Machine Traditional Animation

Peter Schaldenbrand
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213
pschalde@cs.cmu.edu

Jean Oh
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
jeanoh@nrec.ri.cmu.edu

Abstract

As impressive as style transfer machine learning techniques have become at styling photographs as paintings, it does not match the fidelity of true paint on paper. Traditional animation, as seen in the 2017 film *Loving Vincent* [1], in which every frame is hand painted by an artist, requires an extreme amount of resources and effort. In order to lower the resources necessary for hand animation, we introduce the first fully autonomous painting animation agent. Similar to rotoscoping, a robot arm paints a given video frame by frame. Additionally, we explore the issues with machine learning methods for stroke-based rendering and offer an alternative, deterministic algorithm. Our automated, traditional animation method is the first of its kind, affordable (3-100 times less expensive than other painting robots), and available for use by artists now. Please see `https://github.com/pschaldenbrand/AniPainter` for examples and code.

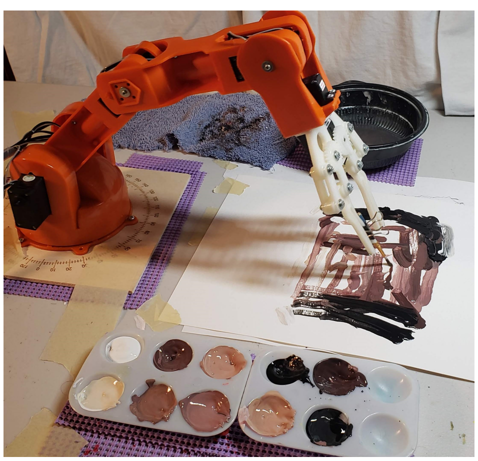

Figure 1: We use a low-end, TinkerKit Braccio robot arm as a painting agent.

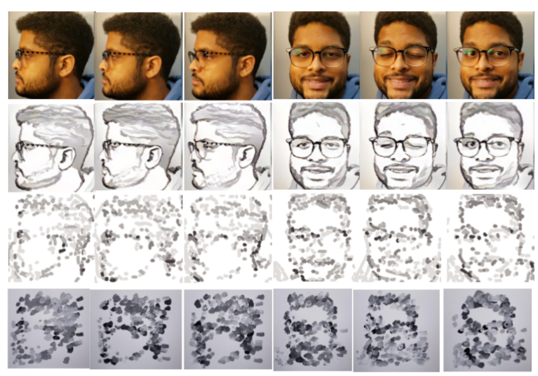

Figure 2: From top to bottom: Input images extracted from a video, the frames stylized using STROTSS [9], computer rendering of the brush stroke instructions, and lastly the robot arm's painting for each stroke instruction sequence.

Motivation Every frame of the 2017 film *Loving Vincent* [1] was hand painted in the style of Vincent van Gogh. The film contained 65,000 frames which took a team of 125 oil painters 6 years to produce. Due to the vast amount of resources necessary, this type of animation is almost completely inaccessible. Our goal was to make an affordable, open-sourced, and automated process for traditional animation.

Related Work Video style transfer provides an approach to simulate painted animation. Most style transfer methods are unable to or require too much computer power to produce high resolution photos

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Figure 3: Painting frames from a computer generated animation.

[9], and the results only resemble paintings and lack distinct brush stroke/painterly qualities [4] despite clever attempts. The second row of Figure 2 shows examples of transferring the content of a frame from a video to the style of a painting. The results resemble the style of the painting but lack true painterly qualities.

EbSynth[1] [8] is a recent, successful video styling tool with which users paint one key frame from a video segment and it applies the style of the painting to the rest of the frames. The key frame painted must exactly match the content in the video's key frame for this method to be successful, which makes EbSynth difficult to use for non-proficient artists. EbSynth can very successfully apply style to the frames in the video, however, the results are still unmistakably computer generated.

Robot painting projects have gained popularity in the past few years. Some of the most precise robot painting machines such as the machine used to produce Rob and Nick Carter's *Dark Factory Portraits* [3] can cost more than $20,000 US dollars. The least expensive machine used for painting that we could find was the uArm Swift used by Joanne Hastie [5] which costs around $750 USD. In this work, we use a TinkerKit Braccio robot arm which costs roughly $220 USD.

Approach As is the case with rotoscoping, our method paints a given video frame by frame. Video frames can be difficult for a robot to paint since they contain fine details that the robot may not be precise enough to capture. Translating the frames into a style that is more possible to paint can make the animation process more successful and creative. In Figure 2, the frames are styled before painting. We use STROTSS [9] for style transfer. Our video style transfer code is available at (`https://github.com/pschaldenbrand/STROTSS-Video`)

A stroke-based rendering (SBR) algorithm converts each frame into a sequence of brush stroke instructions that are fed to the robot. We experimented with using an existing, successful Deep Reinforcement Learning model [7] for SBR. The results in the virtual environment where the model learns to paint look great, but when the instructions are painted, the results are poor. See appendix Figure 4. Attributing the poor quality to the lack of realism of the virtual painting environment, we alter it to use brush strokes that are more realistic to what our robot can perform. We compare these results to a deterministic SBR algorithm we created, and decided that the deterministic algorithm's results appeared to be more accurate.

The brush stroke instructions produced by the SBR algorithm are fed to the TinkerKit Braccio robot arm. Each frame is painted with up to 12 discrete colors of paint determined by performing K-means clustering on the frame. A human must mix and provide these paints to the robot as seen in Figure 1. The arm then autonomously paints a 20×20cm portion of the paper in front of it.

Conclusion In summary, we propose a robotics approach for rotoscoping that does not require any human-painted frames. Based on a preliminary evaluation, we conclude that the proposed approach has a potential for developing affordable solution for the paint-based rotoscoping of animation. Future work will focus on increasing the accuracy of the painting robot, adding a camera so the robot can assess the actual canvas as it paints, and adding temporal consistency between frames.

[1] `https://ebsynth.com/`

1 Acknowledgment

Thank you to Colin Van 't Veld (`https://www.instagram.com/colindesign`) for the animation used in Figures 3 and 4.

2 Ethical Implications

This work is an attempt to automate a task that has until now been performed by human artists. Our desire with this work is to provide animation tools for artists with minimal resources. We do not anticipate this method surpassing the quality of human artists. We believe that as machine animation improves in quality, our value for human created animation will increase. Viewers will learn to identify the quality differences between machine and human painted art, and they learn to appreciate human art more for its superior quality.

References

[1] *Loving Vincent*. BreakThru Productions and Trademark Films, 2017.

[2] Ardavan Bidgoli, Manuel Ladron De Guevara, Cinnie Hsiung, Jean Oh, and Eunsu Kang. Artistic style in robotic painting; a machine learning approach to learning brushstroke from human artists, 2020.

[3] Rob Carter and Nick Carter. Dark factory portraits, 2020.

[4] X. Gao, Y. Tian, and Z. Qi. Rpd-gan: Learning to draw realistic paintings with generative adversarial network. *IEEE Transactions on Image Processing*, 29:8706–8720, 2020.

[5] Joanne Hastie. Red flowers (floral no.1), 2018.

[6] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98*, 1998.

[7] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. *CoRR*, abs/1903.04411, 2019.

[8] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Stylizing video by example. *ACM Transactions on Graphics*, 38(4), 2019.

[9] Nicholas I. Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10051–10060. Computer Vision Foundation / IEEE, 2019.

3 Appendix

3.1 Stroke-Based Rendering

We experimented with using a deep Reinforcement Learning (DRL) painting model [7] for converting a video frame into a sequence of brush strokes as was performed in [2]. The model learns to paint in a virtual painting environment as seen in Figure 4. As is, the virtual painting environment allows brush strokes to be as large as half of the canvas or as small as one pixel. Additionally, the paint can be of varying opacity. When we paint these strokes using the robot arm, the results are extremely different from the virtual environment. We constrained the environment to use strokes that were of the width of our robot's paint brush and ensured that the paint was opaque. The constraints make the virtual environment much more realistic, but the results are still poor. These constraints increase the amount of time for the DRL model to train substantially. The DRL model is trained on faces in the CelebA dataset, and does not generalize well to images such as the second example in Figure 4.

The DRL model generates strokes that are very out of place but will cover them up with white paint with later strokes in the sequence. You can see examples of these with the DRL Painter results in the second example of Figure 4. Our white paint doesn't completely cover the contents below and the painting becomes messy looking.

We introduce a deterministic SBR algorithm similar to Hertzmann's Painterly [6] in order to provide an alternative to the DRL approach. Our algorithm generates brush strokes that minimize the L_1 distance between the current virtual canvas and the target image. More details and code for this SBR algorithm can be found at `https://github.com/pschaldenbrand/AniPainter`.

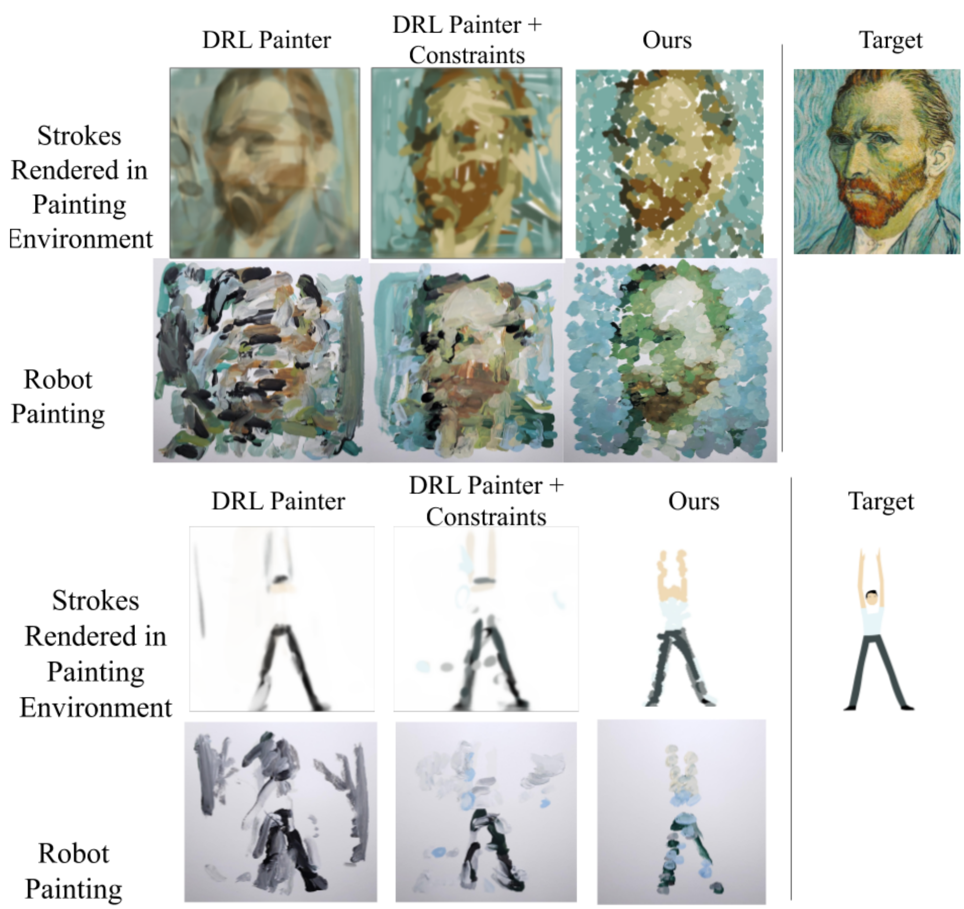

Figure 4: Using a Deep Reinforcement Learning (DRL) model [7] to perform stroke-based rendering versus our deterministic painting algorithm.