Single-Video Temporal Consistency Enhancement With Rolling Guidance

Xiaonan Fang[1][0000−0002−4787−5977] and Song-Hai Zhang[2][0000−0003−0460−1586]

[1] Macau University of Science and Technology, Taipa 999078, Macau, China
xnfang@must.edu.mo
[2] Tsinghua University, Beijing 100084, China shz@tsinghua.edu.cn

Abstract. Image/video synthesis has been extensively studied in academics, and computer-generated videos are becoming increasingly popular among the general public. However, ensuring the temporal consistency of generated videos is still a challenging problem. Most existing algorithms for temporal consistency enhancement rely on the motion cues from a guidance video to filter the temporally inconsistent video. This paper proposes a novel approach that processes single-video input to achieve temporal consistency. The key observation is that we can obtain a coarse guidance video through temporal smoothing and refine its visual quality using a rolling guidance pipeline. We only use an off-the-shelf optical-flow estimation model as external visual knowledge. The proposed algorithm has been evaluated on a wide range of videos synthesized by various methods, including single-image processing models and text-to-video models. Our method effectively eliminates temporal inconsistency while preserving the input visual content.

Keywords: temporal consistency · video enhancement · video filtering.

1 Introduction

Video shot represents a realistic or virtual scene in a period. In most scenarios, the shading and reflectance of the scene remain almost unchanged, leading to temporally consistent content in the image domain. Videos captured by cameras or synthesized by realistic rendering algorithms usually have good temporal consistency. However, the rapid development of image processing algorithms and neural synthesis techniques brings new challenges to temporal consistency. Humans are sensitive to flickering effects in the generated videos and usually prefer temporally consistent presentations. Some videos are created from source videos with an image processing algorithm executed frame by frame. These videos often suffer from poor temporal consistency because the adopted algorithms are unstable under the camera and object motion. Some videos are synthesized with more abstract guidance information, such as label maps, edge maps [40, 52, 33] and text description [14, 35, 9], and in these cases, it is more difficult to achieve temporal consistency.

Currently, there are two main-stream strategies to enhance the temporal consistency of synthesized videos. The first one is adding some temporal constraints in a specific synthesis algorithm. For example, when training a neural network, people can add a loss term that requires two pixels corresponding to the same physical location in consecutive frames to have similar color [16]. Another strategy is to apply a post-processing filter to deal with different types of inconsistency brought by various algorithms [4]. The latter category of methods can transfer the inter-frame correspondence from a source video to a target one. Here the target video is generated by some algorithm from the temporally consistent counterpart. However, the source videos are not always available, and some input representations, such as edge maps and textual descriptions cannot provide temporal correspondence. Therefore, it is of great value to develop an algorithm for temporal consistency enhancement with single-video input. We notice that a few recent papers [1, 24] have similar motivations, but our solution is quite different from those and has its advantages. We will give the theoretical and experimental comparison with the method proposed in [24].

The temporally inconsistent video input is denoted as $\mathcal{I} = \{I_1, I_2, \cdots, I_T\}$, where T is the number of frames. The image resolution is $W \times H$. Our goal is to find another sequence $\mathcal{J} = \{J_1, J_2, \cdots, J_T\}$ with the same resolution that maintains the video content and removes as much temporal inconsistency as possible. A straightforward approach to ensure temporal consistency is smoothing the video content temporally. To tackle view changes and object motion, we can adopt an optical-flow estimation method to find pixel correspondence between consecutive frames and apply a 1D filter on each temporal trajectory independently. However, this operation will inevitably smooth every frame in the spatial domain because the flow estimation is imprecise. In addition, the estimated flow becomes less reliable when there exists a flickering effect.

High-quality temporal filtering is possible if there exists an appropriate guidance video. The blind video consistency method (BVC) [4] and Deep Video Prior (DVP) method [25] are two major solutions. BVC uses gradient-domain optimization, while DVP regards the architecture of neural networks as a kind of regularization. Directly using input $\mathcal{I}$ as guidance is unsuitable for these two algorithms. The weights for warping error in [4] are determined by the pixel similarity in the guidance video, so the guidance video must be stable enough. The DVP algorithm requires more properties of guidance video. It must contain the correct structures and textures. Otherwise, the network will produce blurry results. For example, the DVP algorithm does not work when using edge maps as guidance. The three candidate approaches above have their drawbacks, but we will show that the video temporal consistency can be effectively enhanced if they are carefully combined. Besides, it is difficult to resolve the temporal consistency problem in a single-stage neural network such as [22] because of the lack of precise inter-frame correspondences. Thus, we design a multi-stage method to tackle different aspects of challenges progressively.

Our solution brings a few ideas from image filters. Image filters are designed to reduce the spatial variation, while in this task we need to reduce the tempo-

Fig. 1. Example of single-video temporal consistency enhancement. Our algorithm only takes the temporally inconsistent video (left) as input and creates a consistent version (right) with the rolling guidance framework.

ral variation. Image filtering algorithms can utilize structural information from a guidance image [21]. Similarly, previous video temporal filters need a guidance video with sufficient temporal consistency. The problem we encounter is how to construct an appropriate guidance video from unstable input. Inspired by the Rolling Guidance Filter [49], we propose a pipeline that generates a coarse guidance video at first and refines the video content gradually. The pipeline consists of three stages. In the first stage, we apply a temporal version of domain transform filter [10]. The filtered video can provide a more precise optical flow map so that we can apply the filter repeatedly with the refined temporal correspondence. After a few steps of temporal filtering, we obtain a coarse but temporally stable video. Then we recover the image structures using gradient-domain optimization, which is modified from the method proposed in [4]. The filtered video from the previous step can serve as guidance and provide inter-frame correspondence. In the final stage, we refine the global consistency and suppress visual artifacts using Deep Video Prior [25], where we still use the result from the previous stage as the guidance video. An example of temporal consistency enhancement result is presented in Fig. 1. The temporal color inconsistency in the input is introduced by an image operator, and our algorithm can remove it and achieve a visually pleasing result.

We conducted experiments on a wide range of synthesized videos. We tested single image operators including colorization, enhancement, spatial white balancing, and dehazing algorithms. We also evaluated videos generated by the text-to-video model, line art colorization model, and neural shading model. We exhibit that our algorithm can effectively improve temporal consistency while maintaining the original image content. Our algorithm does not require guidance videos, and we do not need to train any new network on external datasets. The model only relies on a relatively reliable optical-flow estimation model [37]. The main contributions of this paper are:

- We propose a novel rolling guidance framework of temporal consistency enhancement with single-video input.
- Our algorithm achieves better temporal consistency and visual quality compared with previous methods on a wide range of videos.

2 Related Work

We briefly review the representative papers for temporally consistent video processing and synthesis. Some algorithms are designed for specific tasks, and others are task-agnostic, which could serve as a post-processing filter for various types of processed videos. We also discuss some relevant papers about spatial filters that inspired us to do this work.

2.1 Temporal Consistency for Specific Tasks

Optical flow is widely used in video synthesis algorithms. The accuracy of flow estimation is significantly improved by neural networks such as FlowNet [8, 17], PWC-Net [36] and RAFT model [37]. The most popular method to enhance temporal consistency is using a warping loss between consecutive frames as regularization during network training. Usually, an optical-flow estimation model is adopted to determine the warping function. The warping loss could be used for video style transfer [16], colorization [23], scene illumination [43] and low-light enhancement [47]. Besides, test-time training with geometric constraints was proposed to improve the consistency of depth-map estimation [28].

There are some other strategies to ensure temporal consistency in video synthesis. TecoGAN [7] predicts the video sequence with a forward pass and a backward pass, then a ping-pong loss is used to ensure long-range temporal consistency. This network is trained for video super-resolution and video translation. Vid2vid [40] is built upon the Pix2pix model [18, 41]. It fuses the image warped from previous predictions and a synthesized image with a predicted occlusion map. This method can be accelerated by spatial compression and frame interpolation technique [52]. Video translation model can also be trained on unpaired dataset [5] using cycle consistency for both reconstructed frames and their flow maps.

2.2 Blind Video Temporal Consistency

A temporally inconsistent video is usually the processing result of a source video. Bonneel et al. [4] provided the first blind video temporal consistency (BVC) algorithm, which is independent of how the target video is generated. They optimize the video content with a gradient term to maintain the contrast of processed video and a weighted warping term between consecutive frames. Yao et al. [46] constructed the warped frame from a few keyframes and used a content compensation method to refine detail structures. They also proposed a new metric for temporal consistency considering the warping error on both the source video and the target video. Lai et al. [22] trained a ConvLSTM network on DAVIS dataset [30] to achieve fast blind video consistency (FBVC). They adopted the perceptual loss [19] and warping loss as supervision. Deep Video Prior (DVP) [25] extended the concept of Deep Image Prior [26]. A neural network (e.g., U-Net [31]) is trained to reconstruct the processed frame from the input frame. Since the model fits only one sequence, it can implicitly transfer the temporal correspondence of input frames to the processed frames, leading to a temporally smooth output. The thought of DVP can also be adapted to specific tasks such as video segmentation [51]. Recently, researchers have been considering how to improve temporal consistency when the source video is unavailable. Lei et al. [24] proposed the blind video deflickering algorithm, which uses a neural filter to improve the flawed neural atlases [20]. Ali et al. [1] proposed another framework for task-agnostic consistency with a novel tri-frame design for stable flow estimation on flickering data.

2.3 Spatial Smoothing Filters and Rolling Guidance

Smoothing filters in the image domain are extensively studied, but we will only introduce some widely-used filters here. Bilateral filter [38] is probably the most famous tool for edge-preserving smoothing. Some methods are formulated as minimizing the data term and some regularization terms such as L_0 gradient norm [44], L_1 gradient norm [3] and Relative Total Variation [45]. Domain transform filter [10] explicitly defines the smoothing operation and is more efficient for computation.

The smoothing process can also be conducted with a guidance image. A classical formulation is the joint bilateral filter [21], and the guided filter [13] is another famous tool. The guidance map can be constructed in a more sophisticated way, deriving other filters such as bilateral texture filter [6]. Rolling Guidance Filter [49] uses a Gaussian filter to generate the initial guidance and refine the guidance with the joint bilateral filter iteratively. The converged guidance image is also the smoothed version of the input image. The idea of rolling guidance is also applied to geometry processing, known as the rolling guidance normal filter [39].

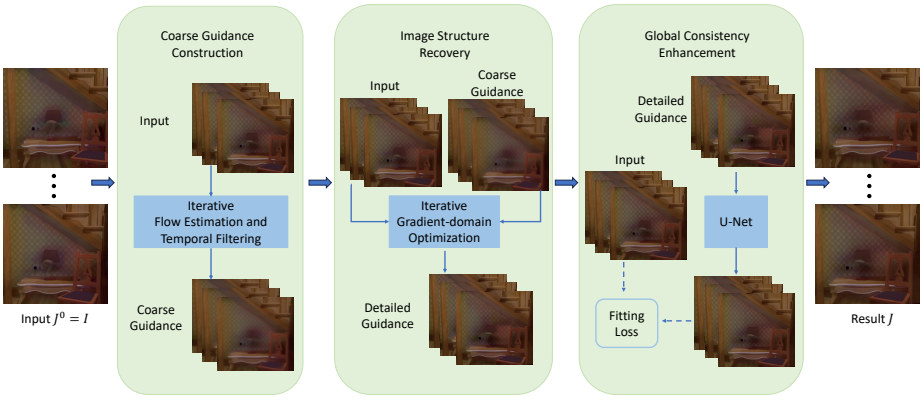

Fig. 2. The pipeline of our algorithm. We adopt the framework of Rolling Guidance Filter [49]. We start with the input video $\mathcal{J}^0 = \mathcal{I}$ and apply a series of operators to filter the input video with guidance iteratively. Specifically, we apply the temporal domain transform filter to get the coarse guidance in the first stage, apply gradient-domain optimization to get detailed guidance in the second stage and enhance the global consistency using the deep-video-prior of a U-Net in the last stage. The final output is denoted by $\mathcal{J}$.

3 Method

3.1 Overview

The pipeline of our temporal consistency enhancement algorithm is illustrated in Fig. 2. The input video is denoted as $\mathcal{I} = \{I_1, I_2, \cdots, I_T\}$. Similar to the Rolling Guidance Filter [49], we compute a series of videos $\mathcal{J}^1, \mathcal{J}^2, \cdots$ step by step. Each video could serve as guidance for the next step. Let $\mathcal{J}^0 = \mathcal{I}$, we can formulate the process as:

$$\mathcal{J}^i = \text{Filter}_i(\mathcal{J}^{i-1}, I), \tag{1}$$

where the function $\text{Filter}_i(\cdot, \cdot)$ represents a joint filter that uses the guidance information inside $\mathcal{J}^{i-1}$ to refine $\mathcal{I}$. However, unlike the pipeline in [49], we adopt three filters in the entire process for different purposes.

Firstly, the input sequence $\mathcal{I}$ is smoothed by a temporal version of domain transform filter [10], resulting in a coarse but temporally stable guidance video $\mathcal{J}^1$. Then it is possible to use $\mathcal{J}^1$ to compute more precise flow maps and reuse the domain transform filter. We repeat this process for s times to obtain $\mathcal{J}^1, \cdots, \mathcal{J}^s$.

Secondly, we try to recover the image structure. In this stage, we adopt a gradient-domain optimization framework modified from BVC [4]. We use $\mathcal{J}^s$ as guidance to filter the target video $\mathcal{I}$ from frame 1 to frame T, obtaining $\mathcal{J}^{s+1}$. Then we regard $\mathcal{J}^{s+1}$ as a new guidance video and apply the optimization algorithm in the opposite direction, obtaining video $\mathcal{J}^{s+2}$ with improving structures.

The final stage is designed to refine $\mathcal{J}^{s+2}$ for global consistency. We use Deep Video Prior [25] to reconstruct $\mathcal{I}$ from $\mathcal{J}^{s+2}$ by training a convolutional neural network from scratch. The final result is denoted by $\mathcal{J}$.

3.2 Constructing Coarse Guidance Video

In this stage, we adopt the domain transform filter [10] to smooth the video content in the temporal order. The basic idea of domain transform is mapping the data points to a line while maintaining the geodesic distance. Given a pixel $p = (x, y)$ in the t-th frame, we can find its corresponding point q in the $(t-1)$-th frame. We adopt the RAFT model [37] for optical flow estimation. The geodesic distance between these two points is defined as:

$$d = 1 + \frac{\sigma_s}{\sigma_r} \|I_t(p) - I_{t-1}(q)\|_1. \tag{2}$$

Parameters σ_s and σ_r represent the variance on the temporal axis and RGB color space. Increasing σ_r will make the result smoother. Since the point q might not lie on the image grid, we cannot efficiently construct the whole trajectory through the video. Therefore, we apply the recursive form of the domain transform filter to smooth the value at p. Since it is not the contribution of this work, please refer to [10] for the detailed implementation. The color at position q is estimated by bilinear interpolation.

Sometimes the content at p in the t-th frame does not appear in the previous frame. One possible situation is that the corresponding position q is outside the image domain, namely $q \notin [0, W-1] \times [0, H-1]$. Another situation is that the point is occluded in the previous frame. We estimate the occlusion by analyzing the optical flow f from the $(t-1)$-th frame to the t-th frame. The visibility map V is constructed as proposed in [42]:

$$\begin{aligned} V(x, y) = \sum_{i=1}^{W} \sum_{j=1}^{H} & \max(0, 1 - |x - i - f^x(i, j)|) \\ & \cdot \max(0, 1 - |y - j - f^y(i, j)|). \end{aligned} \tag{3}$$

Then the binary occlusion map O is defined as:

$$O(p) = \begin{cases} 1, & V(p) > 0.5 \text{ and } q \in [1, H] \times [1, W] \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Let Z_t be the output of the recursive filter in a forward pass, and O_t the occlusion map for the t-th frame, we define

$$Z_t(p) = (1 - O_t(p)a^d)I_t(p) + O_t(p)a^d Z_{t-1}(q). \tag{5}$$

Here the factor a is used to control the amount of local smoothing and is related to σ_s. Similarly, we can filter the video content in the opposite direction. The

video is filtered back and forth with parameter a gradually decreasing. We adopt the same updating rule of a as proposed in [10]. The result of the initial temporal domain transform filter is denoted by $\mathcal{J}^1 = \{J_1^1, J_2^1, \cdots, J_T^1\}$. More accurate optical flow can be estimated on $\mathcal{J}^1$, and we replace the initial flow computed from $\mathcal{I}$. Then we can apply the domain transform filter on $\mathcal{I}$ again, but with new temporal correspondence. We repeat the filtering process for s times. Hence sequence $\mathcal{J}^s$ is the result of this stage. In our experiment, we find that $s = 3$ is adequate to remove most artifacts.

Fig. 3 exhibits the evolution of video content after three iterations. Though the result of the first step (the second row) has serious artifacts, it provides better inter-frame matching (i.e., the artifacts between two frames are also consistent). Thus, the rolling guidance strategy can recover the content gradually.

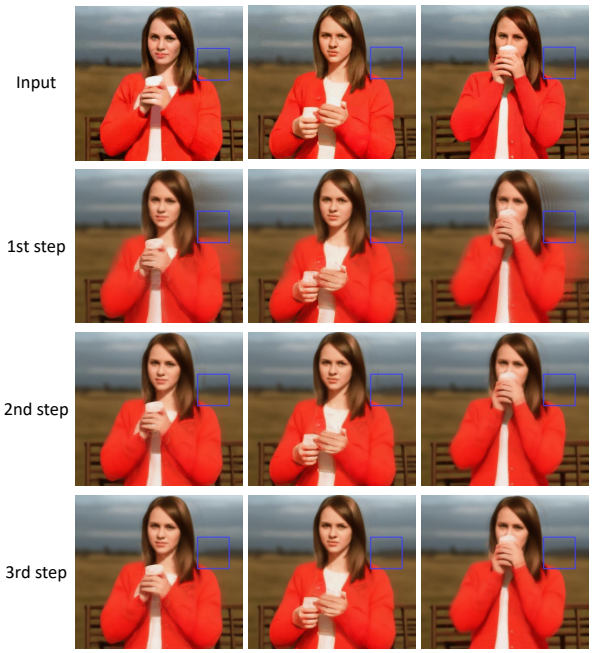

Fig. 3. Example of domain transform filtering in the first stage. The spatial artifact can be significantly reduced using rolling guidance.

3.3 Recovering Image Details

Due to the inherent limitation of optical flow, the result $\mathcal{J}^s$ from the previous stage is usually blurry and might have structural error. The goal of this stage is to recover the clear image structure by optimization. Inspired by the work of

blind video consistency [4], we adopt a modified gradient-domain optimization scheme. The key point is to reconstruct the gradient field of the original frame, namely ∇I_t. Meanwhile, the color at pixel p in the t-th frame is supposed to be similar to a reference point in the neighboring frame. We use $\mathcal{J}^{i-1}$ to represent the result from the previous step. It serves as the guidance video to find the pixel correspondence, and the optimized video is denoted by $\mathcal{J}^i$. For the optimization in a forward pass, our target is to minimize

$$\sum_p \|\nabla J_t^i(p) - \nabla I_t(p)\|^2 + w(p)\|J_t^i(p) - r(p)\|^2. \tag{6}$$

Here $r(p)$ is the reference color for current location p, and weight $w(p)$ is the confidence value of such a reference. The reference could be obtained by warping with the optical flow computed on guidance video $\mathcal{J}^{i-1}$. Let $q_1(p)$ be the corresponding location in the previous frame. The optical-flow estimation is not always correct, so we provide another candidate position $q_2(p)$ in the previous frame found by PatchMatch [2]. Now we have two candidates for reference:

$$r_k(p) = J_{t-1}^i(q_k(p)), k \in \{1, 2\}. \tag{7}$$

Then we define the confidence value as color affinity in the guidance video:

$$w_k(p) = e^{-\|J_t^{i-1}(p) - J_{t-1}^{i-1}(q_k(p))\|^2/2\sigma^2}, k \in \{1, 2\}. \tag{8}$$

The term $w_k(p)$ represents the similarity between position p and $q_k(p)$ in the two consecutive frames. If this value is not high enough or the pixel p is occluded (verified by the value of $O_t(p)$), we consider that the correspondence in the guidance video is inaccurate. Therefore, we compare the similarity with a threshold α. If the similarity is less than α, we will use the color value $I_t(p)$ in the original video as a reference. Specifically, we define

$$r(p) = \begin{cases} r_1(p), w_1(p) \geq w_2(p) \wedge w_1(p) > \alpha \wedge O_t(p), \\ r_2(p), w_2(p) > w_1(p) \wedge w_2(p) > \alpha \wedge O_t(p), \\ I_t(p), \text{ otherwise.} \end{cases} \tag{9}$$

Here the occlusion index $O_t(p)$ is given by Eq. 4. We choose the weight $w(p)$ with the same criteria:

$$w(p) = \begin{cases} w_1(p), w_1(p) \geq w_2(p) \wedge w_1(p) > \alpha \wedge O_t(p), \\ w_2(p), w_2(p) > w_1(p) \wedge w_2(p) > \alpha \wedge O_t(p), \\ \alpha, \text{ otherwise.} \end{cases} \tag{10}$$

We set $\alpha = 0.75$ for all test videos used in the experiment. However, people may choose a lower threshold if the flow estimation is good enough. Moreover, it is possible to use $J_t^{i-1}(p)$ instead of $I_t(p)$ as reference color. Using this alternative leads to results with smoother changes but larger differences from the input video.

The objective in Eq. 6 has a quadratic form and could be converted into a linear system. We initialize $J_1^i = J_1^{i-1}$ and adopt the Gauss–Seidel method to solve $J_2^i, ..., J_T^i$ in order. The result $\mathcal{J}^i$ can serve as the new guidance video for solving $\mathcal{J}^{i+1}$ in the opposite direction, i.e., fixing the last frame and computing the t-th frame from the $(t+1)$-th frame. In this way, we will obtain $\mathcal{J}^{s+2}$ as the output of this stage.

Fig. 4 shows how the detail structures are recovered by gradient-domain optimization. Note that the color of the wall is flickering in the input sequence (first column). The first filtering stage removed the inconsistency but produced blurry textures (second column). The second stage recovered the details while maintaining the color consistency (last column).

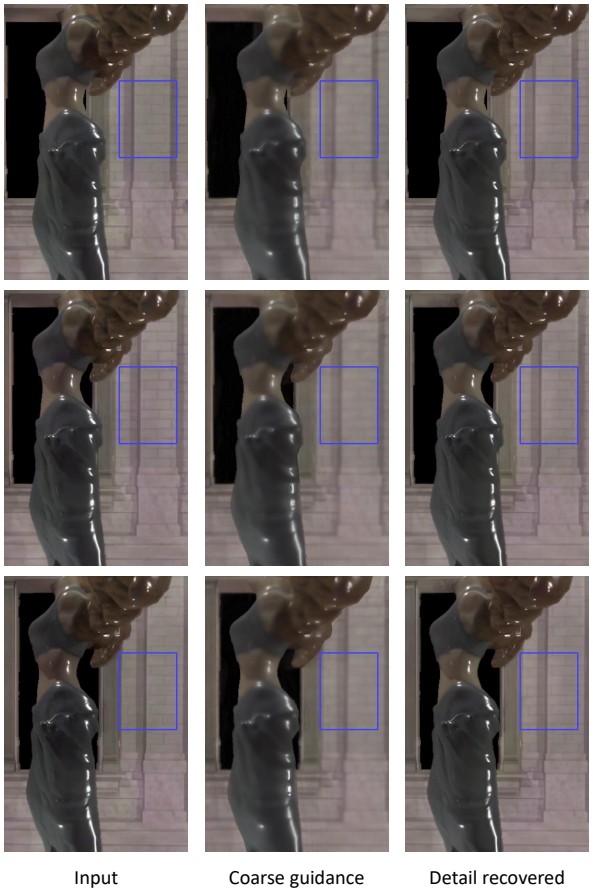

Fig. 4. The effect of image detail recovery via gradient-domain optimization. The coarse guidance video is provided by the previous stage.

3.4 Global Refinement

In the previous stage, the video content is updated sequentially, meaning that the error might accumulate gradually and the later frames will diverge from the original one. Therefore, it is necessary to refine the output video with global optimization. The result of the previous stage will serve as a detailed guidance video to filter the input one. Deep Video Prior [25] has been verified as an effective tool to regularize the visual content and eliminate temporal flickering if a high-quality guidance video is available. A convolutional neural network F is trained from scratch to reconstruct the unstable video $\mathcal{I}$ from some stable input sequence. Each frame is processed independently. In this work, we adopt the U-Net structure [31]. We use the result $\mathcal{J}^{s+2}$ from the second stage as the input of network F. Regarding the video frames as training data, the objective is to minimize the following reconstruction error:

$$\mathcal{L} = \sum_{t=1}^{T} L(F(J_t^{s+2}), I_t). \tag{11}$$

The reconstruction term is defined as the combination of L_1 loss and perceptual loss [19] of VGG-Net features [34]:

$$\begin{aligned} L(F(J_t^{s+2}), I_t) = &\|F(J_t^{s+2}) - I_t\|_1 \\ &+ \sum_{l=1}^{5} \lambda_l \|\phi_l(F(J_t^{s+2})) - \phi_l(I_t)\|_1. \end{aligned} \tag{12}$$

Here $\phi_l(\cdot)$ represents the feature maps in the l-th convolutional block of VGG-Net.

The network is trained through 25 epochs with a learning rate of 10^{-4}. The final enhanced video $\mathcal{J} = \{J_1, J_2, \cdots, J_T\}$ is obtained by applying the trained model F frame by frame.

$$J_t = F(J_t^{s+2}), t \in \{1, 2, \cdots, T\}. \tag{13}$$

3.5 Comparison with the Deflickering Algorithm

The Deflickering algorithm proposed by Lei et al. [24] also aims to improve the temporal consistency using single-video input. Their pipeline also contains three stages, but different techniques are adopted. In both methods, the first stage is designed to obtain a temporally stable intermediate result, but we choose a concise way without tedious training. In the second stage, Lei et al. use a network trained on MS COCO [27] to correct image structures. However, this network is trained on single-image input, so temporal consistency is not explicitly guaranteed. In the last stage, they adopt a network similar to [22]. The advantage is that video can be processed in sequential order by one pass, but the model trained on an external dataset is not as stable as an internal learning method like DVP [25].

4 Experiment

4.1 Dataset

We evaluate our method on two types of data. For the first type, we know the source video from which the target video was generated. We collect the paired test videos from [25], which is generated by colorization [50], dehazing [12], spatial white balancing [15] and enhancement algorithm [11]. We also add a few colorized videos generated by the single image model provided by [23].

For some video generation algorithms, there is no input video, or the input cannot provide sufficient guidance to improve the temporal consistency of the generated video. We collect the text-to-video data from CogVideo [14], Make-A-Video [35] and Gen-2 model [9]. We also consider specific tasks including neural shading [29, 43] and line art colorization [33]. The unpaired dataset contains 31 videos in total.

For most data, we kept the same parameters in the pipeline. We set $\sigma_s = 60$ and $\sigma_r = 1.0$ for stage 1 by default. However, we observe that some videos in the paired dataset have large temporal color variation, so we set $\sigma_s = 300$ and $\sigma_r = 6.0$ to handle these challenging cases.

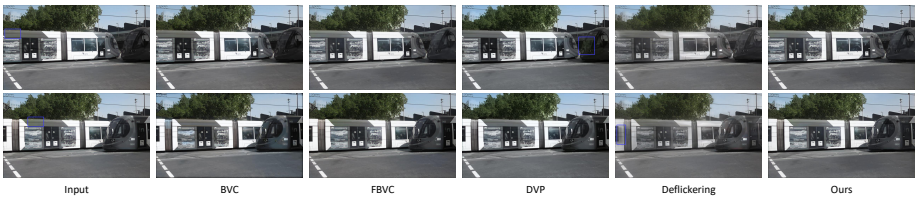

Fig. 5. Comparison on paired data. For the colorization task, the guidance video used in BVC [4], FBVC [22], and DVP [25] is the grayscale version of the input. The Deflickering algorithm [24] and our method do not use the guidance video.

4.2 Quality Assessment

Similar to the DVP paper [25], we assess the quality of refined videos in two aspects: the temporal consistency and the similarity to the input video. We use warping error to measure the temporal consistency. The warping error between two frames J_s, J_t with resolution $W \times H$ is defined as:

$$e(J_s, J_t) = \frac{1}{W \times H} \|M_{s,t}(J_t - \text{warp}(J_s))\|^2. \tag{14}$$

For paired data, the frame J_s is warped by the optical flow computed on the original video. The flow map is predicted by the RAFT model [37], and the

occlusion map is estimated using the method proposed in [32]. Then we construct the temporal consistency measure E_w for video $\mathcal{J}$:

$$E_w(\mathcal{J}) = \frac{1}{2(T-1)} \sum_{t=2}^{T} \left(e(J_{t-1}, J_t) + e(J_1, J_t) \right). \tag{15}$$

The error between consecutive frames reflects the short-range consistency, while the error between the first frame and every other frame represents the long-range consistency.

Apart from temporal consistency, it is also important to maintain the input video content with little appearance change. Thus we define the fidelity term E_f as the average PSNR between input and output frames:

$$E_f(\mathcal{J}, \mathcal{I}) = \frac{1}{T-1} \sum_{i=2}^{T} \mathrm{PSNR}(J_t, I_t). \tag{16}$$

We neglect the first frame because for some methods the first frame is kept the same as the input, and the PSNR value for it is infinity.

As for unpaired videos, there is no temporally stable guidance video for optical-flow estimation. Therefore, we estimate the flow on the output itself as an approximation, and evaluate the following term:

$$\hat{E}_w(\mathcal{J}) = \frac{1}{T-1} \sum_{t=2}^{T} \frac{1}{W \times H} ||J_t - \mathrm{warp}(J_{t-1})||^2. \tag{17}$$

Note that for videos processed by different algorithms, the involved optical flow is also different, and the occlusion map estimated by [32] is not always reliable. So we do not use the occlusion map and the long-range term for unpaired videos. We also use metric E_f to evaluate the fidelity of unpaired videos. Since there is no well-accepted temporal consistency metric for unpaired data, we conduct a user study.

Table 1. Evaluation on Paired Videos. We report the warping error E_w and the fidelity term E_f of different algorithms.

Method	Input	E_w (lower the better)	E_f (higher the better)
Processed	–	0.1877	Inf.
BVC [4]	Paired	0.1513	25.30
FBVC [22]	Paired	0.2692	22.88
DVP [25]	Paired	0.1341	32.25
DeFlickering [24]	Single	0.1160	27.05
Ours	Single	0.1264	30.65

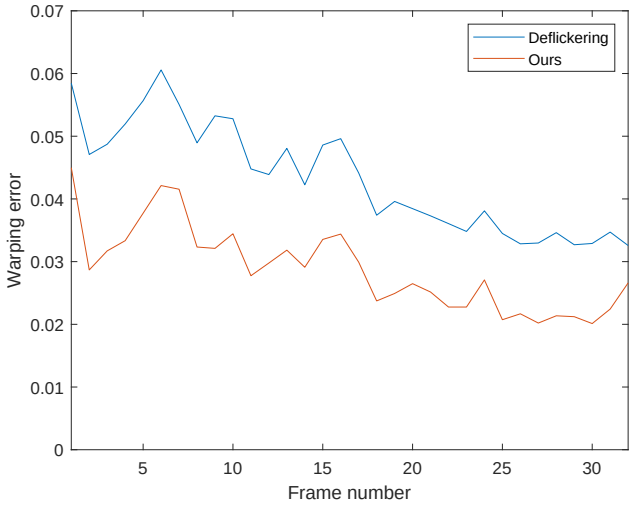

Fig. 6. Frame-by-frame warping error compared to the Deflickering algorithm [24] in one video sequence.

Table 2. Evaluation on Unpaired Videos. We report the warping error with the optical flow computed on the output video, as well as the fidelity term.

Data Type	Processed	Deflickering		Ours	
	$\hat{E}_w$	$\hat{E}_w$	E_f	$\hat{E}_w$	E_f
Make-a-video	0.0624	0.0541	28.90	0.0396	31.45
CogVideo	0.0996	0.0573	31.68	0.0447	30.15
Gen2	0.0487	0.0440	27.75	0.0414	31.31
Shading	0.0304	0.0237	31.71	0.0229	37.36
Colorization	0.0387	0.0297	38.07	0.0249	31.97

4.3 Comparison to State-of-the-Art Methods

For the paired data, we test previous algorithms including BVC [4], FBVC [22], and DVP [25], which require the original video to provide inter-frame correspondence explicitly or implicitly. For the Deflickering algorithm proposed by Lei et al. [24] and our method, the original input video is neglected. The evaluation result is reported in Table 1. The Deflickering algorithm achieved a lower warping error than ours, but our method can better maintain the video content, with a much higher E_f index. All algorithms using paired videos have higher warping errors than ours. We also list the initial errors of the input videos ("Processed" in the table). Fig. 5 shows an example. The Deflickering algorithm produced color-blending artifacts as highlighted in the image.

For the unpaired data, we evaluate our method and the Deflickering algorithm [24]. Table 2 lists the two metrics on each type of video respectively. Due

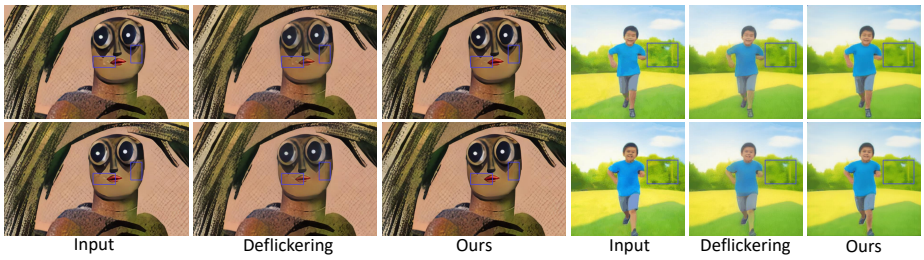

Fig. 7. Comparison between the Deflickering algorithm [24] and our method on unpaired data. Note that the Deflickering algorithm might blend the color among different objects.

Fig. 8. Example of the ablation study. We adopt DVP [25] and an improved version of BVC [4] as components in our pipeline. However, directly applying BVC or DVP using a single video as guidance cannot achieve temporal consistency.

to the large domain difference, the performance on these videos varies a lot. In general, our method is superior to the Deflickering algorithm on $\hat{E}_w$ for all types of videos. The PSNR of our method is similar across different video styles while the Deflickering algorithm is not that stable. We randomly choose 15 videos from the unpaired dataset for the user study. The results generated by our method and Deflickering algorithm were played to users in parallel in random order. Then the users were required to assess the temporal consistency and the general visual quality. We invited 28 users to attend the study and obtained 397 judgments in total. The result is summarized in Table 3. Our method is preferred by more participants on both temporal consistency and visual quality. Fig. 7 exhibits two examples, in which the Deflickering algorithm tends to blend the color of different objects. Compared with the neural filter in [24], our detail recovery scheme can better preserve the original image appearance while improving the temporal consistency. Fig. 6 provides the comparison of warping errors at every frame of one sequence, which is the same one as displayed in Fig. 8. More visual results on long sequences are provided in Fig. 11 and the supplementary material.

Table 3. User Study on Unpaired Videos. Users were required to compare the temporal consistency and general visual quality between our method and the Deflickering algorithm [24]. Then they reported their preferences.

Preference on	Ours	Deflickering	Same
Temporal Consistency	33%	20%	47%
General Visual Quality	45%	21%	34%

Table 4. Results of Ablation Study. We report the warping error and fidelity term.

Method	$\hat{E}_w$	E_f
Processed	0.0996	Inf.
Single-input BVC [4]	0.0861	26.66
Single-input DVP [25]	0.0964	37.13
Ours (stage 1)	0.0258	29.60
Ours (stage 2)	0.0512	31.51
Ours (final)	0.0447	30.15

4.4 Ablation Study

We analyze the intermediate results of our method to verify the effectiveness of our pipeline. In specific, we evaluated the result $\mathcal{J}^s$ of stage 1 and the result $\mathcal{J}^{s+2}$ of stage 2. Since we adopt the BVC method [4] and DVP method [25] as components in our pipeline, we also evaluate these two methods using the input video $\mathcal{I}$ as guidance. We test all these alternatives on the CogVideo dataset [14] containing 13 sequences. The quantitative result is displayed in Tab. 4 and Fig. 8 provides an example. The output of stage 1 is the most consistent under our metric. However, the content is also smoothed in the spatial domain, and some visual artifacts are introduced. The detail recovery process in stage 2 can improve the visual quality and remove most artifacts. The warping error will also increase to some extent. The global optimization in stage 3 can reduce the warping error created by the previous step. Note that using the same video as guidance for BVC [4] or DVP [25] is useless because the inconsistent video cannot provide good visual correspondence.

The first stage aims to obtain a guidance video with reliable optical flow. The definition of the occlusion map in Eq. 4 implies that the forward flow and backward flow should be consistent if the content is not occluded. Therefore, the average value of this map, $\text{Avg}(O)$, can reflect the quality of the estimated flow. Ideally, it should be equal to the actual non-occlusion rate, which is usually close to 1. Fig. 9 shows the change of $\text{Avg}(O)$ for sequences $\mathcal{J}^0$ to $\mathcal{J}^3$ displayed in Fig. 3. A higher value implies that the forward flow and backward flow are more consistent, and hence more reliable. We also visualize the flow maps before and after filtering in Fig. 10. The sequences can be found in the supplementary material.

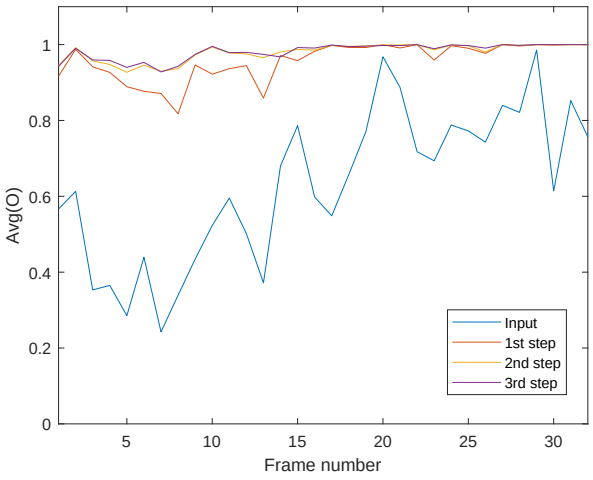

Fig. 9. Average value of the occlusion map O computed from input and intermediate video sequences. A higher value implies a more reliable optical flow estimation.

5 Discussion and Conclusion

Temporal consistency is an important issue in video synthesis. Although AI-generated videos have been widely spread on the Internet, there is no unified framework to ensure temporal consistency on synthesized videos. In this paper, we present a framework to enhance the temporal consistency of a single input video without the guidance of a temporally consistent video. This method can serve as a post-processing operator for a wide range of video synthesis algorithms. We analyze the strengths and drawbacks of existing temporal filters requiring paired input and derive a rolling guidance framework that improves the quality of filtered video with a few iterations. Our pipeline consists of temporal smoothing with a domain transform filter, gradient-domain reconstruction, and global refinement using Deep Video Prior. We evaluate the proposed algorithm with warping error, fidelity term as well as user study, and exhibit that our algorithm can create visually pleasant video content.

Our algorithm cannot handle arbitrary types of temporal inconsistency. For example, Large-scale semantic change in the video is difficult to eliminate (e.g., the frame-by-frame processing result of ControlNet [48]), and we would like to study how to reduce the semantic-level inconsistency in the future. We hope the progress in single-video consistency enhancement can contribute to the whole video synthesis community. If temporal consistency could be achieved by post-processing, the designers of video synthesis models can focus on other aspects of visual quality, such as semantic and aesthetic metrics.

Fig. 10. Visualization of flow maps. The first row shows the input flickering images and the corresponding flow, and the second row shows the result after one-step filtering. It is worth noting that though the initial filtering brings artifacts in the image domain, the updated flow is more accurate and aligned with object boundaries.

References

1. Ali, M.K., Kim, D., Kim, T.H.: Learning task agnostic temporal consistency correction (2022)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (Proc. SIGGRAPH) **28**(3) (Aug 2009)
3. Bi, S., Han, X., Yu, Y.: An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. Acm Transactions on Graphics **34**(4), 78 (2015)
4. Bonneel, N., Tompkin, J., Sunkavalli, K., Sun, D., Paris, S., Pfister, H.: Blind video temporal consistency. ACM Transactions on Graphics (TOG) **34**(6), 1–9 (2015)
5. Chen, Y., Pan, Y., Yao, T., Tian, X., Mei, T.: Mocycle-gan: Unpaired video-to-video translation. In: Proceedings of the 27th ACM International Conference on Multimedia. p. 647–655. MM '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3343031.3350937, https://doi.org/10.1145/3343031.3350937
6. Cho, H., Lee, H., Kang, H., Lee, S.: Bilateral texture filtering. ACM Transactions on Graphics (TOG) **33**(4), 1–8 (2014)
7. Chu, M., Xie, Y., Mayer, J., Leal-Taixé, L., Thuerey, N.: Learning temporal coherence via self-supervision for gan-based video generation. ACM Trans. Graph. **39**(4) (aug 2020). https://doi.org/10.1145/3386569.3392457, https://doi.org/10.1145/3386569.3392457
8. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., Smagt, P.v.d., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2758–2766 (2015). https://doi.org/10.1109/ICCV.2015.316

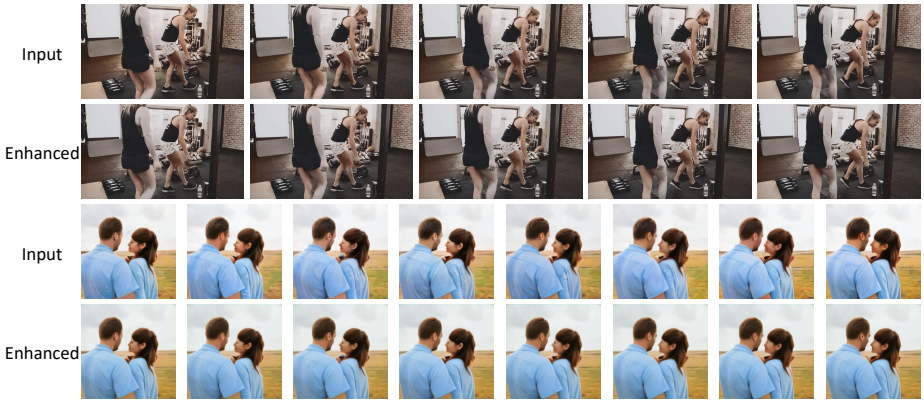

Fig. 11. Additional sequential results of temporal consistency enhancement.

9. Esser, P., Chiu, J., Atighehchian, P., Granskog, J., Germanidis, A.: Structure and content-guided video synthesis with diffusion models. arXiv preprint arXiv:2302.03011 (2023)
10. Gastal, E.S., Oliveira, M.M.: Domain transform for edge-aware image and video processing. In: ACM SIGGRAPH 2011 papers, pp. 1–12 (2011)
11. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. ACM Trans. Graph. **36**(4) (jul 2017). https://doi.org/10.1145/3072959.3073592, https://doi.org/10.1145/3072959.3073592
12. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(12), 2341–2353 (2011). https://doi.org/10.1109/TPAMI.2010.168
13. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1397–1409 (jun 2013). https://doi.org/10.1109/TPAMI.2012.213, https://doi.org/10.1109/TPAMI.2012.213
14. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868 (2022)
15. Hsu, E., Mertens, T., Paris, S., Avidan, S., Durand, F.: Light mixture estimation for spatially varying white balance. In: ACM SIGGRAPH 2008 Papers. SIGGRAPH '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1399504.1360669, https://doi.org/10.1145/1399504.1360669
16. Huang, H., Wang, H., Luo, W., Ma, L., Jiang, W., Zhu, X., Li, Z., Liu, W.: Real-time neural style transfer for videos. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7044–7052 (2017). https://doi.org/10.1109/CVPR.2017.745
17. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017), http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17

18. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 1125–1134 (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. Kasten, Y., Ofri, D., Wang, O., Dekel, T.: Layered neural atlases for consistent video editing. ACM Trans. Graph. **40**(6) (dec 2021). https://doi.org/10.1145/3478513.3480546, https://doi.org/10.1145/3478513.3480546
21. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. In: ACM Transactions on Graphics (ToG). vol. 26, p. 96. ACM (2007)
22. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: European Conference on Computer Vision (2018)
23. Lei, C., Chen, Q.: Fully automatic video colorization with self-regularization and diversity. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
24. Lei, C., Ren, X., Zhang, Z., Chen, Q.: Blind video deflickering by neural filtering with a flawed atlas. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10439–10448 (2023)
25. Lei, C., Xing, Y., Chen, Q.: Blind video temporal consistency via deep video prior. Advances in Neural Information Processing Systems **33**, 1083–1093 (2020)
26. Lempitsky, V., Vedaldi, A., Ulyanov, D.: Deep image prior. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9446–9454 (2018). https://doi.org/10.1109/CVPR.2018.00984
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
28. Luo, X., Huang, J.B., Szeliski, R., Matzen, K., Kopf, J.: Consistent video depth estimation. ACM Transactions on Graphics (ToG) **39**(4), 71–1 (2020)
29. Nalbach, O., Arabadzhiyska, E., Mehta, D., Seidel, H.P., Ritschel, T.: Deep shading: Convolutional neural networks for screen space shading. Comput. Graph. Forum **36**(4), 65–78 (jul 2017). https://doi.org/10.1111/cgf.13225, https://doi.org/10.1111/cgf.13225
30. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 724–732 (2016). https://doi.org/10.1109/CVPR.2016.85
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
32. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos. In: Rosenhahn, B., Andres, B. (eds.) Pattern Recognition. pp. 26–36. Springer International Publishing, Cham (2016)
33. Shi, M., Zhang, J.Q., Chen, S.Y., Gao, L., Lai, Y., Zhang, F.L.: Reference-based deep line art video colorization. IEEE Trans. Vis. Comput. Graph **20**(1) (2022)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)

35. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., Taigman, Y.: Make-a-video: Text-to-video generation without text-video data (2022)
36. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8934–8943 (2018). https://doi.org/10.1109/CVPR.2018.00931
37. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 402–419. Springer International Publishing, Cham (2020)
38. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Computer Vision, 1998. Sixth International Conference on. pp. 839–846. IEEE (1998)
39. Wang, P.S., Fu, X.M., Liu, Y., Tong, X., Liu, S.L., Guo, B.: Rolling guidance normal filter for geometric processing. ACM Transactions on Graphics (TOG) **34**(6), 1–9 (2015)
40. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
41. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
42. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4884–4893 (2018)
43. Xin, H., Zheng, S., Xu, K., Yan, L.Q.: Lightweight bilateral convolutional neural networks for interactive single-bounce diffuse indirect illumination. IEEE Transactions on Visualization and Computer Graphics **28**(4), 1824–1834 (2022). https://doi.org/10.1109/TVCG.2020.3023129
44. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l 0 gradient minimization. Acm Transactions on Graphics **30**(6), 1–12 (2011)
45. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. Acm Transactions on Graphics **31**(6), 1–10 (2012)
46. Yao, C.H., Chang, C.Y., Chien, S.Y.: Occlusion-aware video temporal consistency. In: Proceedings of the 25th ACM International Conference on Multimedia. p. 777–785. MM '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3123266.3123363, https://doi.org/10.1145/3123266.3123363
47. Zhang, F., Li, Y., You, S., Fu, Y.: Learning temporal consistency for low light video enhancement from single images. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4965–4974 (2021). https://doi.org/10.1109/CVPR46437.2021.00493
48. Zhang, L., Agrawala, M.: Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543 (2023)
49. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13. pp. 815–830. Springer (2014)
50. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
51. Zhang, Y., Borse, S., Cai, H., Porikli, F.: Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation. In: 2022

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2633–2642 (2022). https://doi.org/10.1109/WACV51458.2022.00269
52. Zhuo, L., Wang, G., Li, S., Wu, W., Liu, Z.: Fast-vid2vid: Spatial-temporal compression for video-to-video synthesis. In: European Conference on Computer Vision (ECCV) (2022)