2D to 3D Animation Style Transfer

Swathi Iyer
Department of Computer Science
Stanford University
swathii@stanford.edu

Timothy Le
Department of Computer Science
Stanford University
tle7@stanford.edu

Abstract

This project aims to transfer 3D animation style onto 2D animated frames. We train various style transfer models, including feedforward neural style transfer, Pix2Pix, and CycleGAN on the captured frames of various 2D and 3D animated clips. We then evaluate the performance of these models on their ability to transfer 3D style onto new 2D frames. We found that the feedforward model had the best results only when the style image is similar to the content image. Otherwise, the Pix2Pix model was able to capture the style best.

1 Introduction

3D animation is becoming increasingly popular amongst designers and gamers due to its realistic nature, which allows for a more engaging and immersive experience. However, it also suffers from high development cost in terms of time and resources, relative to 2D animation. 2D animation, while less in demand and often perceived to be lacking depth, has the advantages of being quicker and easier to produce, and having a lower production cost in terms of time, required skills, and resources. For our project, we wanted to explore how we might transform low-cost 2D animations to 3D using style transfer methods. This is an interesting task in that if successful, can act as another method of producing 3D animations through 2D animation with lower production costs. This is a challenging task, however, in that animations often lack enough depth for style transfer methods to pick up upon. In this paper, we explore various style transfer methods such as feedforward neural style transfer, Pix2Pix, and CycleGAN with 2D and 3D animation frames, to generate new 3D animated frames from their corresponding 2D versions. We evaluate the performance of these models on their generated image quality and discuss their tradeoffs.

2 Related work

This original style transfer approach [2] uses a network based on the image classification 19-layer VGG network to take in a content and a style image and output the content image in the style of the style image. One characteristic of the results in this paper is that the style images in general are very detailed. We hypothesize that the detail of these style images helps this network's performance. By contrast, 3D animation frames in general are not as detailed as the style images in [2].
This model in [3] uses a CycleGAN architecture that can be trained using multiple style domains. The model is able to convert an input image to an output that combines the styles of these multiple domains such as combining styles from multiple 3D movies. However, this approach was difficult to utilize because the code was not posted at the time although there was a repository for it. We therefore use a basic CycleGAN approach in our "Methods" section.
The authors in [8] have exciting work in transferring a 2D human figure to a 3D mesh of the human that allows the human to perform tasks such as walking around. This architecture uses a convolutional neural network architecture for segmenting the human and uses a patch-based algorithm for filling in occluded body parts. We found this work's ability to be able to capture the 3D aspect of a 2D human

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

figure to be exciting. However, this work does not allow us to accomplish our task directly. We are trying to produce a single 3D frame, not a 3D mesh. Additionally, this work does not necessarily generalize to animating non-human cartoon characters as well as the background. We could explore concepts of this architecture in order to capture 3D characteristics in the future.

The authors in [4] use image analogies to apply to an input image. Using an autoregression model, this approach takes in a style A and an output style A'; this mapping from A to A' is applied to a new input image B so that B can be converted to B'. While this work requires manually collecting pairs of input and output style to learn analogies, this work demonstrates the effectiveness of having paired images in order for a model to learn a mapping.

The work in [1] features manually transforming 2D cartoon clips of the TV show *Rick and Morty* to 3D clips. The author of this work took sequential frames of 2D clips and used 3D animation software by overlaying these 2D frames and edited them. This appears to require much manual work while our task seeks to use deep learning to reduce the amount of manual work needed.

3 Dataset and Features

All of our frames are from YouTube videos in which we did not run into copyright problems downloading. We used a tool called VLC media player to extract our frames. We adjusted the rate at which frames were collected depending on how many images we needed; for example the feedforward neural style transfer approach did not need as many frames as Pix2Pix and CycleGAN. Our initial approach featured collecting a few frames from the movie *The Lion King* (1994) as our content image and a few frames from the movie *Monster's, Inc.* (2001) as our style image to test how the feedforward neural style transfer would perform on images with different environments whereas CycleGAN and Pix2Pix require images with similar environments.

The pix2pix model required us to have exact frames in 2 different styles, in order to learn the mappings between two images styles. To do this, we found an animator's video [1] of his manual recreation of the 2D animated show *Rick and Morty* in 3D animation, and extracted the frames. In total, we extracted 82 frames, and divided the data into train/val/test splits of 80/10/10. We then used an online photo editing tool called Birme to standardize image dimensions.

For the CycleGAN method, for our training set, we have collected 53 frames from the 2D animated *The Little Mermaid: Ariel's Beginning* movie, and 48 frames from the 3D animated *Finding Dory*. As mentioned in [9], while there do not need to be explicit pairings for images for CycleGAN, best results occur if the two domains share similar visual content. We therefore chose our content domain to be frames from *The Little Mermaid: Ariel's beginning* (2008) and our style domain to be frames from *Finding Dory* (2016) because both have underwater scenes. The low number of training set frames is a result of trying to have similar scenes in both movies. Primarily, we needed to remove scenes with mermaids because *Finding Dory* did not have underwater humans, and fast action scenes resulted in blurry images. After training our model, we found more scenes that had similar environments, so our test set included 45 frames from *The Little Mermaid: Ariel's Beginning* and 81 frames from *Finding Dory*. This small dataset is a limitation of using CycleGAN in our project.

4 Methods

4.1 Feedforward Neural Style Transfer

The approach in [2] uses a network that has been pretrained on VGG. The network forms a representation of the content image and a representation of the style image. In order to generate an image that incorporates the content of one image and the style of another image, the network minimizes the distance of a whitenoise image and the content representation as well as the distance of the whitenoise image and the style representation. The loss function has weights α and β that are used to factor how much the content versus the style representations affect the final outcome. The loss is:

$$L_{total}(\vec{p}\,,\vec{a}\,,\vec{x}\,) = \alpha L_{content}(\vec{p}\,,\vec{x}\,) + \beta L_{style}(\vec{a}\,,\vec{x}\,)$$

,

where $\vec{p}$ corresponds to the content image, $\vec{a}$ corresponds to the style image, and $\vec{x}$ corresponds to the resulting image. As mentioned above, α and β are used to weight how much the content and style representations affect the outcome. We applied the Github repository in [6].

4.2 Pix2Pix

Next, we tried applying the Pix2Pix model in [5] by utilizing their Github repository. This model learns a mapping, $G : \{x, z\} \rightarrow y$, from domain X and random noise vector z to domain Y, and then applies mapping to generate new images in domain Y.

We found that this model would be applicable to our particular task as well, where we set domain X to be 2D animated frames, and domain Y to be 3D animated frames. Because this model requires exact pairings of images from the 2D to 3D animated style, we collected new data from frames of an animator's manual translations of the 2D animated show *Rick and Morty* to 3D animations to train this model.

This approach uses a conditional GAN, where a discriminator D learns to distinguish between real and fake images (i.e. images produced by the generator), while the generator G is trained to produce images that can fool the discriminator. In contrast to unconditional GANs that assume pixels in the image to be conditionally independent from one another, conditional GANs use a loss that penalizes the combination of output pixels. The loss for generator G and discriminator G is:

$$L_{cGAN}(G, D) = \mathop{\mathbb{E}}_{x,y}\Big[\log(D(x, y))\Big] + \mathop{\mathbb{E}}_{x,z}\Big[\log(1 - D(x, G(x, z)))\Big]$$

Here, G tries to minimize the loss against an adversarial D that tries to maximize it. Therefore, the final objective of generator G is:

$$G^* = \text{arg min}_G \text{max}_D L_{cGAN}(G, D)$$

The architectures of the generator and discriminator are both Conv-BatchNorm-ReLU blocks in a U-Net structure with skip connections between mirrored layers (adapted from [CHANGE NUMBER][10]). This allows for low-level information, common to be important for image translations problems, to be directly shared across the network.

4.3 CycleGAN

We chose to apply the CycleGAN approach in [10] to our task because this approach is meant to find a mapping from domain X to domain Y without the images being paired. This model assumes that there is some underlying mapping from domain X to domain Y. We used the following Github repository in [9]to apply this approach.

The approach in [10] uses two mapping functions $G : X \rightarrow Y$ to translate images in domain X to domain Y and $F : Y \rightarrow X$ to translate images in domain Y to domain X. There are additionally two discriminator functions D_X and D_Y that are used to discriminate whether of not an image is in a respective domain.

Each mapping function and its associated discriminator function has a generative adversarial loss. The loss for G and the associated discriminator D_Y is:

$$L_{GAN}(G, D_Y, X, Y) = \mathop{\mathbb{E}}_{y \sim p_{data}(y)}\Big[\log(D_Y(y))\Big] + \mathop{\mathbb{E}}_{x \sim p_{data}(x)}\Big[\log(1 - D_Y(G(x)))\Big]$$

(There is also a similar loss $L_{GAN}(F, D_X, Y, X)$ for F and the discriminator D_X.)

In addition to these functions, there is a cycle-consistency loss with a forward cycle-consistency loss and a backward cycle-consistency loss. The forward cycle-consistency loss ensures that for an image x in domain X, $F(G(x)) \approx x$. The backward cycle-consistency loss ensures that for an image y in domain Y, $G(F(y)) \approx y$. Each cycle-consistency loss is intended to make G and F consistent with one another. This loss is expressed as follows:

$$L_{cyc}(G, F) = \mathop{\mathbb{E}}_{x \sim p_{data}(x)}\Big[||F(G(x)) - x||_1\Big] + \mathop{\mathbb{E}}_{y \sim p_{data}(y)}\Big[||G(F(y)) - y||_1\Big]$$

The full objective loss is:

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F)$$

,

where λ is a factor that controls the relative importance of L_{cyc} to that of the L_{GAN} terms. The goal of CycleGAN is to find a G^* and F^* that solves the following:

$$\text{arg min}_{G,F} \ \text{max}_{D_X, D_Y} L(G, F, D_X, D_Y)$$

5 Experiments/Results/Discussion

5.1 Evaluation Metric

We will evaluate our models based on a qualitative analysis of their output images, as well as on the loss values. Using qualitative results is also done in [10] and [7]. We will report loss scores for the feedforward network, but we note that the scores do not necessarily correlate with the output image quality. For example, the generated image in Figure 1 captures the 3D aspect better than the generated image in Figure 2, but Figure 1's output has a higher loss.

Figures 1 and 2 show the results of the feedforward neural style transfer, using style images that are both similar and different from the content image, respectively. Figure 4 shows the results of the Pix2Pix model trained on paired *Rick and Morty* frames. Lastly, Figure 5 shows the results of CycleGAN trained on *Little Mermaid* and *Finding Dory*.

5.2 Feedforward Neural Style Transfer

We trained the feedforward model for 1000 epochs with a learning rate of 1e1 using the Adam optimizer. We found that the feedforward model was the best at capturing the 3D style while preserving the content *if* the style image was similar to the content image. The *Rick and Morty* images, for example, gave results that very much looked 3D (Figure 1), as opposed to the *Little Mermaid* and *Finding Dory* frames (Figure 2), for which the model seemed to capture some of the colors and preserved the content image really well, but didn't necessarily capture the 3D style. While the need for a similar style image is constraining, this result could still mean that fewer frames of a movie need to be created in 3D (e.g. perhaps only one frame per scene could be created in 3D, and applied to the rest of the frames created in 2D). In addition, this model has advantages over Pix2Pix and CycleGAN, which require us to have several paired frames, or several frames from the same domain, respectively, in constrast to feedforward, which only requires one style and content image, and relies on a pretrained model. Figure 3 demonstrates how the feedforward model does not perform well when the content and style images do not have similar content; the content for the outputted image is generally preserved, but the new style applied is not coherent with the entire output image. We note that training for more epochs as well as using higher quality images did not improve results significantly and decreased the loss by a minor amount.

The results in [2] feature style images that are detailed art works in general, but 3D animation frames are not detailed in a fine-grain manner. Therefore the lack of great results in Figures 2 and 3 may be due to relatively low-detail style images. Figure 1 may feature better results because most of the style and content frames map well to one another, so the model can map the coloring and the detail of the style image well to the content image despite a relatively low-detailed style image.

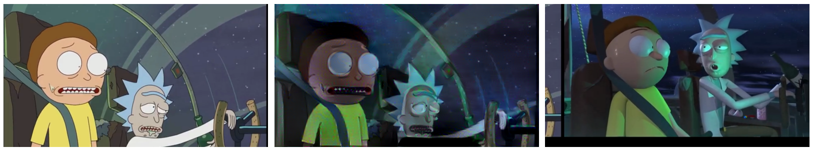

Figure 1: Feedforward Neural Style Transfer results for Rick and Morty frames; content 2D frame (left), generated 3D frame (middle), style 3D frame (right); Loss: 548,076

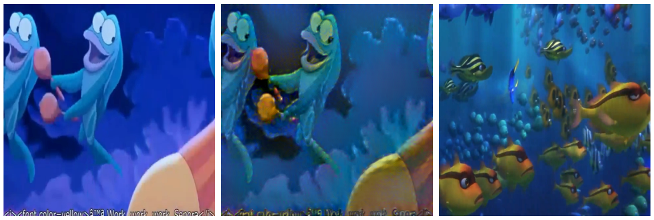

Figure 2: Feedforward Neural Style Transfer results for The Little Mermaid and Finding Dory Frames; Little Mermaid 2D frame (left), generated 3D frame (middle), Finding Dory 3D frame (right); Loss: about 462,063

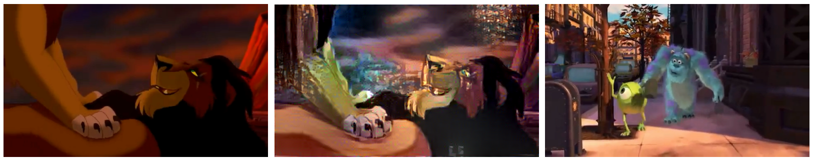

Figure 3: Feedforward Neural Style Transfer results for The Lion King and Monsters, Inc.; Lion King 2D frame (left), generated 3D frame (middle), Monsters, Inc. 3D frame (right); Loss: about 718,971

5.3 Pix2Pix

The Pix2Pix model captured the 3D style of the animation very well, but was not as effective at preserving the content. We see a general outline of all the features of the characters in the 3D style, but the outputs are not very defined and are still blurry. One thing we believe might be causing this is that the true 2D and 3D frames are not exactly aligned. Perfectly aligned data is difficult to find as well as to produce, but testing this would be a good next step.

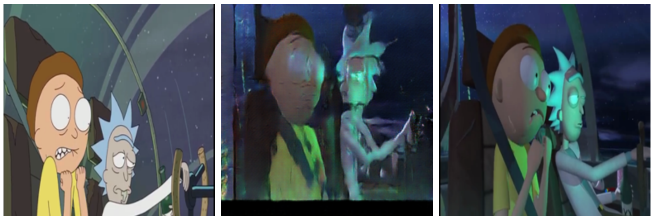

Figure 4: Pix2Pix Results; real 2D frame (left), generated 3D frame (middle), real 3D frame (right)

5.4 CycleGAN

We ran the CycleGAN model for 200 epochs with the Adam optimizer using an initial learning rate of 0.0002. We used $\lambda = 10$. Despite having few training images, we see in Figure 5 that the content image is preserved for the most part in the generated image, but the style is not captured well. While an advantage of this approach is that the model is trained on images from our style and content domains and does not need to be paired, this lack of a quality generated image may be due to the data collection challenges mentioned in the “Dataset" section. One metric of CycleGAN’s performance is the recovered image in Figure 5 (the third image from the left); we see that although the edges of the image are preserved, the image is very blurry. Having more training data would likely improve the mapping from *The Little Mermaid* to *Finding Dory* and back to *The Little Mermaid*.

6 Conclusion/Future Work

Of all the models we tried, feedforward seemed to have the best results when the style image was similar to the content image. It preserved the content, while still producing images with the 3D style. It was not, however, able to capture this style when the style image was significantly different. Pix2Pix seemed to capture the 3D style extremely well, however, the content was quite distorted, possibly due to the misalignment of the domain X and domain Y images. Testing this model with better aligned pictures is something we would like to explore more in the future. Although CycleGAN

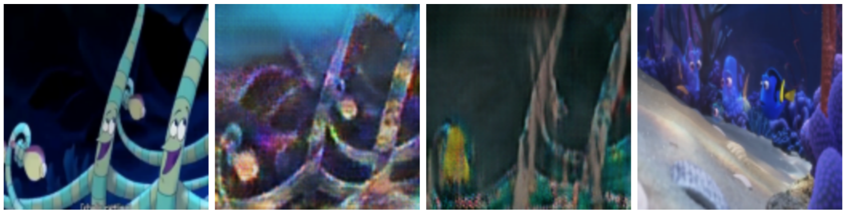

Figure 5: CycleGAN results left to right: Little Mermaid 2D frame, generated 3D frame, 2D frame recovered from generated 3D frame, and 3D Finding Dory frame

does not need paired images, it is still a challenge to gather images with similar environments. As a future work, we would apply data augmentation techniques such as horizontally flipping the images. Additionally, because much of our project has focused on finding an architecture that suited our task well, with more time, we would like to experiment with modifying portions of a certain architecture.

7 Contributions

Swathi: Trained images on the feedforward neural style transfer model and the pix2pix model. Helped with data collection and preprocessing for pix2pix model.
Timothy: Data collection and preprocessing. Helped with training images on the feedforward model. Trained images on the CycleGAN model.

References

[1] Den Beauvais. denbeauvais.com.

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[3] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. DLOW: domain flow for adaptation and generalization. *CoRR*, abs/1812.05418, 2018.

[4] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[6] Justin Johnson. neural-style. `https://github.com/jcjohnson/neural-style`, 2015.

[7] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[8] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. *CoRR*, abs/1812.02246, 2018.

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Cyclegan. `https://github.com/junyanz/CycleGAN`, 2017.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.