Preserving Global and Local Temporal Consistency for Arbitrary Video Style Transfer

Xinxiao Wu
Beijing Laboratory of Intelligent Information Technology,
School of Computer Science,
Beijing Institute of Technology,
Beijing 100081, China
wuxinxiao@bit.edu.cn

Jialu Chen
Beijing Laboratory of Intelligent Information Technology,
School of Computer Science,
Beijing Institute of Technology,
Beijing 100081, China
chenjialu@bit.edu.cn

ABSTRACT

Video style transfer is a challenging task that requires not only stylizing video frames but also preserving temporal consistency among them. Many existing methods resort to optical flow for maintaining the temporal consistency in stylized videos. However, optical flow is sensitive to occlusions and rapid motions, and its training processing speed is quite slow, which makes it less practical in real-world applications. In this paper, we propose a novel fast method that explores both global and local temporal consistency for video style transfer without estimating optical flow. To preserve the temporal consistency of the entire video (i.e., global consistency), we use structural similarity index instead of flow optical and propose a self-similarity loss to ensure the temporal structure similarity between the stylized video and the source video. Furthermore, to enhance the coherence between adjacent frames (i.e., local consistency), a self-attention mechanism is designed to attend the previous stylized frame for synthesizing the current frame. Extensive experiments demonstrate that our method generally achieves better visual results and runs faster than the state-of-the-art methods, which validates the superiority of simultaneously preserving global and local temporal consistency for video style transfer.

CCS CONCEPTS

• **Information systems** → **Multimedia content creation**; • **Computing methodologies** → *Computer vision.*

KEYWORDS

Video Style Transfer; Self-similarity; Self-attention

ACM Reference Format:
Xinxiao Wu and Jialu Chen. 2020. Preserving Global and Local Temporal Consistency for Arbitrary Video Style Transfer. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413872

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413872

1 INTRODUCTION

Video style transfer has gained growing interests in recent years due to its wide applications in video entertainment, augmented reality, animation synthesis and art creation. Several methods [4, 20] extend style transfer from image to video and process each video frame independently without considering the temporal consistency in video. So the temporal inconsistency can be observed visually as flicker artifacts and discontinuity between consecutive stylized frames.

To address this problem, many methods [1, 19] introduce optical flow to initialize the optimization process and incorporate it into the loss function. Although impressive and smoothing stylized videos are obtained by these methods, their processing speed is quite slow, making them less practical in real-world scenarios. Several recent models [3, 5, 8] improve the speed of video style transfer by using feed-forward networks, and the optical flow is still used in the temporal loss as a guidance to maintain the temporal coherence between consecutive frames. Yet optical flow is sensitive to occlusions and rapid motions, which affects the visual quality of the synthesized videos.

In this paper, we propose a novel fast method that preserves both global and local temporal consistency for video style transfer, which achieves real-time processing speed, nice perceptual style quality, and coherent stylization. To maintain the global consistency over the entire video, a self-similarity loss is proposed to enforce the temporal structure similarity between the stylized video and the source video. The temporal structure describes self-similarity of the entire video, represented by a sequence of structural similarity index (SSIM). To further maintain the local coherence between consecutive frames, we design a self-attention module to learn the temporal dependency between adjacent frames, where the previous stylized frame is attended via attention weights for synthesizing the current frame. By this way, the self-similarity loss and self-attention module replace optical flow as model-free signals to preserve the temporal consistency in both global and local terms.

Our model is constructed by a feed-forward neural network under an encoder-decoder framework, coupled with the proposed self-attention module and a feature transformation module [12]. It is trained using the proposed self-similarity loss combined with a style loss and a content loss to simultaneously guarantee great style perceptual quality and coherent stylized effect.

In summary, the contributions of this work are:

- We propose a simple but efficient method for arbitrary video style transfer that preserves both global and local temporal

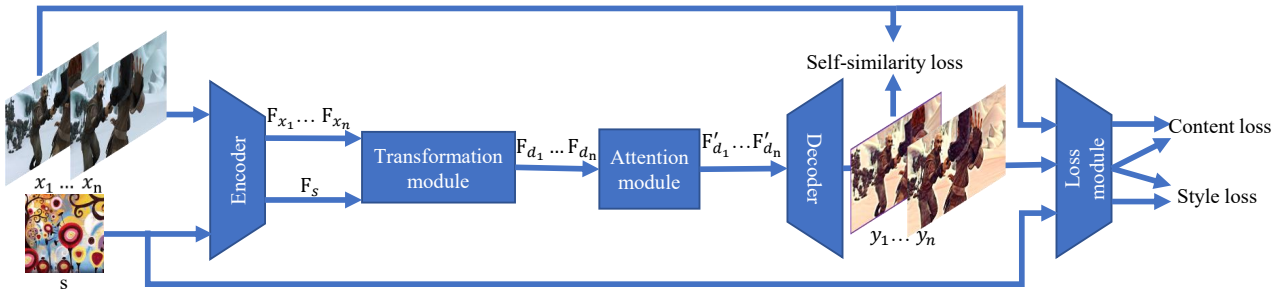

Figure 1: The overview of our proposed method. Our model consists of an encoder-decoder module, a self-similarity module, a transformation module and a loss module. It takes the video frames and a style image as inputs of the encoder, and uses the feature maps of the video frames and the style image as inputs of the transformation module. Then the transformed feature maps of the video frames are fed into the attention module where each transformed feature map and its previous transformed feature map are fused by the attention map. Finally, the attended transformed feature map of each video frame is decoded into a stylized video frame. The self-similarity loss quantifies the difference between the SSIM sequences of the source video and the stylized video. The style loss and the content loss are calculated by the loss module.

consistency of videos, achieving real-time processing speed and nice perceptual style quality.

- We propose a novel self-similarity loss to constrain the temporal structure similarity between the stylized video and the source video. It can effectively suppress the global temporal inconsistency by reducing flicker artifacts and distortions.
- We design a self-attention module to further strengthen the local temporal consistency in adjacent frames by learning the dependency between adjacent frames. It can be readily incorporated into other neural models for video style transfer.
- Experimental results show that our method outperforms the existing methods on both visual effect and proceeding speed.

2 RELATED WORK

2.1 Video Style Transfer

Several previous methods [4, 20] formulate the video style transfer as an extension of image style transfer. Anderson et al. [1] extend [7] to video style transfer by using optical flow to initialize the style transfer optimization and incorporating the flow explicitly into the loss function. To reduce the artifacts at boundaries and occluded regions, Ruder et al. [19] introduce masks to filter out optical flow with low confidences in the loss function. Chen et al. [3] extend [10] to a feed-forward network for video style transfer. It first obtains the current result via a learned flow, and then reduces the artifacts at the occluded regions by fusing the warped result with the independently synthesized result via a learned occlusion mask. Gao et al. [5] propose a feature-map-level temporal loss to penalize variations in the high-level features of the same object in consecutive frames. All these video style transfer methods heavily rely on using optical flow to preserve temporal smoothness and using occlusion mask to stabilize the results. Instead of estimating optical flow, our method uses a self-similarity loss to constrain the structural similarity of the source video and the synthetic video, and a self-attention module to strengthen the temporal consistency in adjacent frames by learning the dependency between adjacent frames, which achieves coherent stylization and real-time processing speed.

2.2 Self-similarity

Structural similarity index (SSIM) [23] is measured from brightness, contrast and structure, which intuitively reflects the structural properties of objects in images. Because of its simple calculation and excellent performance, it is often used as an alternative method of signal-to-noise ratio and mean square error in video compression and reconstruction [22]. SSIM also has been applied to pattern recognition such as image classification [6, 18] where structural relevance of images is expressed by SSIM to learn more accurate image descriptors for recognition. Different from these methods, we make the first attempt to introduce SSIM into video style transfer by using it to represent the temporal self-similarity of the entire video and retain the temporal smoothness in videos.

2.3 Self-attention

Our self-attention module is related to the recent self-attention methods for image generation and machine translation [25]. Several recent works [16, 24] have used it for image style transfer. Park et al. [16] introduce the self-attention to flexibly match the semantically nearest style features onto the content features. Yao et al. [24] incorporate the self-attention into an auto-encoder network to capture the critical characteristics and long-range region relations of the input image. Different from these methods that use the self-attention mechanism to enhance the visual effect of the stylized images, we design a self-attention module to retain the temporal consistency of adjacent frames for video style transfer.

3 METHOD

3.1 Motivation

Directly employing style transfer models to video frames will cause strong flickering and distortions, and affect the visual effects of stylized videos. To tackle this problem, what we intuitively think of

is how to explore the temporal consistency in videos for alleviating the flicker artifacts. Inspired by the fact that the structural similarity index (SSIM) can measure the self-similarity of a video and capture the temporal structure of the entire video, we propose a self-similarity loss based on SSIM to constrain the stylized video to have similar temporal structure as the source video for preserving the global temporal consistency. Furthermore, to maintain the temporal coherence between consecutive frames, we build a self-attention module to learn the temporal dependency between adjacent frames for synthesizing the current frame by attending previous stylized frames.

3.2 Overview

Given a source video $X = \{x_1, \ldots, x_n\}$ and an arbitrary style image S, our goal is to generate a new stylized video $Y = \{y_1, \ldots, y_n\}$, where x_i and y_i represent the i-th source frame and the i-th stylized frame, respectively.

Our video style transfer model is built on an encoder-decoder module coupled with the proposed self-attention module and a transformation module. The encoder-decoder module aims to reconstruct the videos frames faithfully and is fixed when training. The self-attention module learns the temporal dependency between the previous stylized frame and the current synthesized frame, keeping the temporal smoothness and style correspondence between adjacent frames. The transformation module consists of two CNNs, aiming to learn a linear transformation between the content and style information more flexibly and efficiently. To calculate a style loss and a content loss, a loss module is constructed by a pre-trained VGG-19 network [21]. In the training process, our proposed self-similarity loss is combined with the style loss and the content loss to train our network, aiming to enforce the temporal structure similarity between the stylized video and the source video. Figure 1 illustrates the overview of our model.

3.3 Self-attention Module

For each video frame x_i, its feature map F_{x_i} associated with the style image's feature map F_s are given by the encoder module. Then we feed both F_{x_i} and F_s into the transformation module and produce the output feature map F_{d_i}. In the self-attention module, the transformed feature map $F_{d_{i-1}}$ of the previous frame x_{i-1} is attended and combined with the transformed feature map F_{d_i} of the current frame x_i to obtain the updated feature map F'_{d_i}, formulated by

$$F'_{d_i} = SANet(F_{d_{i-1}}, F_{d_i}), \tag{1}$$

where $SANet(\cdot,\cdot)$ represents the self-attention module.

In the self-attention module as shown in Figure 2, we first feed the transformed feature map $F_{d_{i-1}}$ of the previous frame x_{i-1} into one 1×1 convolution to obtain $f(F_{d_{i-1}})$, and feed the transformed feature map F_{d_i} of the current frame x_i into two 1×1 convolutions to obtain $g(F_{d_i})$ and $h(F_{d_i})$, respectively, given by

$$\begin{aligned} f(F_{d_{i-1}}) &= W_f F_{d_{i-1}}, \\ g(F_{d_i}) &= W_g F_{d_i}, \\ h(F_{d_i}) &= W_h F_{d_i}. \end{aligned} \tag{2}$$

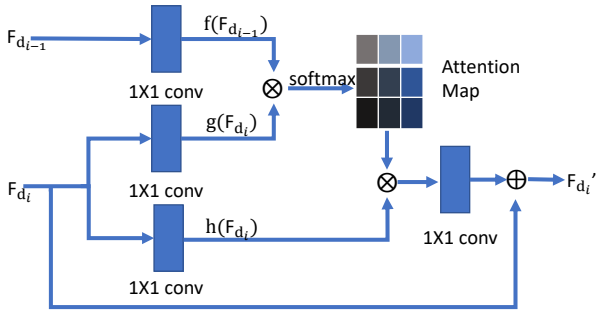

Figure 2: The network architecture of the self-attention module. The inputs are the transformed feature maps of the current video frame and its previous video frame. The output is the attended transformed feature map of the current video frame.

Then the attention map M is calculated by

$$\begin{aligned} m_{k,j} &= \frac{\exp(s_{jk})}{\sum_{j=1}^{N} \exp(s_{jk})}, \\ s_{jk} &= f(F_{d_{i-1}})_j^{\mathrm{T}} g(F_{d_i})_k, \end{aligned} \tag{3}$$

where $m_{k,j}$ represents each element of the attention map M, $f(F_{d_{i-1}})_j$ represents the j-th column of $f(F_{d_{i-1}})$, and $g(F_{d_i})_k$ represents the k-th column of $g(F_{d_i})$. We multiply M with the output $h(F_{d_i})$ of the 1×1 convolution of the transformed feature map F_{d_i} to obtain R_{d_i}:

$$R_{d_i} = M \odot h(F_{d_i}), \tag{4}$$

where $\odot$ denotes the element-wise multiplication operator. Finally, the output F'_{d_i} of the self-attention module is calculated by

$$F'_{d_i} = \delta v(R_{d_i}) + F_{d_i}, \tag{5}$$

where $v(R_{d_i}) = W_v R_{d_i}$ represents 1×1 convolution, and δ is a learnable scalar with an initial value of 0.

In the decoder module, we reconstruct the attended feature map F'_{d_i} to generate the stylized video frame y_i.

3.4 Self-similarity Constraint

In order to maintain the global temporal consistency in videos, we except that the stylized video has similar temporal structure to the source video.

In this work, the temporal structure of a video is represented by the temporal self-similarity over the entire video. We employ structural similarity index (SSIM) to represent the structure similarity of adjacent frames and then use the sequence of SSIM to describe the temporal self-similarity of a video. Accordingly, a novel self-similarity loss is proposed to constrain the stylized video to exhibit similar temporal self-similarity to the source video.

Given two consecutive frames x_{i-1} and x_i, SSIM is defined as

$$SSIM(x_{i-1}, x_i) = \frac{(2\mu_{x_{i-1}}\mu_{x_i} + c_1)(2\sigma_{x_{i-1}x_i} + c_2)}{(\mu_{x_{i-1}}^2 + \mu_{x_i}^2 + c_1)(\sigma_{x_{i-1}}^2 + \sigma_{x_i}^2 + c_1)}, \tag{6}$$

where $\mu_{x_{i-1}}$ and μ_{x_i} represent the means of x_{i-1} and x_i, respectively. $\sigma_{x_{i-1}}$ and σ_{x_i} represent the standard deviations of x_{i-1} and

x_i, respectively. $\sigma_{x_{i-1}x_i}$ is the convariance of x_{i-1} and x_i. c_1 and c_2 denote two constants.

For a source video $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, its self-similarity sequence is represented by $\mathbf{S_X} = [S_1, S_2, ..., S_n]$ where S_i is given by

$$S_i = \begin{cases} 0 & i = 1, \\ SSIM(x_{i-1}, x_i) & \text{otherwise.} \end{cases} \tag{7}$$

In the same way, we can obtain the self-similarity sequence $\mathbf{S_Y}$ of the reconstructed video $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$.

Then the self-similarity loss is defined as the squared L2 norm between the self-similarity sequences $\mathbf{S_X}$ and $\mathbf{S_Y}$, formulated by

$$||\mathbf{S_Y} - \mathbf{S_X}||_2^2. \tag{8}$$

Furthermore, to represent the structure similarity of two temporally distant video frames, SSIM is improved in a long-term way. Let L denote the number of interval frames between the two temporally distant video frames. For example, when $L = 3$, the long-term SSIM is calculated between the frames x_i and x_{i-3}. Thus the long-term self-similarity sequence is represented by $\mathbf{S_{X_L}} = [S_1^L, S_2^L, ..., S_n^L]$ where S_i^L is given by

$$S_i^L = \begin{cases} 0 & i = n - L, \\ SSIM(x_{i-L}, x_i) & \text{otherwise.} \end{cases} \tag{9}$$

In the same way, we can obtain the long-term self-similarity sequence $\mathbf{S_{Y_L}}$ of the reconstructed video $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$. Accordingly, the long-term self-similarity loss can be defined as

$$||\mathbf{S_{Y_L}} - \mathbf{S_{X_L}}||_2^2. \tag{10}$$

Finally, the overall self-similarity loss is given by

$$L_{self-similarity} = ||\mathbf{S_Y} - \mathbf{S_X}||_2^2 + ||\mathbf{S_{Y_L}} - \mathbf{S_{X_L}}||_2^2. \tag{11}$$

3.5 Training and Testing

During the training process, we first feed the first frame of the source video and the style image into the network without the self-attention module and learn the synthetic feature map of the first frame. Then we feed the second frame of the source video and the style image into the network, and the feature maps of the previous frame and the current frame are used as the inputs of the self-attention module. And the same goes for the remaining frames.

Besides the proposed self-similarity loss, we also use a style loss L_{style} and a content loss $L_{content}$ as prior work [7]. Thus the overall loss function L is formulated by:

$$L = L_{content} + \alpha L_{style} + \beta L_{self-similarity}, \tag{12}$$

where α and β are hyper-parameters.

In the testing stage, given an input video $\mathbf{x} = \{x_i\}_{i=1}^n$ and a style image, all of the frames are directly feed into the encoder frame-by-frame with the style image, their corresponding features are then learned by the transformation module, and finally the stylized video $\mathbf{y} = \{y_i\}_{i=1}^n$ is generated via the decoder.

4 EXPERIMENTS

4.1 Datasets

To evaluate the effectiveness of our method, we use the FlyingThing3D and Monkaa datasets [14] as training videos and the MPI Sintel dataset [2] as testing videos. The FlyingThings3D dataset is a large dataset of everyday objects flying along randomized 3D trajectories, which contains around 25,000 frames. The Monkaa dataset is collected from animation short films with about 8,640 frames. The MPI Sintel dataset provides multiple real-world scenarios, which contains 35 videos. The WikiArt dataset [15] is used as the style image dataset, consisting of 11,025 images, and all the test style images are from published implementations [13].

4.2 Experiment Setup

Implementation Details. All the video frames are resized to 256×256. We train the model with a batch size of 2 by 500 iterations and use Adam optimizer[11] with a learning rate of 10^{-7}. The hyper-parameters of α and β are set to $\alpha = 0.02$ and $\beta = 50$, respectively. The number of interval frames in the long-term self-similarity loss is empirically set to $L = 3$. Our model is implemented using PyTorch V0.3 [17] with cuda on a single GTX TITAN X GPU. Code is available at: https://github.com/mcislab-machine-learning/videostyletransfer.

Compared methods. We compare our method with several existing video style transfer methods such as [3, 5, 8, 19] and image style transfer methods such as [9, 12, 13]. We use the model proposed in [12] as our based model.

- ASTV [19] uses optical flow and occlusion mask to preserve the temporal smoothness with an optimization-based method.
- [3] proposes a recurrent model using feature maps of previous frame and consecutive frames as input with optical flow warping in both training and inference stages.
- [8] uses a feed-forward network with a temporal loss to avoid computing optical flow on the fly.
- Reconet [5] incorporates a luminance warping constraint to capture the luminance changes between consecutive frames and increase the stylization stability.
- AdaIN [9] is a classical arbitrary image style transfer method that adjusts the mean and variance of the content input to match those of the style input.
- OST [13] focuses on the theoretical analysis of feature transform and proposes a new closed-form solution.
- [12] learns a transformation matrix that is efficient for arbitrary image style transfer.

4.3 Results

Qualitative Results. Figure 3 shows two examples of two consecutive frames from Alley-1 and Temple-3 videos, respectively, stylized by ASTV and our method with Candy style image, where two local regions for each frame zoom in for detailed demonstration. Since the other video style transfer methods (i.e., [3, 5, 8]) do not release the trained models or the complete training data, we can not reproduce the stylized video frames from these methods for qualitative comparison. From Figure 3, we can have the following observations:

- As shown in Figure 3(b), unexpected color changes appear in some regions of the stylized video frames generated by

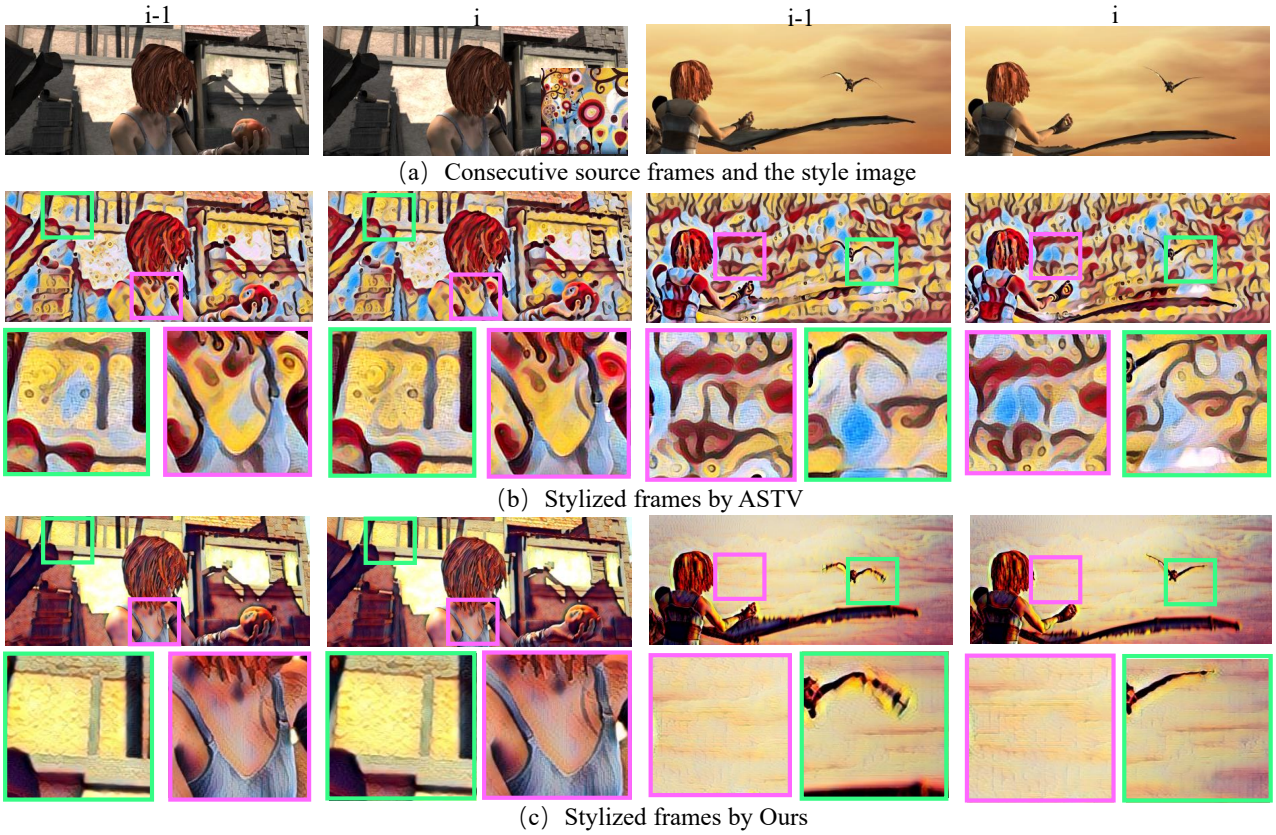

Figure 3: Qualitative comparison results between ASTV and our method. There are two examples of two consecutive frames from Alley-1 and Temple-3 videos, respectively, with Candy style image.

ASTV [19], which causes obvious flickers and distorts the content.

- While the stylized video frames generated by our method performs more consistent and stable, as shown in Figure 3(c), which verifies that our method succeeds in preserving the temporal consistency in both long and short terms.
- Furthermore, the texture in our stylized frames is more authentic, which validates that our method performs better than ASTV in maintaining the detailed content of the source video.

Figure 4 shows two examples of two consecutive frames from CutBunny video that are stylized with Mondrian style image by the image style transfer methods (i.e., AdaIN [9] ,OST [13] and the base model [12]) and our method. The first two columns demonstrate the object motion and the last two columns show the static scene. From Figure 4, it is interesting to observe that

- The synthesized videos by AdaIN and OST have obvious flicker and incoherence obviously in color blocks and object boundary.
- The base model produces unexpected color changes such as the sky and the rabbit that also causes obvious flicker.
- Our method maintains the continuity of video in both object motion and static scene scenarios, which clearly validates the benefits of the self-attention module and the self-similarity loss.

To further demonstrate the difference between the base model [12] and our proposed model, Figure 5 shows two examples of consecutive frames from two different videos (Ambush-5 and Ambush-1) stylized with Candy style image by the base model and our proposed model, where two local regions for each frame zoom in for demonstrations in detail. We can observe that

- Our model captures more detailed texture information. For example, the stone texture looks clearer and more authentic, as shown in the blue and green boxes of Figure 5(c).
- Our model generates more consistent and stable stylized videos. For example, the continuity of the brightness is kept well. While the base model produces unexpected color changes, as shown in red and yellow boxes of Figure 5(c).

All these observations clearly validate that our model not only effectively maintains the temporal consistency, but also retains content details well via the proposed self-similarity loss and self-similarity module.

Quantitative Results. To quantitatively evaluate the temporal consistency captured by the proposed method, we compare the temporal errors of stylized videos between different video style

(a) Consecutive source frames and the style image

(b) Stylized frames by AdaIN

(c) Stylized frames by OST

(d) Stylized frames by the base model

(e) Stylized frames by Ours

Figure 4: Qualitative comparison results between the image style transfer methods (AdaIN, OST and the base model) and our method. There are two examples of two consecutive frames from CutBunny video with Mondrian style image. The left two columns demonstrate the object motion and the right two columns show the static scene.

transfer methods on the MPI Sintel Dataset with Candy style image, as reported in Table 1. The results of all the compared methods are directly copied from their original papers. The temporal error e_{tep_error} is defined by the average pixel-wise Euclidean color difference of consecutive frames [8]:

$$e_{tep\_err} = \sqrt{\frac{1}{(n-1) \times D} \sum_{i=1}^{n-1} M_i ||y_i - W_i(y_{i-1})||^2}, \qquad (13)$$

where n represents the total number of frames, $D = H \times W$ represents the multiplication of height H and width W of the input/output image, M_i is the ground-truth forward occlusion mask, and W_i is the ground-truth forward optical flow. y_{i-1} and y_i denote the stylized previous and current frames, respectively.

Method	Alley-2	Ambush-5	Bandage-2	Market-6	Temple-2
[3]	0.0934	0.1352	0.0715	0.1030	0.1094
Reconet [5]	0.0846	0.0819	0.0662	0.0862	0.0831
[8]	0.0439	0.0675	0.0304	0.0553	0.0513
ASTV[19]	0.0252	0.0512	0.0195	0.0407	0.0361
Ours	0.0205	0.0604	0.0141	0.0853	0.0474

Table 1: Temporal errors of different methods in the testing stage with Candy style image. Five scenes from the MPI Sintel Dataset are selected for evaluation.

From Table 1, we can notice that our method generally achieves lower temporal error than other methods in most cases, which

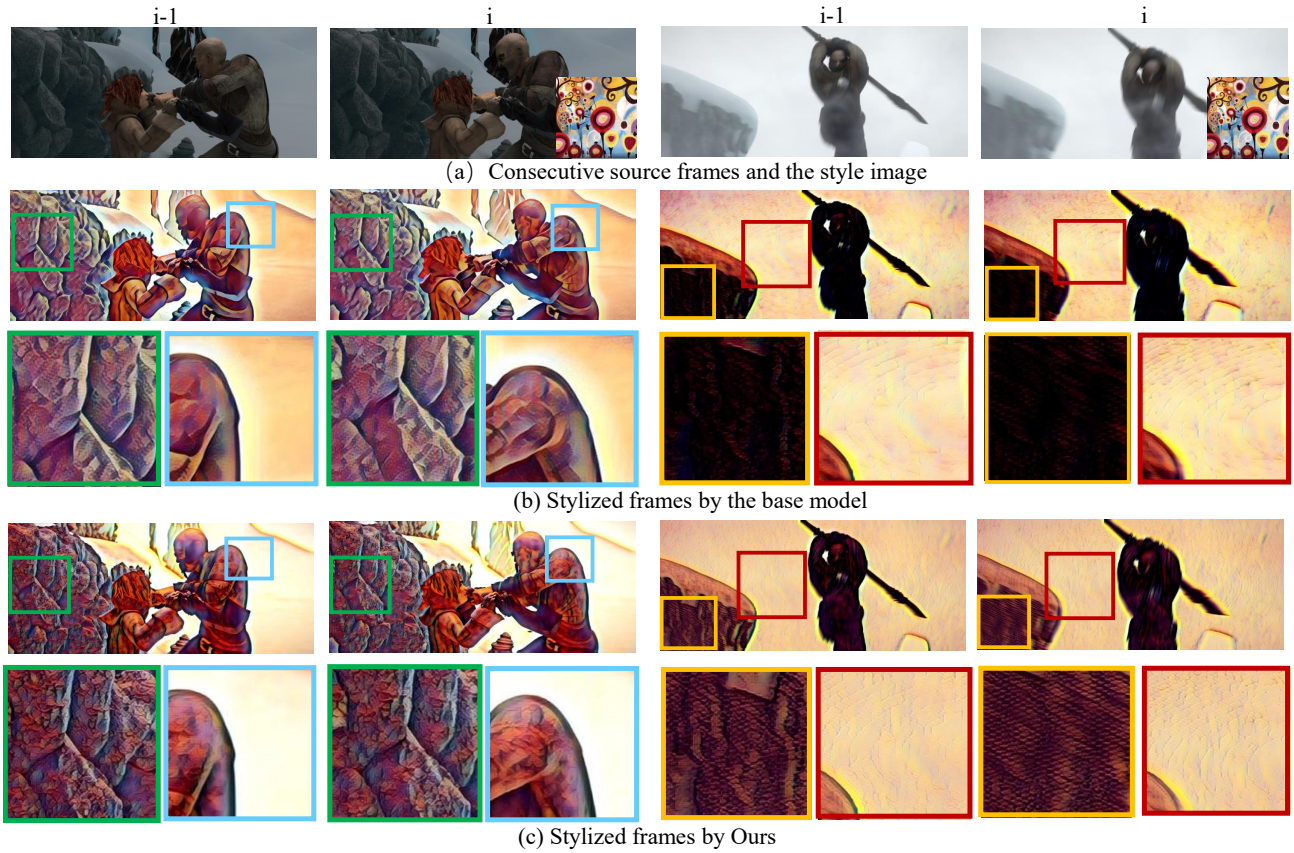

Figure 5: Qualitative comparison results between the base model and our method. There are two examples of consecutive frames from two different videos (Ambush-5 and Ambush-1) with Candy style image.

Figure 6: Ablation studies on the consecutive frames from CutBunny video with Mondrian style image.

validates the effectiveness of replacing optical flow with the self-similarity loss and the self-attention model. Compared with ASTV, our method works worse for some videos such as Market-6 and Temple-2, probably due to that ASTV is an optimized-based method and uses the temporal error as the optimization objective.

Table 2 shows the run time of ASTV and our method using different frame scales: 256×256, 360×640, 436×1024. It is interesting to observe that our model is superior in computation cost and reaches real-time processing speed, owing to the efficient network design and avoidance of optical flow calculation. Although ASTV achieves good results on preserving style imformation and low temporal error (as shown in Table 1), it has very high computation cost, which makes it unfeasible for real-time video style transfer.

Figure 7: Two exemplars of ablation studies on SSIM sequence. (a) Temple-2 video. (b) Shaman-2 video.

Frame Size	256 × 256	360 × 640	436 × 1024
ASTV[19]	0.202	0.096	0.063
Ours	96.15	65.10	26.81

Table 2: Run time of different methods. Run time is measured by the average FPS on a single TITAN X GPU.

Method	Alley-1	Ambush-2	Cave-2	Market-2	Shaman-2
w/o ssim	0.0205	0.1688	0.1119	0.0265	0.0216
w/o attention	0.0215	0.1804	0.1167	0.0253	0.0225
w/o both	0.0217	0.1817	0.1217	0.0267	0.0229
Ours	0.0152	0.1513	0.0374	0.0169	0.0079

Table 3: Ablation studies on temporal error with Candy style image. Five scenes from the MPI Sintel Dataset are selected for evaluation.

4.4 Ablation Studies

We conduct ablation studies by comparing our method with three variants: without the self-similarity loss (denoted as "w/o ssim"), without the self-attention module (denoted as "w/o attention"), and without both total self-similarity loss and self-attention module (denoted as "w/o both"). Figure 6 illustrates two consecutive frames from CutBunny video synthesized with Mondrian style images. We can find that

Method	Alley-1	Market-1	Temple-2	Sleeping-2	Shaman-2
w/o ssim	0.0048	0.0320	0.0156	0.0045	0.0055
w/o attention	0.0051	0.0349	0.0152	0.0043	0.0050
w/o both	0.0059	0.0400	0.0168	0.0049	0.0061
Ours	0.0039	0.0189	0.0113	0.0027	0.0037

Table 4: Ablation studies on SSIM loss with Candy style image. Five scenes from the MPI Sintel Dataset are selected for evaluation.

- The stylized frames of "w/o ssim" are less clear and lost more texture details, demonstrating the importance of the self-similarity on visual effect.
- The stylized frames of "w/o attention" have some obvious flicker artifacts and discontinuity between consecutive stylized frames, demonstrating the effectiveness of the self-attention on preserving the local consistency.
- The performance of "w/o both" substantially degrades, which indicates that both of them are critical to retaining the temporal smoothness for video style transfer.

We futher show two exemplars of ablation study results on SSIM sequence in Figure 7. It is interesting to observe that the SSIM sequence of the stylized video by our method is closet to that of the source video, which clearly demonstrates that our method is able to preserve the global consistency over the entire video.

Table 3 shows the temporal errors of five testing videos with Candy style image of different variants of our method. Besides the temporal error, we also use the self-similarity loss to measure the global temporal consistency of different variants of our method and the results are shown in Table 4. It is obvious that our method achieves smallest temporal error and SSIM loss value for all the videos, further validating the effectiveness of our method on maintaining the global consistency and local consistency simultaneously for video style transfer.

5 CONCLUSIONS

We have presented a novel fast method for arbitrary video style transfer. The proposed self-similarity loss enforces the temporal structure similarity between the stylized video and the source video and thus can preserve the global temporal consistency. The built self-attention module learns the dependency between adjacent frames and is able to maintain the local temporal coherence in videos. Based on a feed-forward network with a transformation module, our method is capable of performing real-time video stylizing with the relief of on-the-fly optical flow computation. Experiment results clearly demonstrate the superiority and efficacy of our method.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant No.61673062.

REFERENCES

[1] Alexander G Anderson, Cory P Berg, Daniel P Mossing, and Bruno A Olshausen. 2016. Deepmovie: Using Optical Flow and Deep Neural Networks to Stylize Movies. *arXiv preprint arXiv:1605.08153*.

[2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *European Conference on Computer Vision*.

[3] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent Online Video Style Transfer. In *Proceedings of the IEEE International Conference on Computer Vision*.

[4] Tian Qi Chen and Mark Schmidt. 2016. Fast Patch-based Style Transfer of Arbitrary Style. In *NIPS Workshop on Constructive Machine Learning*.

[5] Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. 2018. ReCoNet: Real-time Coherent Video Style Transfer Network. In *Asian Conference on Computer Vision*.

[6] Yang Gao, Abdul Rehman, and Zhou Wang. 2011. CW-SSIM Based Image Classification. In *18th IEEE International Conference on Image Processing*.

[7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[8] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. 2017. Real-time Neural Style Transfer for Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[9] Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision*.

[10] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-time Style Transfer and Super-resolution. In *European Conference on Computer Vision*.

[11] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[12] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. 2018. Learning Linear Transformations for Fast Arbitrary Style Transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[13] Ming Lu, Hao Zhao, Anbang Yao, Yurong Chen, Feng Xu, and Li Zhang. 2019. A closed-form solution to universal style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*.

[14] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical flow, and Scene Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[15] K Nichol. 2016. Painter by Numbers, wikiart.

[16] Dae Young Park and Kwang Hee Lee. 2019. Arbitrary Style Transfer With Style-Attentional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic Differentiation in Pytorch.

[18] Abdul Rehman, Yang Gao, Jiheng Wang, and Zhou Wang. 2013. Image Classification Based on Complex Wavelet Structural Similarity. *Signal Processing: Image Communication* 28, 8, 984–992.

[19] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2016. Artistic Style Transfer for Videos. In *German Conference on Pattern Recognition*.

[20] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. 2018. Avatar-net: Multi-scale Zero-shot Style Transfer by Feature Decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[21] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-scale Image Recognition. In *International Conference on Learning Representations*.

[22] Zhou Wang and Alan C Bovik. 2009. Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures. *IEEE Signal Processing Magazine* 26, 1, 98–117.

[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13, 4, 600–612.

[24] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. 2019. Attention-aware Multi-stroke Style Transfer Supplementary Materials. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[25] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.