ReCoNet: Real-time Coherent Video Style Transfer Network

Chang Gao*[1], Derun Gu*[1], Fangjun Zhang*[1], and Yizhou Yu[1,2]

The University of Hong Kong
{u3514174,greatway,u3514241}@connect.hku.hk
Deepwise AI Lab
yizhouy@acm.org

Abstract. Image style transfer models based on convolutional neural networks usually suffer from high temporal inconsistency when applied to videos. Some video style transfer models have been proposed to improve temporal consistency, yet they fail to guarantee fast processing speed, nice perceptual style quality and high temporal consistency at the same time. In this paper, we propose a novel real-time video style transfer model, ReCoNet, which can generate temporally coherent style transfer videos while maintaining favorable perceptual styles. A novel luminance warping constraint is added to the temporal loss at the output level to capture luminance changes between consecutive frames and increase stylization stability under illumination effects. We also propose a novel feature-map-level temporal loss to further enhance temporal consistency on traceable objects. Experimental results indicate that our model exhibits outstanding performance both qualitatively and quantitatively.

Keywords: Video style transfer · Optical flow · Real-time processing.

1 Introduction

As a natural extension of image style transfer, video style transfer has recently gained interests among researchers [4,14,17,27,28,1,6]. Although some image style transfer methods [19,10] have achieved real-time processing speed, i.e. around or above 24 frames per second (FPS), one of the most critical issues in their stylization results is high temporal inconsistency. Temporal inconsistency, or sometimes called incoherence, can be observed visually as flickering between consecutive stylized frames and inconsistent stylization of moving objects [4]. Figure 1(a)(b) demonstrate temporal inconsistency in video style transfer.

To mitigate this effect, optimization methods guided by optical flows and occlusion masks were proposed [1,27]. Although these methods can generate smooth and coherent stylized videos, it generally takes several minutes to process each video frame due to optimization on the fly. Some recent models [4,14,17,28] improved the speed of video style transfer using optical flows and occlusion masks

* Joint first authors.

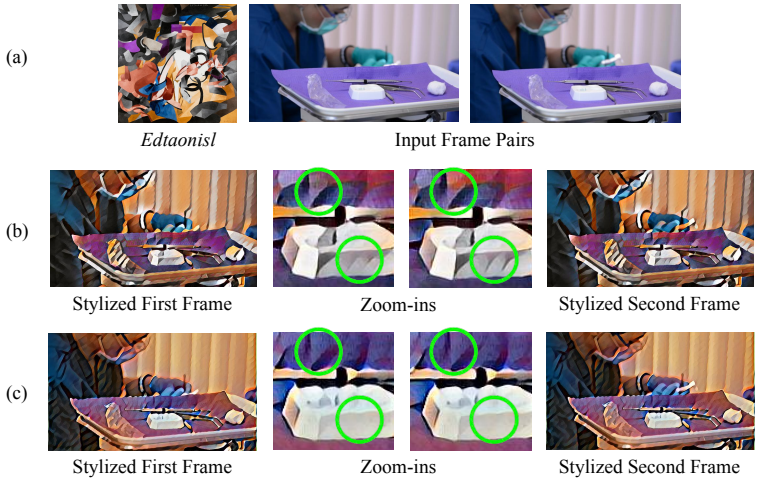

Fig. 1. Temporal inconsistency in video style transfer. (a) The style target *Edtaonisl* (Francis Picabia, 1913) and two consecutive video frames from Videvo.net [34] (b) Style transfer results by Chen *et al* [4] (c) Style transfer results by ReCoNet. The circled regions show that our model can better suppress temporal inconsistency, while Chen *et al*'s model generates inconsistent color and noticeable flickering effects

explicitly or implicitly, yet they failed to guarantee real-time processing speed, nice perceptual style quality, and coherent stylization at the same time.

In this paper, we propose ReCoNet, a real-time coherent video style transfer network as a solution to the aforementioned problem. ReCoNet is a feed-forward neural network which can generate coherent stylized videos with rich artistic strokes and textures in real-time speed. It stylizes videos frame by frame through an encoder and a decoder, and uses a VGG loss network [19,30] to capture the perceptual style of the transfer target. It also incorporates optical flows and occlusion masks as guidance in its temporal loss to maintain temporal consistency between consecutive frames, and the effects can be observed in Figure 1(c). In the inference stage, ReCoNet can run far above the real-time standard on modern GPUs due to its lightweight and feed-forward network design.

We find that the brightness constancy assumption [16] in optical flow estimation may not strictly hold in real-world videos and animations, and there exist luminance differences on traceable pixels between consecutive image frames. Such luminance differences cannot be captured by temporal losses purely based on optical flows. To consider the luminance difference, we further introduce a luminance warping constraint in our temporal loss.

From stylization results of previous methods [4,14,17,27,28], we have also observed instability such as different color appearances of the same moving object in consecutive frames. With the intuition that the same object should possess the same features in high-level feature maps, we apply a feature-map-level temporal loss to our encoder. This further improves temporal consistency of our model.

In summary, there exist the following contributions in our paper:

- Our model highly incorporates perceptual style and temporal consistency in the stylized video. With a new feed-forward network design, it can achieve an inference speed over 200 FPS on a single modern GPU. Our model can reproduce various artistic styles on videos with stable results.
- We first propose a luminance warping constraint in the output-level temporal loss to specifically consider luminance changes of traceable pixels in the input video. This constraint can improve stylizing stability in areas with illumination effects and help suppress overall temporal inconsistency.
- We first propose a feature-map-level temporal loss to penalize variations in high-level features of the same object in consecutive frames. This improves stylizing stability of traceable objects in video scenes.

In this paper, related work for image and video style transfer will be first reviewed in Section 2. Detailed motivations, network architecture, and loss functions will be presented in Section 3. In Section 4, the experiment results will be reported and analyzed, where our model shows outstanding performance.

2 Related Work

Gatys *et al* [11,12] first developed a neural algorithm for automatic image style transfer, which refines a random noise to a stylized image iteratively constrained by a content loss and a style loss. This method inspired many later image style transfer models [19,10,29,3,32,22,13,5,21]. One of the most successful successor is the feed-forward perceptual losses model proposed by Johnson *et al* [19], using a pre-trained VGG network [30] to compute perceptual losses. Although their model has achieved both preferable perceptual quality and near real-time inference speed, severe flickering artifacts can be observed when applying this method frame by frame to videos since temporal stability is not considered. Afterwards, Anderson *et al* [1] and Ruder *et al* [27] introduced a temporal loss function in video stylization as an explicit consideration of temporal consistency. The temporal loss is involved with optical flows and occlusion masks and is iteratively optimized for each frame until the loss converges. However, it generally takes several minutes for their models to process each video frame, which is not applicable for real-time usage. Although Ruder *et al* [28] later accelerated the inference speed, their stylization still runs far below the real-time standard.

To obtain a consistent and fast video style transfer method, some real-time or near real-time models have recently been developed. Chen *et al* [4,6] proposed a recurrent model that uses feature maps of the previous frame in addition to the input consecutive frames, and involves explicit optical flows warping on feature maps in both training and inference stages. Since this model requires optical flow estimation by *FlowNetS* [9] in the inference stage, its inference speed barely reaches real-time level and the temporal consistency is susceptible to errors in optical flow estimation. Gupta *et al* [14] also proposed a recurrent model which takes an additional stylized previous frame as the input. Although their

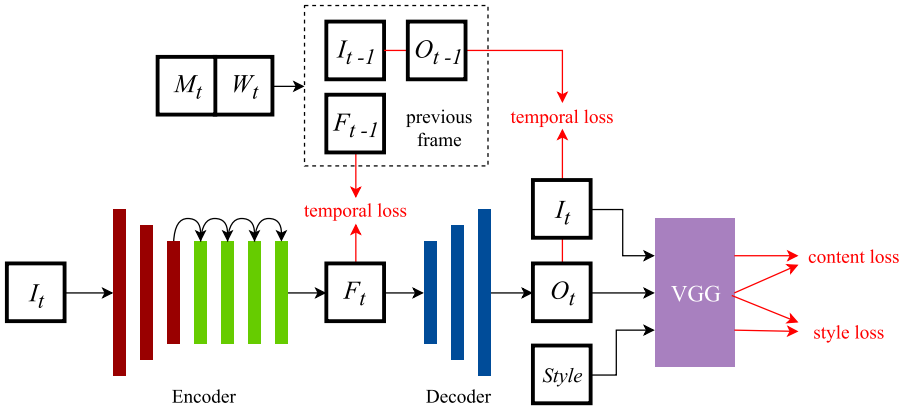

Fig. 2. The pipeline of ReCoNet. I_t, F_t, O_t denote the input image, encoded feature maps, and stylized output image at time frame t. M_t and W_t denote the occlusion mask and the optical flow between time frames $t-1$ and t. *Style* denotes the artistic style image. The dashed box represents the prediction results of the previous frame, which will only be used in the training process. Red arrows and texts denote loss functions

model performs similarly to Chen *et al*'s model in terms of temporal consistency, it suffers from transparency issues and still barely reaches real-time inference speed. Using a feed-forward network design, Huang *et al* [17] proposed a model similar to the perceptual losses model [19] with an additional temporal loss. This model is faster since it neither estimates optical flows nor uses information of previous frames in the inference stage. However, Huang *et al*'s model calculates the content loss from a deeper layer $relu4\_2$, which is hard to capture low-level features. Strokes and textures are also weakened in their stylization results due to a low weight ratio between perceptual losses and the temporal loss.

Noticing strengths and weaknesses of these models, we propose several improvements in ReCoNet. Compared with Chen *et al* [4]'s model, our model does not estimate optical flows but involves ground-truth optical flows only in loss calculation in the training stage. This can avoid optical flow prediction errors and accelerate inference speed. Meanwhile, our model can render style patterns and textures much more conspicuously than Huang *et al* [17]'s model, which could only generate minor visual patterns and strokes besides color adjustment. Our lightweight and feed-forward network can run faster than all video stylization models mentioned above [4,14,17,27,28,1].

3 Method

The training pipeline of ReCoNet is shown in Figure 2. ReCoNet consists of three modules: an encoder that converts input image frames to encoded feature maps, a decoder that generates stylized images from feature maps, and a VGG-16 [30] loss network to compute the perceptual losses. Additionally, a multi-level

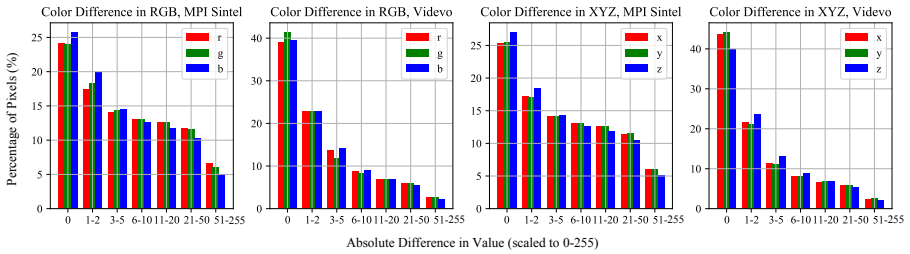

Fig. 3. Histograms of temporal warping error in different datasets and color spaces

temporal loss is added to the output of encoder and the output of decoder to reduce temporal incoherence. In the inference stage, only the encoder and the decoder will be used to stylize videos frame by frame.

3.1 Motivation

Luminance Difference In real-world videos, the luminance and color appearances can be different on the same object in consecutive frames due to illumination effects. In such cases, the data does not satisfy the assumption known as brightness constancy constraint [16], and direct optical flow warping will ignore luminance changes in traceable pixels [9,35,18]. In animations, many datasets use the albedo pass to calculate ground-truth optical flows but later add illuminations including smooth shading and specular reflections to the final image frames, such as MPI Sintel Dataset [2]. This also results in differences on luminance and color appearances.

To further examine the illumination difference, we computed the absolute value of temporal warping error $I_t - W_t(I_{t-1})$ over MPI Sintel Dataset and 50 real-world videos download from Videvo.net [34], where W is the forward optical flow and I is the input image frame. We used *FlowNet2* [18] to calculate optical flows and the method of Sundaram *et al* [31] to obtain occlusion masks for downloaded videos. Figure 3 demonstrates the histograms of temporal warping error in both RGB and XYZ color space. We can draw two conclusions based on the results. First, RGB channels share similar warping error distributions. There is no bias of changes in color channel. Second, despite changes in relative luminance channel Y, the chromaticity channels X and Z in XYZ color space also contribute to the total inter-frame difference. However, since there is no exact guideline of chromaticity mapping in a particular style, we mainly consider luminance difference in our temporal loss.

Based on our findings, we propose a novel luminance constraint in our temporal loss to encourage the stylized frames to have the same luminance changes as the input frames. This can reduce unstable color changes under illumination effects and improve temporal consistency of the stylized frames. Experiments in Section 4.3 show that this new constraint can bring significant improvements to the output perceptual quality and temporal stability.

Feature-map-level Temporal Loss Another new loss function we propose for feature-map-level consistency is based on the intuition that the same object should preserve the same representation in high-level feature maps. Although warping frames directly at the output level may not be accurate due to illuminations, the same method can be very suitable at the feature-map level as examined by Chen *et al* [4]. We use ground-truth optical flows and occlusion masks to calculate feature-map-level temporal loss between the warped feature maps and the current ones. Experiments in Section 4.3 show that this new loss can improve stylization consistency on the same object in consecutive frames.

3.2 Network Architecture

ReCoNet adopts a pure CNN-based design. Compared to feed-forward networks in literature [17,19], we separate the whole network to an encoder and a decoder for different purposes. The encoder is designed to encode image frames to feature maps with aggregated perceptual information, and the feature-map-level temporal loss is computed on its output. The decoder is designed to decode feature maps to a stylized image where we compute the output-level temporal loss. Table 1 shows our encoder and decoder design. There are three convolutional layers and four residual blocks [15] in the encoder, and two up-sampling convolutional layers with a final convolutional layer in the decoder. We use an up-sample layer and a convolutional layer instead of one traditional deconvolutional layer in the decoder to reduce checkerboard artifacts [25]. We adopt instance normalization [33] after each convolution process to attain better stylization quality. Reflection padding is used at each convolutional layer.

The loss network is a VGG-16 network [30] pre-trained on the ImageNet dataset [8]. For each iteration, the VGG-16 network processes each of the input image frame, output image frame and style target independently. The content and style losses are then computed based on the generated image features.

3.3 Loss functions

Our multi-level temporal loss design focuses on temporal coherence at both high-level feature maps and the final stylized output. At the feature-map level, a strict optical flow warping is adopted to achieve temporal consistency of traceable pixels in high-level features. At the output level, an optical flow warping with a luminance constraint is used to simulate both the movements and luminance changes of traceable pixels. The perceptual losses design is inherited from the perceptual losses model [19].

A two-frame synergic training mechanism [17] is used in the training stage. For each iteration, the network generates feature maps and stylized output of the first image frame and the second image frame in two runs. Then, the temporal losses are computed using the feature maps and stylized output of both frames, and the perceptual losses are computed on each frame independently and summed up. Note again that in the inference stage, only one image frame will be processed by the network in a single run.

Table 1. Network layer specification. Layer and output sizes are denoted as channel × height × width. *Conv, Res, InsNorm, ReLU, Tanh* denote convolutional layer, residual block [15], instance normalization layer [33], ReLU activation layer [24], and Tanh activation layer respectively

Layer	Layer Size	Stride	Output Size
Encoder			
Input			$3 \times 640 \times 360$
Conv + InsNorm + ReLU	$48 \times 9 \times 9$	1	$48 \times 640 \times 360$
Conv + InsNorm + ReLU	$96 \times 3 \times 3$	2	$96 \times 320 \times 180$
Conv + InsNorm + ReLU	$192 \times 3 \times 3$	2	$192 \times 160 \times 90$
(Res + InsNorm + ReLU) ×4	$192 \times 3 \times 3$	1	$192 \times 160 \times 90$
Decoder			
Up-sample		1/2	$192 \times 320 \times 180$
Conv + InsNorm + ReLU	$96 \times 3 \times 3$	1	$96 \times 320 \times 180$
Up-sample		1/2	$96 \times 640 \times 360$
Conv + InsNorm + ReLU	$48 \times 3 \times 3$	1	$48 \times 640 \times 360$
Conv + Tanh	$3 \times 9 \times 9$	1	$3 \times 640 \times 360$

Output-level Temporal loss The temporal losses in previous works [4,14,17,27,28] usually ignore changes in luminance of traceable pixels. Taking this issue into account, the *relative luminance* $Y = 0.2126R + 0.7152G + 0.0722B$, same as Y in XYZ color space, is added as a warping constraint for all channels in RGB color space:

$$\mathcal{L}_{temp,o}(t-1,t) = \sum_c \frac{1}{D} M_t \| (O_t - W_t(O_{t-1}))_c - (I_t - W_t(I_{t-1}))_Y \|^2 \qquad (1)$$

where $c \in [R, G, B]$ is each of the RGB channels of the image, Y the relative luminance channel, O_{t-1} and O_t the stylized images for previous and current input frames respectively, I_{t-1} and I_t the previous and current input frames respectively, W_t the ground-truth forward optical flow, M_t the ground-truth forward occlusion mask (1 at traceable pixels or 0 at untraceable pixels), $D = H \times W$ the multiplication of height H and width W of the input/output image. We apply the relative luminance warping constraint to each RGB channel equally based on the "no bias" conclusion in Section 3.1. Section 4.3 further discusses different choices of the luminance constraint and the output-level temporal loss.

Feature-map-level Temporal loss The feature-map-level temporal loss penalizes temporal inconsistency on the encoded feature maps between two consecutive input image frames:

$$\mathcal{L}_{temp,f}(t-1,t) = \frac{1}{D} M_t \| F_t - W_t(F_{t-1}) \|^2 \qquad (2)$$

where F_{t-1} and F_t are the feature maps outputted by the encoder for previous and current input frames respectively, W_t and M_t the ground-truth forward

optical flow and occlusion mask downscaled to the size of feature maps, $D = C \times H \times W$ the multiplication of channel size C, image height H and image width W of the encoded feature maps F. We use downscaled optical flows and occlusion masks to simulate temporal motions in high-level features.

Perceptual Losses We adopt the content loss $\mathcal{L}_{content}(t)$, the style loss $\mathcal{L}_{style}(t)$ and the total variation regularizer $\mathcal{L}_{tv}(t)$ in the perceptual losses model [19] for each time frame t. The content loss and the style loss utilize feature maps at $relu3\_3$ layer and $[relu1\_2, relu2\_2, relu3\_3, relu4\_3]$ layers respectively.

Summary The final loss function for the two-frame synergic training is:

$$\mathcal{L}(t-1,t) = \sum_{i \in \{t-1,t\}} (\alpha \mathcal{L}_{content}(i) + \beta \mathcal{L}_{style}(i) + \gamma \mathcal{L}_{tv}(i)) + \lambda_f \mathcal{L}_{temp,f}(t-1,t) + \lambda_o \mathcal{L}_{temp,o}(t-1,t) \quad (3)$$

where $\alpha, \beta, \gamma, \lambda_f$ and λ_o are hyper-parameters for the training process.

4 Experiments

4.1 Implementation Details

We use Monkaa and FlyingThings3D in the Scene Flow datasets [23] as the training dataset, and MPI Sintel dataset [2] as the testing dataset. The Scene Flow datasets provide optical flows and motion boundaries for each consecutive frames, from which we can also obtain occlusion masks using the method provided by Sundaram *et al* [31]. Monkaa dataset is extracted from the animation movie Monkaa and contains around 8640 frames, resembling MPI Sintel dataset. FlyingThings3D dataset is a large dataset of everyday objects flying along random 3D trajectories and contains around 20150 frames, resembling animated and real-world complex scenes. Same as the verification process of previous works [4,14,17], we use MPI Sintel dataset to verify the temporal consistency and perceptual styles of our stylization results.

All image frames are resized to 640×360. We train the model with a batch size of 2 for 30,000 steps, roughly two epochs over the training dataset. We pair up consecutive frames for the two-frame synergic training and adopt random horizontal flip on each pair. The frame pairs are shuffled in training process. We use Adam optimizer [20] with a learning rate of 10^{-3}, and set the default training hyper-parameters to be $\alpha = 1, \beta = 10, \gamma = 10^{-3}, \lambda_f = 10^7, \lambda_o = 2 \times 10^3$. We implement our style transfer pipeline on PyTorch 0.3 [26] with cuDNN 7 [7]. All tensor calculations are performed on a single GTX 1080 Ti GPU. Further details of the training process can be found in our supplementary materials.

We also download 50 videos from Videvo.net [34] to verify our generalization capacity on videos in real world. Figure 4 shows style transfer results of four

Fig. 4. Video style transfer results using ReCoNet. The first column contains two groups of three consecutive image frames in videos downloaded from Videvo.net [34]. Each video frames are followed by two style target images and their corresponding stylized results of the video frames. The styles are *Mosaic*, *Dream*, *Autoportrait* (Picasso, 1907), and *Candy*

Table 2. Temporal error e_{stab} and average FPS in the inference stage with style *Candy* on different models. Five scenes from MPI Sintel Dataset are selected for validation

Model	Alley-2	Ambush-5	Bandage-2	Market-6	Temple-2	FPS
Chen *et al* [4]	0.0934	0.1352	0.0715	0.1030	0.1094	22.5
ReCoNet	0.0846	0.0819	0.0662	0.0862	0.0831	235.3
Huang *et al* [17]	0.0439	0.0675	0.0304	0.0553	0.0513	216.8
Ruder *et al* [27]	0.0252	0.0512	0.0195	0.0407	0.0361	0.8

different styles on three consecutive video frames. We observe that the color, strokes and textures of the style target can be successfully reproduced by our model, and the stylized frames are visually coherent.

4.2 Comparison to Methods in the Literature

Quantitative Analysis Table 2 shows the temporal error e_{stab} of four video style transfer models on five scenes in MPI Sintel Dataset with style *Candy*. e_{stab} is the square root of output-level temporal error over one whole scene:

$$e_{stab} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\frac{1}{D}M_t\|O_t - W_t(O_{t-1})\|^2} \tag{4}$$

where T is the total number of frames. Other variables are identical to those in the output-level temporal loss. This error function verifies the temporal consistency of traceable pixels in the stylized output. All scene frames are resized to 640×360. We use a single GTX 1080 Ti GPU for computation acceleration.

From the table, we observe that Ruder *et al* [27]'s model is not suitable for real-time usage due to low inference speed, despite its lowest temporal error among all models in our comparison. Among the rest models which reach the real-time standard, our model achieves lower temporal error than Chen *et al* [4]'s model, primarily because of the introduction of the multi-level temporal loss. Although our temporal error is higher than Huang *et al* [17]'s model, our model is capable of capturing strokes and minor textures in the style image while Huang *et al*'s model could not. Please refer to the qualitative analysis below for details.

Another finding is that ReCoNet and Huang *et al*'s model achieve far better inference speed than the others. Compared with recurrent models [4,14,28], feed-forward models are easier to be accelerated with parallelism since the current iteration do not need to wait for the previous frame to be fully processed.

Qualitative Analysis We examine our style transfer results qualitatively with other real-time models proposed by Chen *et al* 's [4] and Huang *et al* 's [17].

Figure 5(a) shows the stylization comparison between Huang *et al* 's model and ReCoNet. Although Huang *et al* 's model achieves low temporal error quantitatively and is able to capture the color information in the style image, it fails

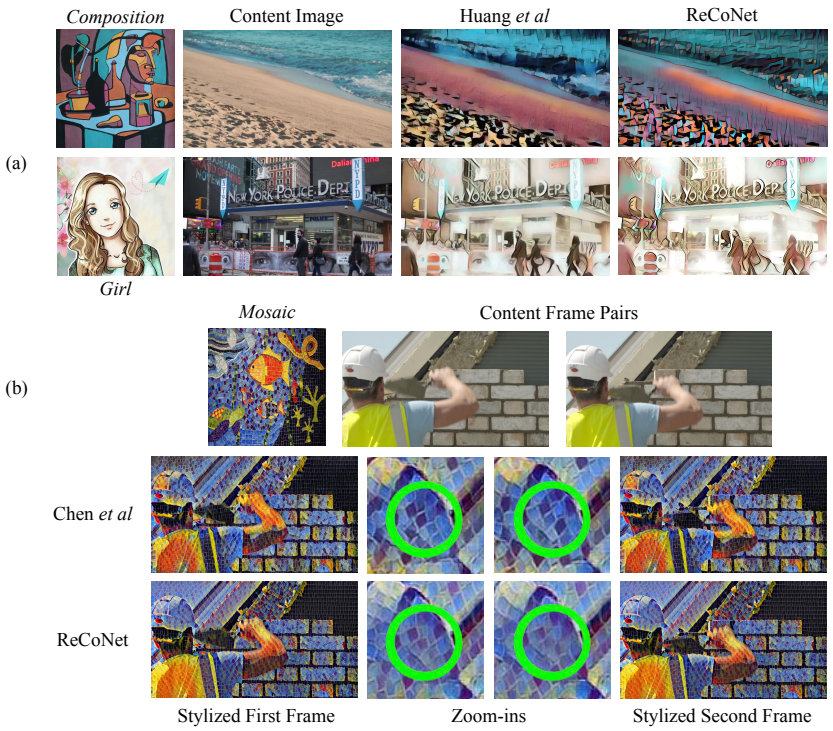

Fig. 5. Qualitative comparison of style transfer results in the literature. (a) Style transfer results between Huang *et al* [17]'s model and ReCoNet on image frames. (b) Style transfer results between Chen *et al* [4]'s model and ReCoNet on consecutive image frames with zoom-ins of flickering regions

to learn much about the perceptual strokes and patterns. There are two reasons that may account for their weak perceptual styles as shown in the two examples in the figure. First, they use a low weight ratio between perceptual losses and temporal loss to maintain temporal coherence, which brings obvious reduction to the quality of output style. However, in ReCoNet, the introduction of the new temporal losses makes it possible to maintain temporal coherence with a larger perceptual to temporal losses ratio, leading to better preserved perceptual styles. As shown in the first example, our stylized image reproduces the distinct color blocks in the *Composition* style much better than Huang *et al*'s result, especially on the uneven sand surfaces and the sea wave. Second, Huang *et al*'s model uses feature maps from a deeper layer $relu4\_2$ in the loss network to calculate the content loss, which is difficult to capture low-level features such as edges. In the second example, although sharp bold contours are characteristic in the *Girl* image, their model fails to clearly reproduce such style. Unlike Huang *et al* 's model, as shown in Figure 1 and 5(b), Chen *et al* 's work can well maintain the perceptual information of both the content image and the style image. How-

Table 3. User study result. In each of the two comparisons, we aggregate the results of all four video clips for the three questions. "Same" means the voter find that results of both models are similar to each other or it is hard to support one against another

Models	Q1	Q2	Q3	Models	Q1	Q2	Q3
ReCoNet	64	162	152	ReCoNet	164	42	115
Chen *et al* [4]	64	15	23	Huang *et al* [17]	22	91	42
Same	72	23	25	Same	14	67	43

Table 4. Temporal error e_{stab} with style *Candy* for different temporal loss settings in ReCoNet. Five scenes from MPI Sintel Dataset are selected for validation

Loss Levels	Alley-2	Ambush-5	Bandage-2	Market-6	Temple-2	Average
Feature-map only	0.1028	0.1041	0.0752	0.1062	0.0991	0.0975
Output only	0.0854	0.0840	0.0672	0.0868	0.0820	0.0813
Both	0.0846	0.0819	0.0662	0.0862	0.0831	0.0804

ever, from zoom-in regions, we can find noticeable inconsistency in their stylized results, which can also be quantitatively validated by its high temporal errors.

To further compare our video style transfer results with these two models, we conducted a user study. For each of the two comparisons (ReCoNet vs Huang *et al* 's and ReCoNet vs Chen *et al* 's), we chose 4 different styles on 4 different video clips downloaded from Videvo.net [34]. We invited 50 people to answer (Q1) which model perceptually resembles the style image more, regarding the color, strokes, textures, and other visual patterns; (Q2) which model is more temporally consistent such as fewer flickering artifacts and consistent color and style of the same object; and (Q3) which model is preferable overall. The voting results are shown in Table 3. Compared with Chen *et al* 's model, our model achieves much better temporal consistency while maintaining good perceptual styles. Compared with Huang *et al* 's model, our results are much better in perceptual styles and the overall feeling although our temporal consistency is slightly worse. This validates our previous qualitative analysis. Detailed procedures and results of the user study can be found in our supplementary materials.

4.3 Ablation Study

Temporal Loss on Different Levels To study whether the multi-level temporal loss does help reduce temporal inconsistency and maintain perceptual style, we implement our video style transfer model on *Candy* style with three different settings: feature-map-level temporal loss only, output-level temporal loss only, and feature-map-level temporal loss plus output-level temporal loss.

Table 4 shows the temporal error e_{stab} of these settings on five scenes in MPI Sintel Dataset. We observe that the temporal error is greatly reduced with

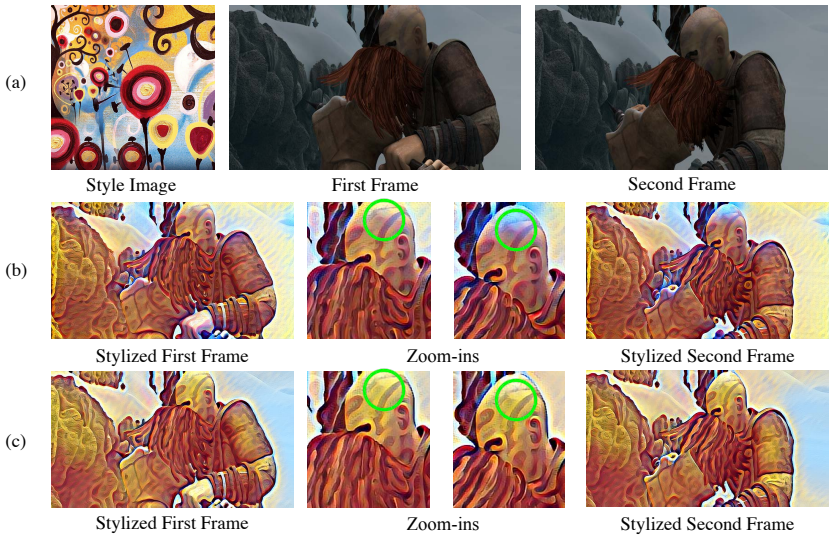

Fig. 6. Temporal inconsistency in traceable objects. (a) The style target and two consecutive frames in MPI Sintel Dataset. (b) Stylized frames generated without feature-map-level temporal loss. (c) Stylized frames generated with feature-map-level temporal loss. A specific traceable region is circled for comparison

the output-level temporal loss, while the feature-map-level temporal loss also improves temporal consistency on average.

Figure 6 demonstrates a visual example of object appearance inconsistency. When only using output-level temporal loss, the exactly same object may alter its color due to the changes of surrounding environment. With the feature-map-level temporal loss, features are preserved for the same object.

Luminance Difference We compare three different approaches taking or not taking luminance difference into consideration at the output level:

1. A relative luminance warping constraint on each RGB channel (Formula 1);
2. Change color space of output-level temporal loss into XYZ color space, then add a relative luminance warping constraint to Y channel: $\mathcal{L}^o_{temp} = \frac{1}{D} M_t(\|(O_t - W_t(O_{t-1}))_Y - (I_t - W_t(I_{t-1}))_Y\|_2 + \|(O_t - W_t(O_{t-1}))_{X,Z}\|_2)$ where X, Y, Z are the XYZ channels;
3. No luminance constraint: $\mathcal{L}^o_{temp} = \frac{1}{D} M_t \|(O_t - W_t(O_{t-1}))_{R,G,B}\|_2$.

Other variables in approach 2 and 3 are identical to those in Formula 1. As shown in Figure 7, all three approaches can obtain pleasant perceptual styles of *Candy* despite some variations in color. However, the first approach has a more similar luminance-wise temporal error map to the input frames compared with the other two methods, especially in the circled illuminated region. This shows

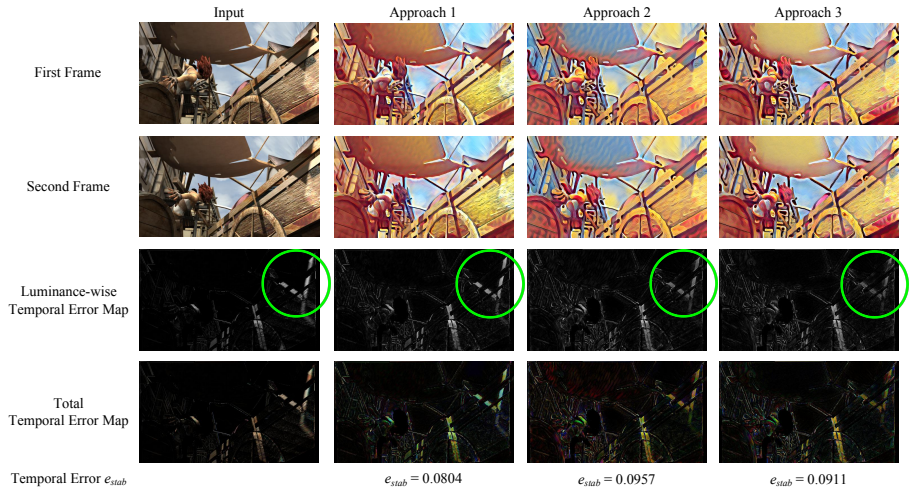

Fig. 7. Style transfer results using three different approaches described in Section 4.3 to target luminance difference. The style target is *Candy*, and the validation scenes are same as Table 4 for temporal error e_{stab} calculation. The total and the luminance-wise temporal error maps show the absolute value of temporal errors in all color channels and in the relative luminance channel respectively

the first approach can preserve proper luminance changes between consecutive frames as those in the input, and therefore leads to more natural stylizing outputs. Moreover, the total temporal error map of the first approach is also closer to zero than the results of other two approaches, implying more stable stylized results. This is also supported numerically by a much lower overall temporal error produced by the first approach in the validation scenes. Based on both qualitative and quantitative analysis, we can conclude that adding a relative luminance warping constraint to all RGB channels can generate smoother color change on areas with illumination effects and achieve better temporal coherence.

5 Conclusions

In this paper, we present a feed-forward convolutional neural network *ReCoNet* for video style transfer. Our model is able to generate coherent stylized videos in real-time processing speed while maintaining artistic styles perceptually similar to the style target. We propose a luminance warping constraint in the output-level temporal loss for better stylization stability under illumination effects. We also introduce a feature-map level temporal loss to further mitigate temporal inconsistency. In future work, we plan to further investigate the possibility of utilizing both chromaticity and luminance difference in inter-frame warping results for better video style transfer algorithm.

References

1. Anderson, A.G., Berg, C.P., Mossing, D.P., Olshausen, B.A.: Deepmovie: Using optical flow and deep neural networks to stylize movies. arXiv preprint arXiv:1605.08153 (2016)
2. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European Conference on Computer Vision. pp. 611–625. Springer (2012)
3. Champandard, A.J.: Semantic style transfer and turning two-bit doodles into fine artworks. arXiv preprint arXiv:1603.01768 (2016)
4. Chen, D., Liao, J., Yuan, L., Yu, N., Hua, G.: Coherent online video style transfer. In: Proceedings of the IEEE International Conference on Computer Vision (Oct 2017)
5. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stylebank: An explicit representation for neural image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1897–1906 (2017)
6. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stereoscopic neural style transfer. CVPR 2018 (2018)
7. Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E.: cudnn: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759 (2014)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009)
9. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2758–2766 (2015)
10. Dumoulin, V., Shlens, J., Kudlur, M., Behboodi, A., Lemic, F., Wolisz, A., Molinaro, M., Hirche, C., Hayashi, M., Bagan, E., et al.: A learned representation for artistic style. arXiv preprint arXiv:1610.07629 (2016)
11. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
12. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2414–2423. IEEE (2016)
13. Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.: Controlling perceptual factors in neural style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
14. Gupta, A., Johnson, J., Alahi, A., Fei-Fei, L.: Characterizing and improving stability in neural style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4067–4076 (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Horn, B.K.: Determining lightness from an image. Computer Graphics and Image Processing **3**(4), 277–299 (1974)
17. Huang, H., Wang, H., Luo, W., Ma, L., Jiang, W., Zhu, X., Li, Z., Liu, W.: Real-time neural style transfer for videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)

18. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2 (2017)
19. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision. pp. 694–711. Springer (2016)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
21. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088 (2017)
22. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6997–7005. IEEE (2017)
23. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: 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. pp. 4040–4048 (2016)
24. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
25. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003, `http://distill.pub/2016/deconv-checkerboard`
26. Paszke, A., Chintala, S., Collobert, R., Kavukcuoglu, K., Farabet, C., Bengio, S., Melvin, I., Weston, J., Mariethoz, J.: Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration (2017)
27. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos. In: German Conference on Pattern Recognition. pp. 26–36. Springer (2016)
28. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos and spherical images. International Journal of Computer Vision pp. 1–21 (2018)
29. Selim, A., Elgharib, M., Doyle, L.: Painting style transfer for head portraits using convolutional neural networks. ACM Transactions on Graphics (ToG) **35**(4), 129 (2016)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by gpu-accelerated large displacement optical flow. In: European Conference on Computer Vision. pp. 438–451. Springer (2010)
32. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: Feed-forward synthesis of textures and stylized images. In: International Conference on Machine Learning. pp. 1349–1357 (2016)
33. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
34. Videvo: Videvo free footage (2018), `https://www.videvo.net/`, [Online at https://www.videvo.net/; accessed 26-February-2018]
35. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Computer Vision (ICCV), 2013 IEEE International Conference on. pp. 1385–1392. IEEE (2013)