Artist-Guided Semiautomatic Animation Colorization

Harrish Thasarathan and Mehran Ebrahimi
University of Ontario Institute of Technology
Oshawa, Ontario, Canada
harrish.thasarathan@uoit.net mehran.ebrahimi@uoit.net
http://www.ImagingLab.ca

Abstract

There is a delicate balance between automating repetitive work in creative domains while staying true to an artist's vision. The animation industry regularly outsources large animation workloads to foreign countries where labor is inexpensive and long hours are common. Automating part of this process can be incredibly useful for reducing costs and creating manageable workloads for major animation studios and outsourced artists. We present a method for automating line art colorization by keeping artists in the loop to successfully reduce this workload while staying true to an artist's vision. By incorporating color hints and temporal information to an adversarial image-to-image framework, we show that it is possible to meet the balance between automation and authenticity through artist's input to generate colored frames with temporal consistency.

1. Introduction

From multi-generational children's cartoons to modern adult classics, 2D animation continues to have a large impact on global television viewership. This is especially true with the recent growth of Japanese animation (anime). While traditional 2D animation was painstakingly hand drawn and painted frame by frame on thousands of pieces of parchment, its renaissance in the modern era has seen the field perfectly blend creativity with new technologies to expedite workflows. Even in its modern form, the frame by frame animation process remains a tedious endeavour. Developing these shows is a multistage process that requires multiple teams of artists simultaneously working on multiple tasks to meet production timelines. Most animation studios outsource large portions of their workloads to South Korea. When a single episode requires drawing and coloring between three to ten thousand frames, making corrections and alterations with overseas communication results in an 8-12 month production cycle per episode [1]. The drawing of key frames that define major character movements are performed by lead artists while in-between frames that fill in these motions are animated by inexperienced artists. The colorization of line art sketches from key frames and in-between frames is considered to be tedious, repetitive, and low pay work. Thus finding an automatic pipeline to consistently produce thousands of colored frames from line art frames can be incredibly valuable to animation studios globally to expedite workflows.

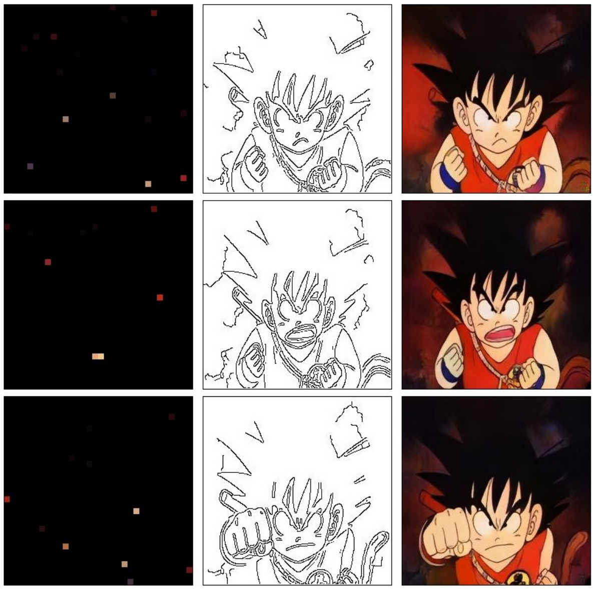

Figure 1. (Left) Color hint. (Center) Computed synthetic line art. (Right) Generated colorized temporally coherent frame from line art.

The image to image translation method presented by Isola *et al.* [8] uses Generative Adversarial Networks (GANs) conditioned on some input to learn a mapping from input to target image by minimizing a loss function. The mapping learned from the proposed technique works with some sparse inputs and has been used in applications such as converting segmentation labels to street view images. We aim to extend this model to work with synthetic line arts.

Our model incorporates temporal information by encouraging consistency between colorized frames produced by the generator through discriminating against frame sequences.

2. Related Works

Image to image translation using conditional GANs [6, 12] is especially effective in comparison to CNN-based models for colorization tasks [11]. This model successfully maps a high dimensional input to a high dimensional output using a U-Net [13] based generator and patch-based discriminator [8]. The closer the input image is to the target, the better the learned mapping is. As a result, this technique is particularly suited to colorization tasks. User-guided colorization methods [17] have also shown to be highly effective when the true color of an image is ambiguous and more accurate colors are required.

The neural algorithm for artistic style presented by Gatys *et al*. [5] provides a method for the creation of artistic imagery. This is relevant as it demonstrates a way to learn representations of both content and style between two images using the pretrained VGG network [14] and transfer that learned representation to a target image. The ability to learn and differentiate style and content using a pretrained network can supplement training for our purposes.

There are very few existing methods for line art colorization. These methods are made to be highly generalized for use by independent artists and also incorporate scribble hints [2]. As a result, for production settings where sequential frames are related to each other, colorizing these frames independently creates a flicker effect from the inconsistencies between generated frames when made into a video. Additionally, providing detailed hints for every frame can be tedious. For our use case, a method that learns representations of recurring places and characters while keeping hints at a minimum is the most efficient in production.

3. Model

Our method attempts to colorize line art frames for large-scale production settings by taking into consideration temporal information to create consistency between frames. We also aim to keep artists in the loop by incorporating color hints. This allows for greater control by animators and accounts for variations that may be required during realistic production scenarios $i.e.$ new characters, new locations, existing characters wearing new clothes, etc.

Our generator is comprised of 2 downsampling layers followed by 8 residual blocks [7] and 2 upsampling layers back to the input size. Instance normalization [16] is used in place of batch normalization because we use smaller batches during training. Our discriminator architecture uses a 70×70 patch GAN as in Isola *et al.* [8] who map generator predictions to a scale of $N \times N$ outputs and classifies each patch as real or fake. Our discriminator also takes advantage of spectral normalization to stabilize training as shown in [10].

In Chan *et al*. [4] temporal information is introduced in a cGAN architecture by adding extra conditions on their generator and discriminator. The current input and previously generated image provide conditions for their generator. The discriminator then differentiates between real and fake frame sequences instead of individual images. We take inspiration from this method and incorporate temporal information in our network with added conditions to our generator and discriminator.

3.1. Loss Objective

Let G and D represent the generator and discriminator of our colorization network. Our generator takes a greyscale or line art image $\mathbf{I}_{line}$ as input conditioned on the previous generated color frame prediction $\mathbf{F}_{t-1}$ and a color hint map $\mathbf{I}_{hint}$ to return a color prediction $\mathbf{F}_t$ temporally consistent to the previously colored frame prediction.

$$\mathbf{F}_t = G(\mathbf{I}_{line}, \mathbf{I}_{hint}, \mathbf{F}_{t-1}). \tag{1}$$

A joint loss is used to train the network that takes advantage of both conditional GAN and neural style algorithm that consists of an adversarial loss, style loss, content loss, and l_1 loss. The adversarial loss is defined in equation (2) where the generator is encouraged to learn real sequences of colored frames because the discriminator differentiates sequences rather than individual images.

$$\begin{aligned}\mathcal{L}_{adv} &= \mathbb{E}_{(\mathbf{I}_{line}, \mathbf{F}_{gt})}\left[\log D(\mathbf{I}_{line}^{(t-1)}, \mathbf{F}_{gt}^{(t-1)}, \mathbf{I}_{line}^{(t)}, \mathbf{F}_{gt})\right] \\ &+ \mathbb{E}_{(\mathbf{I}_{line})} \log\left[1 - D(\mathbf{I}_{line}^{(t-1)}, \mathbf{F}_{t-1}, \mathbf{I}_{line}^{(t)}, \mathbf{F}_t)\right]. \end{aligned} \tag{2}$$

We incorporate content and style loss described in [5, 9] to further supplement the training of our colorization network. Content loss encourages perceptual similarity while style loss encourages texture similarities between predicted and ground truth color frames. Perceptual similarity is accomplished by minimizing the Manhattan distance between feature maps generated by intermediate layers of VGG-19. Content loss is defined by equation (3) where ϕ_i represents the activation map at a given layer i of VGG-19, $\mathbf{F}_{gt}$ is the current ground truth color frame, and $\mathbf{F}_t$ is the generated frame. N_i represents the number of elements in the i^{th} activation layer of VGG-19. For our work, we use activation maps from layers `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1` and `relu5_1`

$$\mathcal{L}_{cont} = \mathbb{E}_i\left[\frac{1}{N_i}\left\|\phi_i(\mathbf{F}_{gt}) - \phi_i(\mathbf{F}_t)\right\|_1\right]. \tag{3}$$

Style loss is calculated in a similar manner but rather than calculating the Manhattan distance between feature maps,

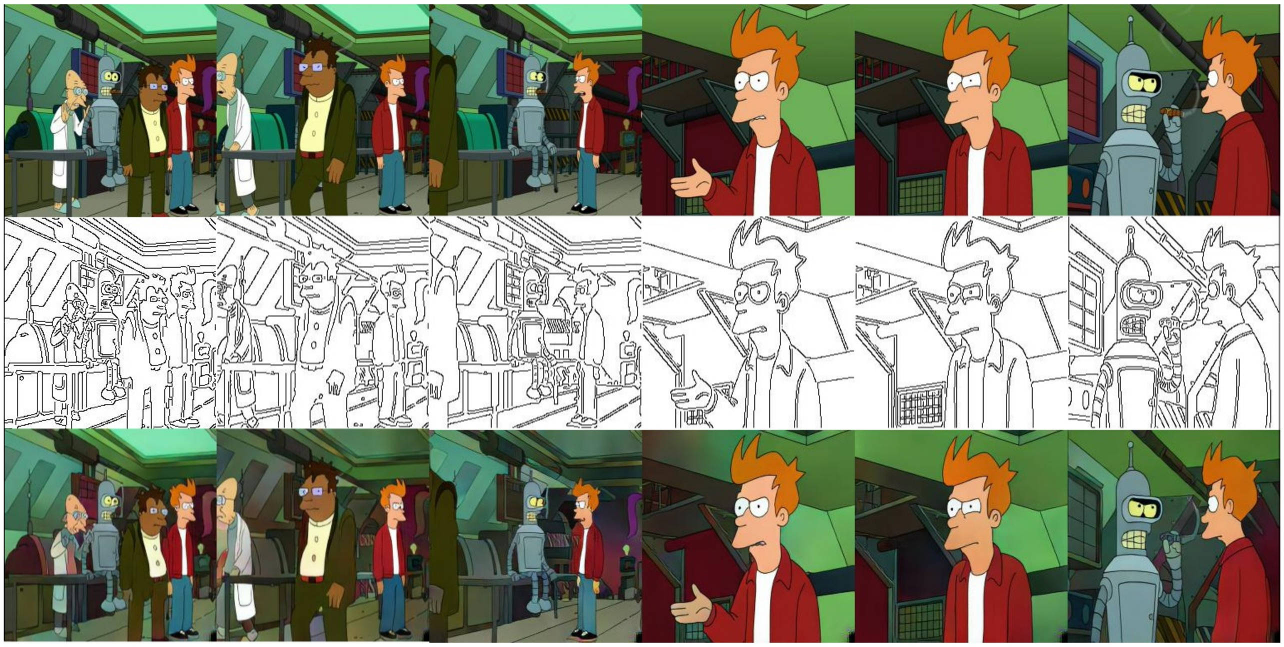

Figure 2. (Top) Ground truth frame sequence. (Middle) Line art frame sequence using Canny edge detection [3]. (Bottom) Colorized line art output from our proposed color hint and temporally coherent colorization model. The generated frame sequence has the same minor inconsistencies across frames showing that the temporal consistency method works as intended. The colorized output is accurate across different backgrounds.

we calculate the distance between the Gram matrices of the feature maps

$$\mathcal{L}_{style} = \mathbb{E}_j \left[\| G_j^{\phi}(\mathbf{F}_t) - G_j^{\phi}(\mathbf{F}_{gt}) \|_1 \right]. \quad (4)$$

The Gram matrix of feature map ϕ is defined by G_j^{ϕ} in equation (4) which distributes spatial information containing non-localized information such as texture, shape, and style. We also add an l_1 term to our overall loss objective to preserve structure and encourage the generator to produce results similar to the ground truth. We use adversarial loss with l_1 to produce sharper generated outputs. The resulting final loss objective is the following

$$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{cont}\mathcal{L}_{cont} + \lambda_{style}\mathcal{L}_{style} + \lambda l_1 \mathcal{L}_{l_1}. \quad (5)$$

For our experiments λ_{adv} = $\lambda_{content}$ = 1, λ_{style} = 1000 and λ_{l1} = 10. We do not incorporate content loss for greyscale experiments since the content of a greyscale and colored image are the same.

4. Experiments

4.1. Dataset

In order to show viability for large-scale productions, both for Japanese animation and traditional Western animation, we create two datasets by extracting frames from DragonBall and Futurama. We extract frames from seasons 1 and 2 from both shows using OpenCV to create datasets of $100,000$ images for each. To mimic line art we use Canny edge detection [3].

4.2. Training

During training, the first $\mathbf{F}_{t-1}$ is a blank image. In our application scenario, animators can choose patches of color and drag them onto the line art image to have more control over the colorization output of the network. To mimic this during training, we convert the ground truth image into a set of 4×4 patches of the averaged color in the region. One percent of these patches from random locations are shown to the network against a black background as the color hint.

5. Evaluation

Results in Table 1 obtained from an existing baseline model trained on the Dragonball dataset with the previous ground truth frame as a condition in [15], still suffers from the flicker effect due to inconsistencies in color between subsequent frames. In addition, unfamiliar backgrounds and characters suffered the most as the model colored them differently for each frame. On the other hand, when incorporating user-hints and temporal information, we can both achieve higher level of color consistency between related frames as well as color new objects unseen by the network previously with better accuracy. This is also reflected in our quantitative results where our model performs better than

Statistic	Model							
	DragonBall (Baseline)		DragonBall (Ours)		Futurama (Baseline)		Futurama (Ours)	
	Greyscale	Line art	Greyscale	Line art	Greyscale	Lineart	Greyscale	Line art
FID	9.29	19.1	2.48	8.84	6.14	9.50	3.51	9.46
SSIM	0.78	0.57	0.90	0.73	0.79	0.65	0.87	0.68
PSNR	17.4	16.04	24.9	19.57	20.6	18.1	24.8	19.4

Table 1. Quantitative results Structure Similarity Index (SSIM), Peak Signal to Noise Ratio (PSNR), and Fréchet Inception Distance (FID) between results of a baseline without hints conditioned with $F_{gt}^{(t-1)}$ and our proposed model on Futurama and Dragonball datasets.

the baseline across each metric. The score similarity between our model trained on the Futurama and Dragonball datasets highlight the efficacy of our approach for both Western and Japanese animation styles. Our method makes a more significant difference when colorizing Japanese animation styles where color and shading is more complex versus flat colors and shading used in Western animation.

6. Discussion and Future Work

It is evident from our results that the proposed method is a viable solution to heavy colorization workloads. By incorporating color hints, large-scale production animations can reduce the amount of time that is allocated for colorization, while keeping artists in control over an automation pipeline. This allows for the artist's desired color results when new settings and characters are introduced to a series. By incorporating a temporal condition to our adversarial framework, we encourage the generator to produce temporally coherent frames by classifying generated frame sequences instead of images with our discriminator. We notice that simply conditioning the generator on the ground truth image from the previous time step so that the network learns to copy over colors from the generated image at test time is not as effective. Future work can include a more extensive model to effectively replicate line art during training.

Acknowledgments

This research was supported by the NSERC Discovery Grant for M.E. H.T. is supported by an institutional Undergraduate Student Research Fellowship. We gratefully acknowledge the support of NVIDIA Corporation for the donation of GPUs used in this research.

References

[1] Each episode of 'futurama' takes at least a year to make. https://www.theatlantic.com/entertainment. Accessed: 2019-08-08. 1

[2] Paintschainer. https://paintschainer.preferred.tech/index_en.html. Accessed: 2019-08-08. 2

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, pages 679–698, 1986. 3

[4] C. Chan, S. Ginosar, T. Zhou, and A. Effros. Everybody dance now. *arXiv preprint arXiv:1808.07371*, 2018. 2

[5] L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. In *Proceedings of the IEEE CVPR*, 2015. 2

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE CVPR*, 2015. 2

[8] P. Isola, J.-Y. Zhu, and A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE CVPR*, 2017. 1, 2

[9] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 2

[10] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 2

[11] K. Nazeri, E. Ng, and M. Ebrahimi. Image colorization with generative adversarial networks. In *Proceedings of tenth international conference on Articulated Motion and Deformable Objects (AMDO)*, 2018. 2

[12] A. Odena, J. Buckman, C. Olsson, T. B. Brown, C. Olah, C. Raffel, and I. Goodfellow. Is generator conditioning causally related to gan performance? In *Proceedings of the 35 International Conference on Machine Learning*, 2018. 2

[13] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 2

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2

[15] H. Thasarathan, K. Nazeri, and M. Ebrahimi. Automatic temporally coherent video colorization. In *Computer and Robot Vision (CRV)*, 2019. 3

[16] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE CVPR*, 2017. 2

[17] R. Zhang, J. Zhu, P. Isola, X. Geng, A. Lin, T. Yu, and A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36. 2