Interactive Video Stylization Using Few-Shot Patch-Based Training (Supplementary Material)

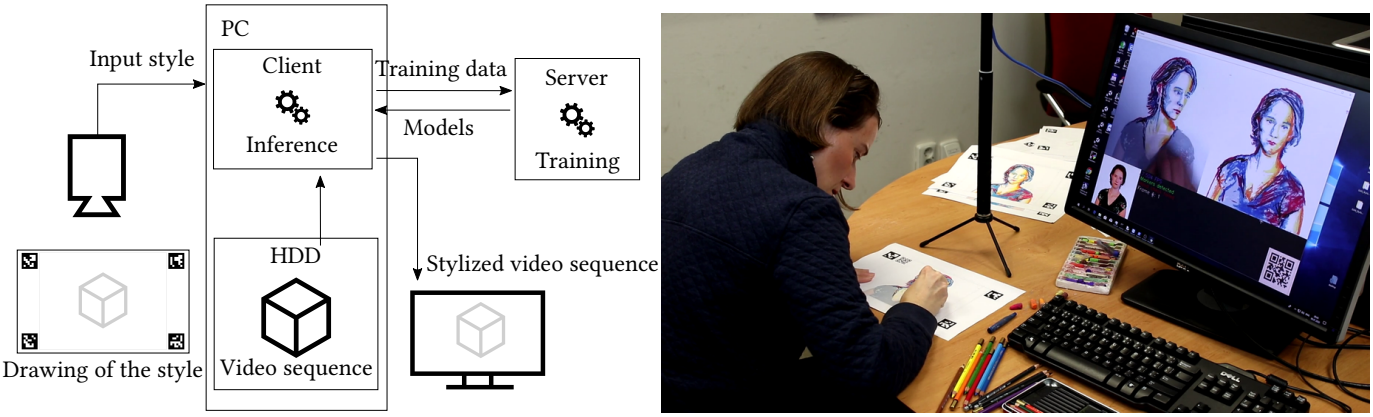

Fig. 1. Scenario No. 1: an artist is drawing over a stencil of a keyframe using traditional media. The stencil contains markers that allow us to perfectly align the frames to prevent shift in images.

1 INTRODUCTION

In this supplementary material we describe the interactive applications of our framework in more detail, presenting the overall architecture of the solution as well as mentioning the specific hardware we used. Furthermore, we show example photos of our framework during real-time stylization sessions with artists (see our supplementary video for live recordings from those sessions) and discuss feedback we received during our informal user study. Lastly, we show additional results produced by our framework, and additional experiments with hyper-parameter setting.

2 INTERACTIVE APPLICATIONS

To demonstrate interactive applications, we provide artists with a setup of our framework in a few variations. Each scenario involves working with a workstation PC, equipped with a consumer-grade GPU (we use Nvidia RTX 2080), on which the artists perform a task. This machine runs our framework executable, which displays visual feedback for the artist. Training of the model is done off-site on a server with an Nvidia Tesla V100 GPU. The client machine sends necessary training data to this server and the training server in turn periodically sends back models trained with the new data. The training data is replaced every time the server receives a new version of a frame. Our training process quickly adapts the model to the new data.

Trained models are used on the artist's PC to generate stylized video frames. Our approach allows us to display an acceptable result in as little as 5 seconds, which improves with time as better models arrive. In practice, the potentially lengthy process of art creation amortizes training time, largely masking the downside of this delay.

Note that inference could also be performed on the server but we do it locally to reduce delay during live-feed stylization.

We devise the following real-time style transfer tasks:

2.1 Pre-recorded video + live style capture (traditional)

The artist is provided with (or creates) a pre-recorded video sequence and selects one or more keyframes which they will paint over. These keyframes are printed in low contrast on a stencil with markers. These markers allow us to perfectly match and align the contents of the stencil with the input sequence frames, so as to avoid misalignment of the training data and achieve the best performance possible. In case of multiple keyframes, we differentiate stencils using additional markers so that the artist is free to swap between them during the session.

As the artist starts painting the first keyframe, the server recognizes which keyframes are ready and only uses previously seen keyframes to train on. Unfinished or unseen parts will likely produce poor visual results which will indicate spots which need to be fixed in current or other keyframes. The artist may also wish to create masks for each keyframe, to prevent introducing ambiguity of different appearances for identical content or to save repetitive work, especially if the keyframes are relatively similar. Diagram for this setup and an example photograph are shown in Fig. 1.

2.2 Live video capture + live style capture (digital)

This scheme is different from the previous in that there is no pre-recorded video sequence, instead, we arrange a camera, capturing a scene in real-time. Our framework allows the artist to export a

Fig. 2. Scenario No. 2: an artist is stylizing an object as seen by the camera in real-time using image editing software.

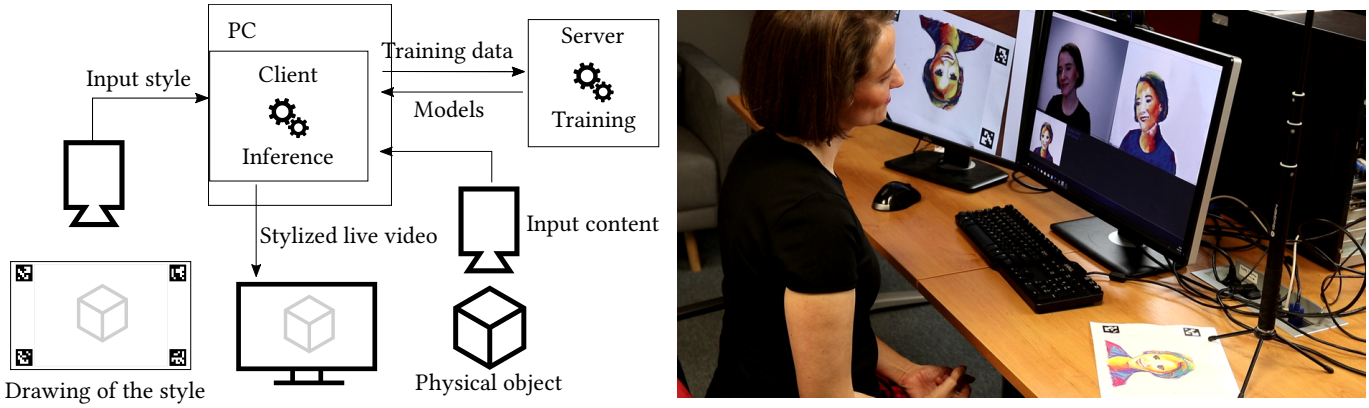

Fig. 3. Scenario No. 3: an artist is stylizing an object as seen by the camera in real-time using a physical stencil.

still image of the scene into image editing software of their choice. This image can then be edited or painted over to achieve an artistic look. Its modified version is periodically sent to the training server, where it serves as the current style exemplar used for training.

During the session, the artist is free to change the scene, while observing the stylization in real-time. If the scene contains some object, a common modification of the scene would be rotating or moving the object. Once the artist is satisfied with the result, they can export additional still images to fix any issues in the scene. This could be, for example, one image for the front of an object and another image for the back of the object. Diagram for this scenario and an example photograph of a session are shown in Fig. 2.

2.3 Live video capture + live style capture (traditional)

We design our framework to also let us combine the two previous scenarios. When a still image of a live scene is exported, it can be printed on a stencil. Artist draws on that stencil and we set up a second camera to capture it. The framework automatically aligns it to the still image and sends it to the training server again. Defining multiple keyframes is then as simple as printing multiple different stencils with identifying markers.

Although working with a digital image is often faster, this setup is useful due to the preference of some artists to work with traditional artistic media. Our framework is well suited for capturing real strokes and stylizing the video frames in a way similar to traditional animation. This scenario is visually explained in Fig. 3.

3 USER STUDY

We asked the artists for their comments on using our framework. Although our user study was informal, we believe it still presents an interesting insight into the contibution of this work.

One of the very first impressions was the moment of surprise and awe whenever a new model arrived on the client machine and a better stylization started appearing on the screen. Thanks to this effect, the artists felt engaged throughout the whole session, some even asked us for further sessions so they could explore the implications of our framework more.

Generally, artists tended to describe the proposed system as a completely new tool to approaching artistic animation, thanks to the real-time feedback and continuous improvement. The other aspect that makes using our framework easy and entertaining, according to the comments, is using the photo stencils, as painting over a photograph using brushes is much easier than creating art from

Fig. 4. The keyframe (a) was used to produce the sequence of 148 frames. While the body part is faithfully represented in both [Jamriška et al. 2019] (b) and ours (c), our approach better preserves the facial region; see the zoom-in views [Jamriška et al. 2019] (d) and ours (e). Video frames (insets of a–c) courtesy of © MAUR film and style exemplar (a) courtesy of © Jakub Javora.

scratch. This also makes it suitable for children, who are largerly familiar with using stencils from coloring books.

Lastly, artists appreciated the fact that no explicit masking needs to be done during the creation process (e.g., background masking). The model we use seems good at representing identity transformation, thus leaving parts of the image unstylized means that the original background just propagates to the output.

While the overwhelming majority of the comments we received were positive, the one negative remark was that the result image quality is somewhat lower than well-optimized sequence created by Jamriška et al. [2019]. However, compared to the inability of their method to deliver such a real-time experience, we feel our framework makes for a reasonable trade-off.

4 ADDITIONAL RESULTS AND EXPERIMENTS

In this section, we first present an additional result of our approach compared to the result of Jamriška et al. [2019], see Fig. 4.

Second, as already primarily covered in the main text, we discuss hyper-parameter optimization on one more example. As it is a common practice to reduce the network size to prevent overfitting, in Fig. 5, we demonstrate that in the task of style transfer, certain network capacity is necessary to achieve high-quality results.

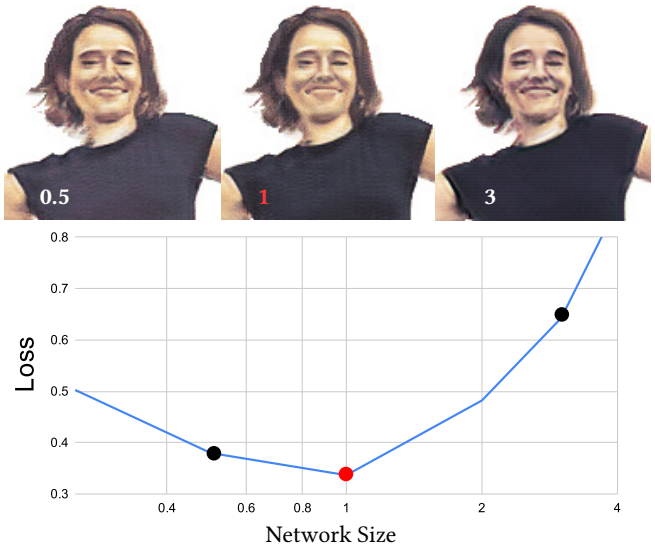

Fig. 5. Impact of network size on the visual quality of results. The loss, y-axes, is computed w.r.t. the output of Jamriška et al. [2019]. The x-axes shows the network size (i.e., number of filters) relative to the best setting we found via hyper-parameter search. Other hyper-parameters are fixed. The middle image (1) depicts the best setting, the left image (0.5) represents setting with half number of filters, and the right image (3) represents setting with three times more filters compared to the middle image. The difference in the visual quality of images, as well as the loss curve, clearly show that there exists a saddle point.

REFERENCES

Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Stylizing Video by Example. *ACM Transactions on Graphics* 38, 4 (2019), 107.