●	●		●				●
14	●	●	●				●			●
15	●	●	●		●		●	●		●
16	●	●			●		●			●
17	●	●			●	●				●
18		●	●				●	●	●	
19		●	●				●		●	
20		●	●			●			●	
21	●	●		●		●			●	

● Categories that are represented in both training and testing phases.

● Non-Partial: Images with models completely visible in them.

● Partial: Images with models cut in some areas.

● Standard Colour Map: Images with original model and shadow colours, and white backgrounds.

● Coloured Background: Images with different background colours of which there are six, in addition to the standard white.

● Switched colour maps: images that are paired with colour maps that belongs to another category.

● Soft Shadows: models that are rendered with soft cast shadows.

● Hard Shadows: models that are rendered with hard cast shadows.

● Non-Shadowed: Models with cast shadows removed from them.

● Syncronized: Image pairs with the models and colour maps has the same position and rotation.

● Non-Syncronized: Image pairs with the models and colour maps has different position and rotation.

The training set (shown in Table 1) is broken into multiple categories, with each one represented within 200 images with overlapping features sometimes. These images help understand how input image/label affects the behaviour of the output images. For example, if we fed a standard colour map to a coloured image, will it be able to translate the colours across or will the output be of standard colour, this will better inform the training process for future applications. All networks are trained from scratch using our image sets, and the weights are initialized from a Gaussian distribution with mean 0 and standard deviation 0.02.

3.2 Testing and Validation

The test image set consisted of 600 images that are not seen in the training phase. During this, some of the features learnt from the training set are tested and the weights are adjusted accordingly.

For validation, a set of 800 images that are not seen in the training set were used, they are derived from the testing image set, but have been modified heavily. The objective here is to test the ability to generate soft and hard shadows from unseen label images, as well as colour shadows, partial shadows, shadow generation for images with no shadows. Also, in some cases we have taken the image set to extremes in order to generate shadows for images that has different orientation and colour maps than their original labels.

From here, the experiments progressed in three phases. First, we trained the model with the focus on standard labels to produce soft and hard shadows, using an image set of 1000 images, 400 of them are dedicated for standard colours and shadows. The remaining 600 images are an arbitrary collection of cases mentioned in the training section (Table 1), with similar arbitrary testing and validation image sets of 300 images each. In this experiment, we had the standard colours and shadows provide comprehensive 360-degree view of the 3D model. While purposefully choosing arbitrary coloured samples of 10–25 images, we created 6 colour variations for both images and their respective colour labels. Our first objective was to observe whether the model can generalise colour information into texture, meaning to fill the details of the image from what has been learnt from the 360 view, and overlaying colour information on top of it. Even though the general details can be seen on the models, there were heavy artefacts such as blurring and tiling in some of the coloured images with fewer training images.

With that knowledge in mind, a second experiment was carried out. We adjusted the image set to reduce the tiling effect, which is mainly due to the lack of sufficient training images for specific cases. Hence, the number of training images was increased to 200 per case, to increase the training set to 1600 images.

In the validation image set, we pushed the code to extreme cases, such as pairing images of different colour maps and different directions, as well as partial and non-shadowed images. Thus, accumulating the validation set to 800 images, assuming beforehand that we will get the same tiling effect from the previous experiment in cases where we have different angles or different colours.

4 Discussion

The significant training time is one of the main challenges that we face, as the training time for our set of experiments ranged between 3 days to one week using

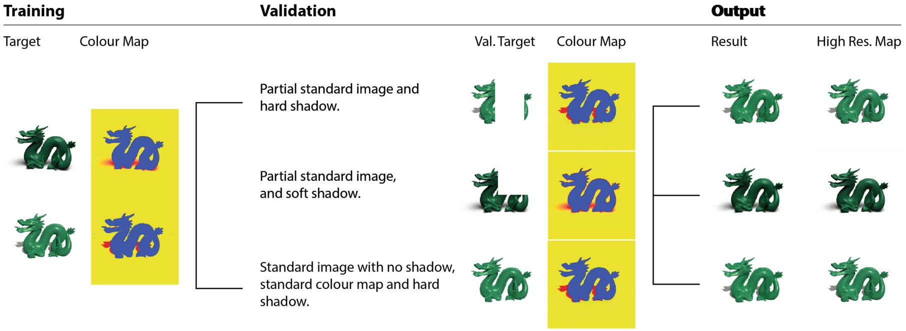

Training
Validation
Output
Target
Colour Map
Val. Target
Colour Map
Result
High Res. Map
Partial standard image and hard shadow.
Partial standard image, and soft shadow.
Standard image with no shadow, standard colour map and hard shadow.

Fig. 5. The initial phase of the third set, which shows direct translation tasks. Our focus here is mainly the ability to generate believable soft and hard shadows, as well as inpainting for missing patches.

our laptop, which is considered long for training 1600 images. This why our image set was limited to 256 × 256 pixels. For this work, we overcome the issue with augmenting the output image with a high-resolution render. Once trained, however, with some optimisation the model should be capable of real-time execution, but this issue has not been tested by us. The two biggest limitations for this method are, it is a still semi-supervised model that needs a pair of colour maps and a target image. The second limitation is that the colour maps and image pairs are manually created, and the process is labour intensive. These issues should be considered for future work.

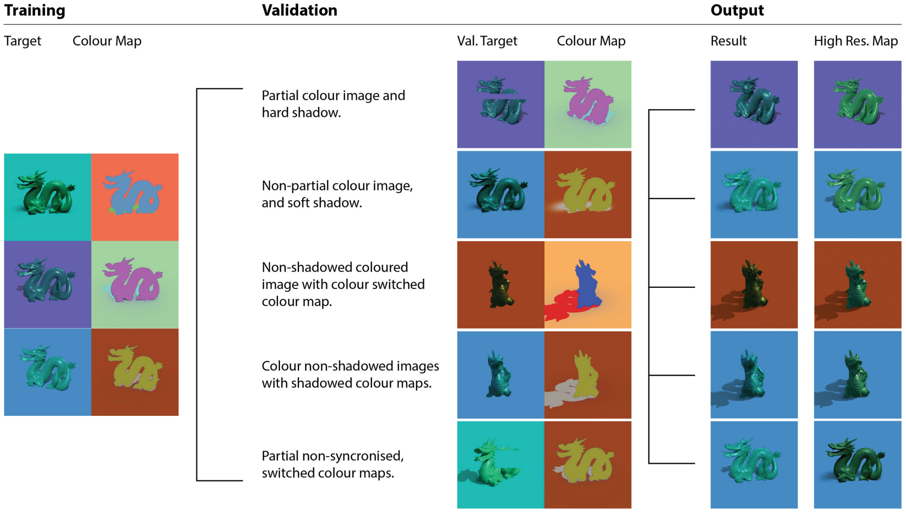

Training
Validation
Output
Target
Colour Map
Val. Target
Colour Map
Result
High Res. Map
Partial colour image and hard shadow.
Non-partial colour image, and soft shadow.
Non-shadowed coloured image with colour switched colour map.
Colour non-shadowed images with shadowed colour maps.
Partial non-syncronised, switched colour maps.

Fig. 6. Phase two of the third set focused on the interpretation of coloured shadows, since shadows are darker values of colours and not gray values. (Color figure online)

Our method performed well in almost all cases with minimal to no errors and sharp image reproduction, especially when faced with direct translation tasks, such as Fig. 5, and colour variations Fig. 6. Even with partial and non-shadowed images, the colours remained consistent and translated correctly across most outputs. This is promising, given a relatively small training set (approximately 200 images per case) we have used.

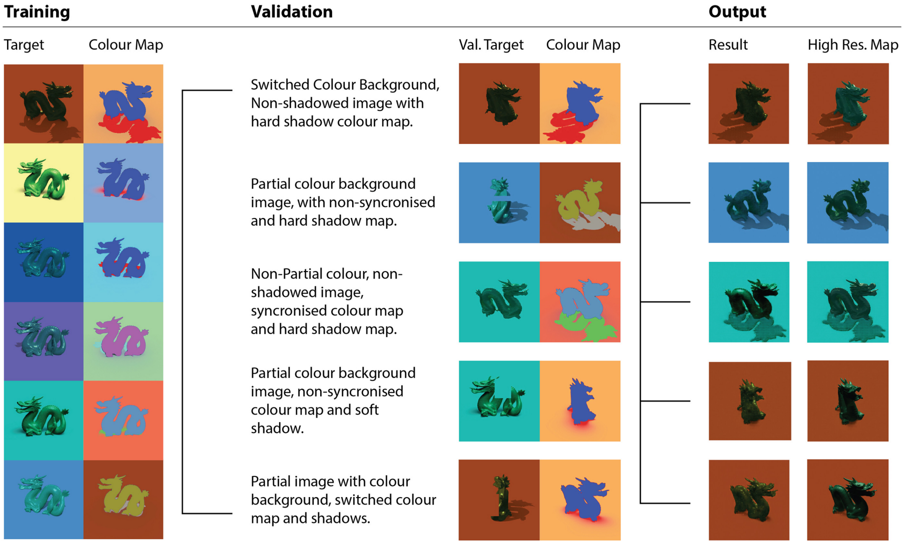
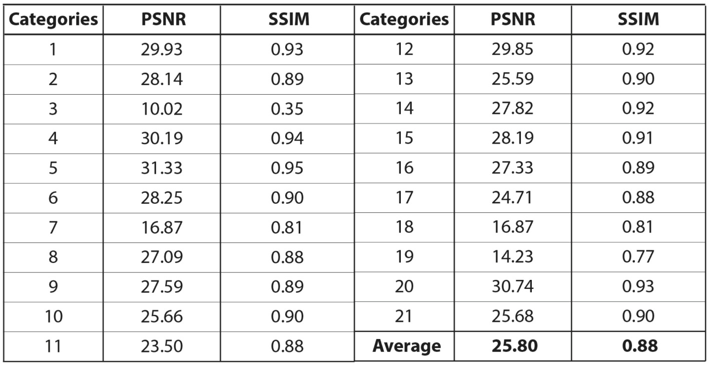

Training
Validation
Output
Target
Colour Map
Val. Target
Colour Map
Result
High Res. Map
Switched Colour Background, Non-shadowed image with hard shadow colour map.
Partial colour background image, with non-syncronised and hard shadow map.
Non-Partial colour, non-shadowed image, syncronised colour map and hard shadow map.
Partial colour background image, non-syncronised colour map and soft shadow.
Partial image with colour background, switched colour map and shadows.

Fig. 7. In the third and final phase of the third set, we pushed the code to its limits by switching targets, colour background and colour maps. (Color figure online)

By examining Fig. 7, we notice that the model generalised correctly in most cases even though the colour maps are non-synchronised. This means our method has the breadth to interpret correctly when training set fails. However, it tends to take a more liberal approach in translating colour difference between the label and target with bias towards the colour map. This was also visible in partial images, non-shadowed images, as well as soft and hard shadows. The network struggled mostly when more than two parameters are changed, for example, a partial image and non-shadowed model will translate well. However, partial image, non-synchronised shadow and position will start to show tiling in the output image. The model seems to struggle the most with position switching than any other change, especially when paired with non-synchronised colour map as well. This is usually manifested in the form of noise, blurring and tiling (see Fig. 7), while the colours remain consistent and true to training images, and the shadows are correct in shape and intensity but are produced with noise and tiling artefacts. We conducted our quantitative assessment by applying the similarity matrices PSNR and SSIM [22] and we can confirm the previous observations.

When looking at the Table 2, the lowest score were in the categories with non-synchronized image pairs such as categories 3 and 19, while the image pairs that were approximately present in both training and testing performed the highest, which are categories 4 and 5, with overall performance leaning towards a higher scores spectrum.

Table 2. This table shows how each category performed in both PSNR and SSIM similarity indices between output images and corresponding ground truth images.



Categories	PSNR	SSIM	Categories	PSNR	SSIM
1	29.93	0.93	12	29.85	0.92
2	28.14	0.89	13	25.59	0.90
3	10.02	0.35	14	27.82	0.92
4	30.19	0.94	15	28.19	0.91
5	31.33	0.95	16	27.33	0.89
6	28.25	0.90	17	24.71	0.88
7	16.87	0.81	18	16.87	0.81
8	27.09	0.88	19	14.23	0.77
9	27.59	0.89	20	30.74	0.93
10	25.66	0.90	21	25.68	0.90
11	23.50	0.88	**Average**	**25.80**	**0.88**

5 Conclusion

This paper explored a framework based on conditional GANs using a pix2pix Tensorflow port to perform computer graphic functions, by instructing the network to successfully generate shadows for 3D rendered images given training images paired with conditional colour labels. To achieve this, a variety of image sets were created using an off-the-shelf 3D program.

The first set targeted soft and hard shadows under standard conditions and coloured labels and backgrounds, using 6 colour variations in the training set to test different variations, such as partial and non-shadowed images. The image set consisted of 1000 training images and 600 images for testing and validation. The results were plausible in most cases but showed clear blurring and tiling with coloured samples that did not have enough training images paired with it.

Next, we updated the image set to 3000 images, with 1600 training images, providing an equal number of training images for each of the 8 cases. We used 600 images for testing, and 800 for validation, which included more variations such as partial and non-shadowed images. In the validation set, the images included extreme cases such as non-sync pairing of position and colour. The results were believable in all cases, except the extreme cases, which resulted in tiling and blurring.

The results are promising for shadow generation especially when challenged to produce accurate partial shadows from training image set. The model is reliable to interpret successful output for images not seen during the training phase, except when paired with different colours viewpoints. However, there are still challenges to resolve. For example, the model requires a relatively long time to be trained and the output images still suffer from minor blurriness.

6 Future Work

This is only a proof of concept. The next logical step is to optimise the process by training the model to create highly detailed renders from lower poly-count models. This also can be tested with video-based models such as HDGAN. It is expected to output flickering results due to its learning nature and current state of the art. Another direction of interest may be to automate the generation of colour maps from video or live feed such as the work in [29]. The main challenge, however, is the computation complexity, especially for higher resolution training.

References

1. Aubry, M., Maturana, D., Efros, A.A., Russell, B.C., Sivic, J.: Seeing 3D chairs: exemplar part-based 2D–3D alignment using a large dataset of cad models. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3762–3769 (2014)
2. Baran, I., Chen, J., Ragan-Kelley, J., Durand, F., Lehtinen, J.: A hierarchical volumetric shadow algorithm for single scattering. In: ACM Transactions on Graphics (TOG), vol. 29, p. 178. ACM (2010)
3. Blinn, J.: Me and my (fake) shadow. IEEE Comput. Graph. Appl. **8**(1), 82–86 (1988)
4. Dosovitskiy, A., Springenberg, J.T., Tatarchenko, M., Brox, T.: Learning to generate chairs, tables and cars with convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 692–705 (2017)
5. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1538–1546 (2015)
6. Herzog, R., Eisemann, E., Myszkowski, K., Seidel, H.P.: Spatio-temporal upsampling on the GPU. In: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 91–98. ACM (2010)
7. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
8. Kalantari, N.K., Bako, S., Sen, P.: A machine learning approach for filtering Monte Carlo noise. ACM Trans. Graph. **34**(4), 122 (2015)
9. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1857–1865. JMLR. org (2017)

10. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. In: Advances in Neural Information Processing Systems, pp. 2539–2547 (2015)
11. Li, S.C.X., Jiang, B., Marlin, B.: MisGAN: learning from incomplete data with generative adversarial networks. arXiv preprint arXiv:1902.09599 (2019)
12. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems, pp. 700–708 (2017)
13. Liu, M.Y., et al.: Few-shot unsupervised image-to-image translation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 10551–10560 (2019)
14. Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR abs/1411.1784 (2014). http://arxiv.org/abs/1411.1784
15. Mo, S., Cho, M., Shin, J.: InstaGAN: instance-aware image-to-image translation. arXiv preprint arXiv:1812.10889 (2018)
16. Nalbach, O., Arabadzhiyska, E., Mehta, D., Seidel, H.P., Ritschel, T.: Deep shading: convolutional neural networks for screen-space shading. Comput. Graph. **36**(4), 65–78 (2017)
17. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 2642–2651. JMLR. org (2017)
18. Ren, P., Dong, Y., Lin, S., Tong, X., Guo, B.: Image based relighting using neural networks. ACM Trans. Graph. **34**(4), 111:1–111:12 (2015)
19. Ren, P., Wang, J., Gong, M., Lin, S., Tong, X., Guo, B.: Global illumination with radiance regression functions. ACM Trans. Graph. **32**(4), 130:1–130:12 (2013)
20. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv preprint arXiv:1511.06390 (2015)
21. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 318–335. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_20
22. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
23. Williams, L.: Casting curved shadows on curved surfaces. SIGGRAPH Comput. Graph. **12**(3), 270–274 (1978)
24. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in neural information processing systems, pp. 82–90 (2016)
25. Wu, Z., et al.: 3D ShapeNets: a deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1912–1920 (2015)
26. Yi, Z., Zhang, H., Tan, P., Gong, M.: DualGAN: unsupervised dual learning for image-to-image translation. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2868–2876 (2017)
27. Yoo, D., Kim, N., Park, S., Paek, A.S., Kweon, I.S.: Pixel-level domain transfer. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 517–532. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_31
28. Zhou, K., Hu, Y., Lin, S., Guo, B., Shum, H.Y.: Precomputed shadow fields for dynamic scenes. In: ACM Transactions on Graphics (TOG), vol. 24, pp. 1196–1201. ACM (2005)

29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp. 2223–2232 (2017)
30. Zsolnai-Fehér, K., Wonka, P., Wimmer, M.: Gaussian material synthesis. ACM Trans. Graph. (TOG) **37**(4), 76 (2018)