Multimedia Tools and Applications (2020) 79:4661–4679
https://doi.org/10.1007/s11042-019-7413-y

Video flickering removal using temporal reconstruction optimization

Chao Li[1] · Zhihua Chen[1] · Bin Sheng[2,3] · Ping Li[4] · Gaoqi He[1]

Received: 30 September 2018 / Revised: 5 January 2019 / Accepted: 22 February 2019 /
Published online: 15 March 2019

Abstract
In this paper, we introduce an approach to remove the flickers in the videos, and the flickers are caused by applying image-based processing methods to original videos frame by frame. First, we propose a multi-frame based video flicker removal method. We utilize multiple temporally corresponding frames to reconstruct the flickering frame. Compared with traditional methods, which reconstruct the flickering frame just from an adjacent frame, reconstruction with multiple temporally corresponding frames reduces the warp inaccuracy. Then, we optimize our video flickering method from following aspects. On the one hand, we detect the flickering frames in the video sequence with temporal consistency metrics, and just reconstructing the flickering frames can accelerate the algorithm greatly. On the other hand, we just choose the previous temporally corresponding frames to reconstruct the output frames. We also accelerate our video flicker removal with GPU. Qualitative experimental results demonstrate the efficiency of our proposed video flicker method. With algorithmic optimization and GPU acceleration, the time complexity of our method also outperforms traditional video temporal coherence methods.

Keywords Video processing · Flickering removal · Multiple frames · Temporal coherence · Spatial coherence

1 Introduction

Video flicker removal refers to removing flickers in videos and recovering temporal coherence between neighboring frames. When we apply traditional image processing methods to videos, temporal coherence between frames will be neglected. As a consequence, flickers occur in the video processed with image-based methods. When viewing one frame in a flickering video separately, the flickers cannot be felt. However, if the video is played consecutively, annoying flickers can be perceived clearly. In addition, since flickers always show as brightness variation, and many applications are always done under the brightness variation condition, it is necessary to remove video flickers and obtain a temporally consistent video. As demonstrated in [22–24], temporal consistency can also be utilized to

✉ Zhihua Chen
czh@ecust.edu.cn

Extended author information available on the last page of the article.

represent activities for automated recognition, so maintaining temporal consistency of a video is of great significance.

Practically speaking, frames in videos may have different degrees of flickering, and it is common to meet brightness variation and tonal variation in consecutive frames. In this way, some non-flickering frames can be used to reconstruct flickering frames. Based on this principle, Farbman and Lischinski [12] proposes to choose several frames as key frames, then other non-key frames will be aligned with the previous selected key frames. Besides this, Bonneel et al. [5] also designates anchor frames in advance, and these frames can be used to describe the interpolation transformation curve in color grading. Above mentioned methods can eliminate flickers, but they are designed for specific applications and may fail to remove flickers produced in general situations. In this paper, we put forward a general video deflickering approach without the limitation of ceratin application.

Because video flickers mentioned in this paper are mainly caused by the loss of temporal coherence between neighboring frames, many methods manage to recover the temporal coherence in a video with a least-squares energy, and the least-squares energy contains an objective temporal consistency term at least [7, 9, 25]. For instance, Bonneel et al. [6, 32] formulate the video deflickering objective with an energy function, but they are used to address the flickers from video intrinsic decomposition. Although above mentioned methods can recover the video temporal coherence, they ignore spatial coherence. Thus, a video deflickering approach which can maintain spatial consistency is imperative.

Generally speaking, utilizing non-flickering frames to reconstruct a flickering frame is useful in some cases, but when an object presents in current frame for the first time, it is inexact to estimate the information from it last adjacent frame. Compared with video temporal consistency method in [7], which warps a non-flickering video frame just with an adjacent frame, multi-frames based reconstruction can reduce the warp inaccuracy efficiently. As demonstrated in [27], if one object or a scene in current frame once appeared in a video sequence and is missing in the previous frame, since lack of occlusion content of current, utilizing the method of [7] may fail to generate a temporally consistent output. Instead of warping single previous frame to reconstruct current frame, we warp multiple temporally corresponding frames to reconstruct a flickering frame. In our proposed multi-frames matching model, selected multiple temporally corresponding frames contain sufficient information to obtain the occlusion or visible content in current frame. Considering this, a multiple frames based video flicker removal model is proposed in this paper. Initially, the simple linear iterative clustering (SLIC) method [1] is used to segment each frame in the original video into superpixels. With the image superpixel method, the spatial texture features and boundary structure can be well-preserved at a region level. Then, we adopt SIFT Flow [21] to realize dense correspondence between adjacent frames, and the correspondence will be used to construct corresponding frames set. Next, with output frames of the corresponding frames for an original frame we can reconstruct an output frame of the original frame. Most importantly, we enforce the video fidelity, temporal and spatial coherence to remove flickers with a least-squares energy during reconstruction. Finally, the objective least-squares energy can be solved by the least angle regression [11], and the temporal consistent output video can be obtained. Our video flicker removal model can be summarized as Algorithm 1.

This paper mainly makes following contributions:

– We propose a multi-frame based video flicker removal method, where we first match multiple temporally corresponding frames, and then we use these frames to reconstruct the flickering frames. Multiple frames reconstruction can reduce warp inaccuracy.

– We propose algorithmic optimization and GPU acceleration to improve the efficiency of our proposed video flicker removal method. With algorithmic optimization and GPU acceleration, deflickering results and algorithmic running time outperform traditional video temporal coherence methods.

Algorithm 1 Video flickering removal using temporal reconstruction optimization.

Require: Original video, processed video
Ensure: Output video
Segment original frames into superpixels with SLIC;
for each superpixel region in current frame **do**
    Utilize SIFT Flow to find corresponding pixels between frames;
    Establish temporally corresponding frames set for the original frame;
end for
for each original frame in the original video **do**
    Construct video fidelity term with (4);
    Construct temporal coherence term with (5);
    Construct spatial coherence term with (8);
    Formulate video flickering removal objective with a least-square energy (2);
    Calculate the least-square energy with least angle regression;
end for
Output non-flickering video;

2 Related work

Temporal consistency has been researched extensively. In this paper, we focus on removing video flickering artifact and discuss some important and related works on this topic.

2.1 Flickers compensation based methods

Flickering artifact compensation based methods, which intend to remove flickering artifacts by aligning the tonal or brightness level between frames, is not the typical choice for removing temporal artifacts, so only a few techniques along these lines have been proposed.

By aligning color or tone of adjacent frames, flickers can be removed in a certain extent. So, flicker compensation based methods are always utilized to remove flickering artifacts in color transfer process. In [15], to eliminate the flicker flaw in video halftoning, they propose a video halftoning method which is composed of temporal and spatial error diffusion. The standard flicker compensation model is frame align. For instance, in [12, 30], a desired number of frames are designated as key frames used to align other frames.

In addition, some authors propose selecting key frames first to align other frames. For example, Bonneel et al. [5] first interpolates the transformation between input video and model video, and then it selects a set of key frames which can represent the transformation curve to manually refine the color grade. In [17], the pixel values of the remapped frames are interpolated according to the key colors in the re-mapped video. In [31], they first select key frames with scene change to store the visual information of the temporally occluded or disappearing objects, then they warp these key frames to reconstruct the frames. Finally, they employed a color affine model to preserve the image details.

However, above mentioned methods are all designed for color transfer. To remove flicker artifacts in video dehaze, Shin et al. [28] utilizes adaptive temporal average to calculate atmospheric light values. In these methods, the key frames need to be selected first. So they fail to be directly used in our method where the key frames selected from the processed video have flickering artifact and aligning other frames to them is inaccurate.

2.2 Adjustable parameter based methods

There are some methods based on the adjustable parameter models [4, 14, 25, 26, 33, 34]. By adjusting parameters to alter color or tone of the video, flickering artifacts can be removed. Ye et al. [32] propagates the clustered reflectance values of the first frame to subsequent frames based on a Bayesian framework. In [25], they formulate the temporal artifact in the tone mapping process as an optimization problem, and the solution of the optimization problem is a set of adjustable parameters used to minimize the difference between the original scene and the HDR displaying scene. Oskam et al. [26] formulates the optimized shape parameters in the color balancing process as a normalized radial basis function.

Some methods can adjust the parameter according to the experimental results and remove flickers iteratively, and feedback in the process can make the flicker removal result better. For instance, Guthier et al. [14] detect the temporal artifact by comparing the log average pixel brightness between two consecutive frames with a preset threshold, and the threshold is an adjustable parameter based on the experimental results. With an adjustable parameter, Zhao et al. [34] optimize the spatial-temporal coherence to remove haze in the unmanned aerial vehicle aerial video, Bhattacharya et al. [4] utilize video decomposition to detect and remove intensity flicker. In [33], they design a temporal consistency filter to reduce the flickering artifacts, and they further adjust the filtering process based on the video content and experimental results. Aydin et al. [2] proposes a high-quality spatiotemporal edge-aware filtering method by a mathematically-justified iterative model.

Above mentioned methods can reduce flickering artifact in a video to some extent, but they are all designed for a certain specific application, while our proposed video flicker removal approach can be applied to more generic cases.

2.3 Least-squares energy based methods

Some methods remove flickers artifacts by minimizing the energy function which contains the temporal consistency term. In [6, 18], a temporal consistency term is added into original energy function. Because both of them are used to separate an image into shading and reflectance layer, applying their methods to more general cases is limited. Dan and Lischinski [8] constructs a graph whose edges represent the photo pairs with similar content, then it optimizes a cost function over the graph to remove flickering artifact in color consistency process.

The limitation of these methods is that they are all tailored to specific application. Application-specific methods are always unsuitable and inapplicable for more generic cases, while our video flicker removal approach can be applied without limitation of specific application scenario, and spatiotemporal consistency are preserved well in our method.

To remove the flickering artifact in more general cases, some state-of-arts video temporal consistency methods are proposed in [7, 9, 20]. However, Lang et al. [20] has to revise original enhancement algorithm, which is unknown in general cases. Dong et al. [9] deals with flickering result of enhancement methods, which is enforced with the spatial consistent enhancement prior, and [7] fails to take the spatial consistency into account. In more

extensive applications, based on motion track and superpixel image representation, Kanj et al. [19] proposes a flicker removal method to improve colorimetry consistency. To remove flickering artifacts in video deblurring, Gong et al. [13] proposes an energy according to the relationship of the current frame and its adjacent frames.

Compared with above least-squares energy based methods, our proposed video temporal consistency approach in this paper can be applied to remove flickering artifacts in general cases and operate without any prior knowledge.

3 Temporally and spatially corresponding frames estimation

In this model, we introduce how to compute spatiotemporally corresponding frames. First, we denote the i^{th} frame in an original video as F_i. After applying the image superpixel method SLIC to segment F_i [1], the superpixel regions can be obtained. We denote the x^{th} region in the i^{th} frame F_i as Ω_i^x, and the number of regions in F_i is denoted as A_i. To compute spatiotemporally corresponding frames, we first compute corresponding regions between frames. Before computing corresponding regions, we use SIFT Flow [21] to compute dense correspondence between adjacent frames, then the motion path of a scene point can be obtained by connecting the output corresponding pixels of SIFT Flow. We will judge two frames as corresponding pixels if they move along the same motion path, and we regard two regions Ω_i^x in frame F_i and Ω_j^y in frame F_j as corresponding regions if the amount of corresponding pixels between Ω_i^x and Ω_j^y is greater than a threshold. In case that Ω_i^x and Ω_j^y are corresponding regions, assign 1 to $\phi(\Omega_i^x, \Omega_j^y)$, otherwise assign 0.

With obtained corresponding regions for a region of a given frame in original video, we denote the corresponding regions set of Ω_i^x as $c(\Omega_i^x)$. In our temporally and spatially corresponding frames estimation approach, we compute a corresponding frame with both the number of corresponding regions $n(F_i, F_j)$ and frame interval $d(F_i, F_j)$ between them. We calculate the frame interval $d(F_i, F_j)$ as the absolute difference between the frame index of F_i and F_j, and we compute the amount of corresponding regions $n(F_i, F_j)$ with:

$$n(F_i, F_j) = \sum_{x=1}^{A_i} \sum_{y=1}^{A_j} \phi(\Omega_i^x, \Omega_j^y) \tag{1}$$

where A_i denotes the number of corresponding regions in the original frame F_i, and A_j denotes the number of corresponding regions in the original frame F_j. The number of regions $n(F_i, F_j)$ between F_i and F_j is obtained by traversing all regions in F_i and F_j. For one frame F_i, its temporally corresponding frames can be computed by the corresponding degree $\xi(F_i, F_j)$ between F_i and F_j. We calculate the corresponding degree $\xi(F_i, F_j)$ by $\xi(F_i, F_j) = n(F_i, F_j)/d(F_i, F_j)$. Representation in this way, it is obvious that either a greater number of corresponding regions or a smaller value of distance between F_i and F_j will contribute more for F_i and F_j to be corresponding frames.

4 Video flickering removal model

In this section, we introduce the construction process of the proposed Video Flickering Removal Model. We formulate the video fidelity, video temporal coherence and spatial

coherence objective as a least-squares energy, and the solution of the least-squares energy is the optimal output frame.

$$\arg\min_{O_i} \int [E_q(O_i) + \alpha_t \cdot E_t(O_i) + \alpha_s \cdot E_s(O_i)] \mathrm{d}v \tag{2}$$

where we denote the i^{th} output frame as O_i, v as the pixel spatial position in a frame. $E_q(O_i)$ is denoted as the video fidelity term, which is computed as the difference between a frame in output video and corresponding frame from the original video. $E_t(O_i)$ and $E_s(O_i)$ are denoted as the video temporal coherence term and spatial coherence term, respectively. α_t is the temporal coherence weight and α_s is spatial continuity weight. We assign corresponding value to the two weights according to spatiotemporal coherence of the original video. If the spatiotemporal continuity of the original video is high, a larger value will be assigned to α_t or α_s.

4.1 Video fidelity term

In order to preserve video content in the processed video, we make an attempt to minimize the discrepancy between the original video and the processed video. With the aim, we formulate the video fidelity term as:

$$E_q(O_i) = \|O_i - P_i\|^2 \tag{3}$$

where O_i is the i^{th} output frame and P_i is the i^{th} processed frame. Representation in this form, flickers in the processed video have a negative impact on the output video. So in our model, we represent video fidelity term in gradient field to get rid of the flickers in the processed video:

$$E_q(O_i) = \|\nabla O_i - \nabla P_i\|^2 \tag{4}$$

where ∇O_i is the gradient field of the i^{th} output frame, and ∇P_i is the gradient field of the i^{th} processed frame. In this way, the impact of flickers in the processed video on the output video can be reduced reasonablely.

4.2 Video temporal coherence term

As mentioned in Section 3, corresponding frames set $r(F_i)$ is made up of corresponding frames of an original frame F_i. We denote the corresponding frames set which is composed of frames locate in the previous (index of the frame is smaller than i) of F_i as $p(F_i)$. Since an output frame of F_i is denoted as O_i, output frames of frames in $r(F_i)$ or $p(F_i)$ compose corresponding output frames set $r(O_i)$ or $p(O_i)$. Location of corresponding frames can be both previous or latter of F_i, to save the time complexity of our approach, we reconstruct the output frame of F_i with its previous corresponding frames in $p(F_i)$. The temporal coherence cost $E_t(O_i)$ is computed as:

$$E_t(O_i) = \sum_{F_j \in p(F_i)} \omega_t(F_i, F_j) \left\| O_i - \mathrm{warp}(O_j) \right\|^2 \tag{5}$$

where we denote $p(F_i)$ as the previous corresponding frames set of F_i, and F_j as a previous corresponding frame of F_i. Mathematically explaining the previous corresponding frame, that is the index i of F_i is greater than that the index j of F_j. The temporal coherence

weight between F_i and F_j is denoted as $\omega_t(F_i, F_j)$, and the computation of $\omega_t(F_i, F_j)$ takes both the weight of corresponding regions $\omega_n(F_i, F_j)$ and the weight of distance $\omega_d(F_i, F_j)$ between F_i and F_j. In our model, $\omega_t(F_i, F_j)$ is calculated as $\omega_n(F_i, F_j)/\omega_d(F_i, F_j)$. We compute $\omega_n(F_i, F_j)$ as:

$$\omega_n(F_i, F_j) = \frac{n(F_i, F_j)}{\sum\limits_{F_{j_1} \in p(F_i)} n(F_i, F_{j_1})} \tag{6}$$

where $n(F_i, F_j)$ is denoted as the amount of corresponding regions between F_i and F_j, and F_{j_1} is denoted as a previous corresponding frame of F_i. We compute $\omega_d(F_i, F_j)$ as:

$$\omega_d(F_i, F_j) = \frac{d(F_i, F_j)}{\sum\limits_{F_{j_1} \in p(F_i)} d(F_i, F_{j_1})} \tag{7}$$

where $d(F_i, F_j)$ is as the frame interval between F_i and F_j.

4.3 Video spatial continuity term

To make tonal variation or brightness fluctuation between neighboring regions consistent, it is necessary to take spatial continuity into account. In our video deflickering model, we formulate the spatial continuity objective as:

$$E_s(O_i) = \sum_{x=1}^{A_i} \sum_{\Omega_i^y \in N(\Omega_i^x)} \omega_s(\Omega_i^x, \Omega_i^y) \sum_{F_j \in (c(\Omega_i^y) \otimes p(F_i))} \omega_t(F_i, F_j) \left\| O_i - \text{warp}(O_j) \right\|^2 \tag{8}$$

where A_i is denoted as the number of superpixel regions in F_i, and $N(\Omega_i^x)$ is denoted as the neighboring regions set of Ω_i^x. We denote the weight of spatial coherence between Ω_i^x and Ω_i^y as $\omega_s(\Omega_i^x, \Omega_i^y)$, and $c(\Omega_i^y) \otimes p(F_i))$ as frames set which is made up of feature frames. The feature frame refers to frame which contains corresponding regions $c(\Omega_i^y)$ of Ω_i^y. Meanwhile, the feature frame needs to be a previous corresponding frame $p(F_i)$ of F_i. Then with certain temporal coherence weight $\omega_t(F_i, F_j)$, output frames of these spatially corresponding frames can be utilized to output non-flickering frame O_i. We compute spatial coherence weight $\omega_s(\Omega_i^x, \Omega_i^y)$ as:

$$\omega_s(\Omega_i^x, \Omega_i^y) = \frac{R(\Omega_i^x)}{\sum\limits_{\Omega_i^{y_1} \in N(\Omega_i^x)} \Omega_i^{y_1}} \tag{9}$$

where $R(\Omega_i^x)$ is denoted as the spatial area of Ω_i^x, and $\Omega_i^{y_1}$ is denoted as a neighboring region of Ω_i^x. For the least-squares energy composed of video fidelity term, video temporal coherence term and spatial coherence term, we can solve it by the least angle regression [11].

Algorithm 2 Faster video flickering removal using temporal reconstruction optimization.

Require: Original video, processed video
Ensure: Output video
Segment original frames into superpixels with SLIC;
for each video frame **do**
    Detect the flickering frames with TMC
end for
for each superpixel region in the flickering frame **do**
    Utilize SIFT Flow to find corresponding pixels between frames;
    Establish previous temporally corresponding frames set for the original frame;
end for
for each detected flickering frame in the original video **do**
    Construct video fidelity term with (4);
    Construct temporal coherence term with (5);
    Construct spatial coherence term with (8);
    Formulate video flickering removal objective with a least-square energy (2);
    Calculate the least-square energy;
end for
Output non-flickering video;

4.4 Video flickering removal model optimization

To speed up our algorithm, we improve our approach with algorithmic optimization and GPU acceleration.

In our multiple frames based video flickering removal model, we reconstruct each frame in the original video with corresponding computed temporally or spatially frames. As a result, a non-flickering output video can be obtained at the expense of huge time consumption. However, is it really necessary to reconstruct each frame in the video? Based on this, we intend to optimize our video video flickering removal model and just reconstruct flickering frames. Detailedly speaking, given a frame with flickering artifacts, we compute the value of TCM (Temporal Consistency Metric) [31] for each frame. As explained in [31], larger TCM value indicates high temporal consistency. In [31], the temporal consistency metric (TCM) is proposed based on the warping errors and defined as:

$$\mathrm{TCM}_i = \exp(-|\frac{\|O_i - \mathrm{warp}(O_{i-1})\|^2}{\|F_i - \mathrm{warp}(F_{i-1})\|^2} - 1|) \tag{10}$$

where we first estimate the motion field by successive original frame, then we use the estimated motion to warp the intermediate frames. Finally, if the temporal coherence between original input video is small, the inconsistency between the output video could be accepted. In our optimization method, we set a TCM threshold in advance, and designate the frame whose TCM value is lower than the TCM threshold as flickering frame. Then our proposed temporally and spatially corresponding frames estimation method can be used to find the spatiotemporally corresponding frames of the flickering frame. Then video deflickering model can be used to reconstruct these flickering frames with its spatiotemporally corresponding frames. The whole optimization algorithm can be concluded as Algorithm 2.

Besides just optimizing the flickering frames in a video, we propose an optimization framework based on reconstruction process. In our method, we reconstruct a flickering

frame with multiple temporally corresponding frames. However, Given a frame F_i, providing that F_j is in the subsequent of F_i (the index j of the frame F_j is bigger than the index i of the frame F_i), when we intend to reconstruct O_i with the output frame O_j of F_i, since we process the whole video from left-right, F_j will be unprocessed. So as we have proposed in our method, we optimize our reconstruction process by just warp previous temporally corresponding frames. Compared with reconstructing with whole temporally corresponding frames, reconstruction just with previous frame can improve algorithmic accuracy due to the reduction of reconstructing from unprocessed temporally corresponding frames, and can accelerate algorithmic operation. Since we can reconstruct just with these already process temporally corresponding frames, so video flicker removal approach can also be real-time.

Based on existing algorithmic optimization, we add GPU acceleration to speed up our approach as well. In our paper, there are no correlations between the superpixel segmentation of each frame in the original video, so parallel calculation is also available between them. After calculating the superpixel segmentation, the SIFT key point matching for each image pair is parallelizable. Finally, we can obtain the corresponding frames for each frame based on the key point matching.

5 Experimental results

We demonstrate the performance of our proposed video flickering removal method by lots of contrast experiments. Experiments in this paper are done on a laptop with a 2.4GHz of Inter Core i3-3110M CPU, 8 GB memory, and video sets used in our experiments are provided by the videos in [7]. In our experiments, we apply various classical image processing methods to videos frame by frame, and then remove the flickering artifacts in the processed video with different video temporal consistency methods.

5.1 Contrast experiments

Color grading To measure the video flickering removal performance when we utilize the image-based color grading methods to original videos, the color grading method [5] has been applied to a video *Cable* frame by frame, and then flickers will produced in the processed video. As can be seen in Fig. 1b, in the processed video, there are flickers in the

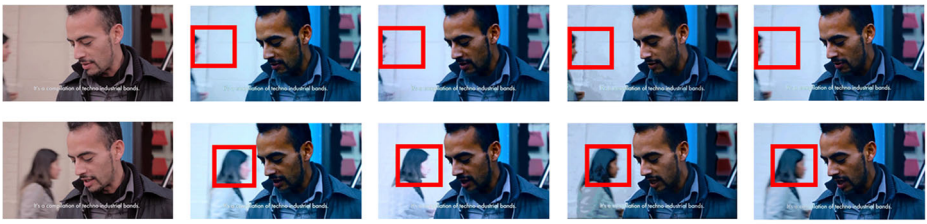


Fig. 1 **a** Shows the non-flickering original video *Interview*. After handling the original video with image-based color grading method, the flickers occur in the processed video (**b**), we can see from the brightness variation of the wall. The temporal coherence method of Lang [20] cannot reduce the flickers, which can be observed from the brightness variation on the face of the woman in corresponding output frames (**c**). Utilizing the method of Bonneel [7], the temporal artifacts can be reduced, but texture feature of the hair is lost (**d**). Because our video flickering removal model utilize both temporal and spatial consistency, our results (**e**) outperform results of Lang [20] and Bonneel [7]

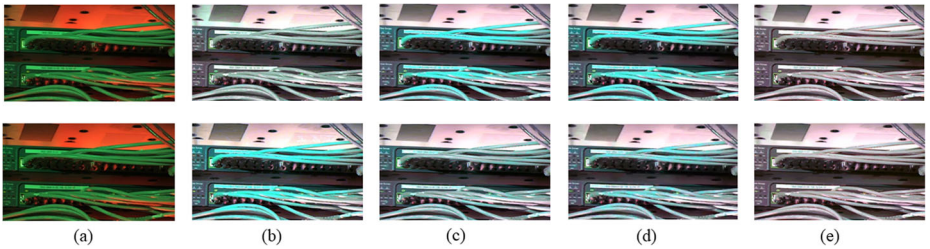


Fig. 2 **a** Shows two frames in the original videos *Cable*. When we apply the image-based spatial white balancing method to the original video, flickers occur (**b**). The generalized video temporal coherence method of Lang [20] cannot remove the flickers completely (**c**). Similar to the method of Lang [20], the method of Bonneel [7] also fails to maintain the temporal coherence (**d**), while temporal consistent video can be obtained with our proposed video flickering removal model approach (**e**)

brightness variation of wall. With the video deflickering methods of Lang [20], flickers in corresponding output video still remain (Fig. 1c). Compared with one state of the art video deflickering method Bonneel [7], our video deflickering methods (Fig. 1e) can handle spatial hair texture features better than the methods of Bonneel (Fig. 1d).

Spatial white balancing To measure the video flickering removal performance when we utilize the image-based spatial white balancing method to original videos, the spatial white balancing method [16] has been applied to a video *Cable* frame by frame. Then as shown in Fig. 2b, flickers occur. In the video deflickering results (Fig. 2c) of Lang [20] proves that the method fails to remove the inconsistency. As shown in Fig. 2d, deflickering results of Bonneel [7] share the same temporal inconsistency with the method of Lang [20]. However, as can be seen in Fig. 2e, non-flickering output video can be obtained with our deflickering approach.

Intrinsic images To measure the video flickering removal performance when we utilize the image-based intrinsic decomposition method, the intrinsic decomposition method [3] has been applied to an original video *Chicken* frame by frame, and then the temporal incoherence artifacts occurs (Fig. 3b). With the deflickering method of Lang [20], the flickers (Fig. 3c) still remains. Meanwhile, the method of Bonneel [7] cannot eliminate the flickers totally (Fig. 3d). Our approach can output temporally consistent videos (Fig. 3e).

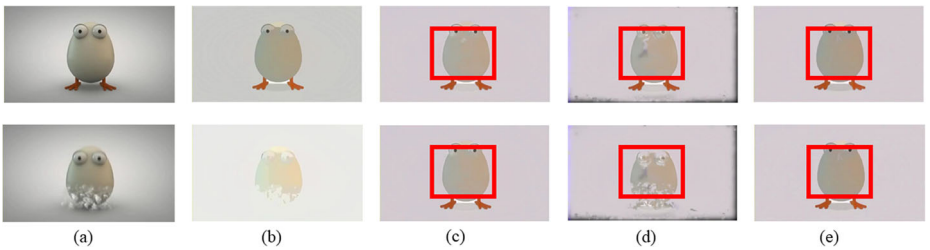


Fig. 3 We apply the image-based intrinsic decomposition to an original videos *Chicken*, then brightness variation occurs in the transferring process (**b**). The video temporal consistent result obtained with the method of Lang [20] cannot reduce the flickers completely (**c**), which have been emphasized with the red box. The method of Bonneel [7] fails to handle the boundary issue due to its limitation of only handling the temporal coherence (**d**). Our video flickering removal model outperforms the methods of Lang [20] and Bonneel [7] (**e**)

Fig. 4 After we apply the image-based hdr tone mapping method to the original video *CheckingEmail* (**a**), flickers occur (**b**) (please see the brightness variation between the bosk and the building in the close-up view). Although the method of Lang [20] maintain the video temporal coherence, it brings out blurring artifacts (**c**). The method of Bonneel [7] fails to preserve the texture feature, which can be seen from the color variation around the neck of the man (**d**). Our video flickering removal model outputs non-flickering video (**e**)

HDR To measure the video flickering removal performance when we utilize the hdr tone mapping method to original videos, the hdr tone mapping method [10] has been applied to a video *CheckingEmail* frame by frame. Then as shown in Fig. 4b, flickers occur. Since the video temporal consistency methods of both Lang [20] and Bonneel [7] ignores the spatial coherence, in Fig. 4c and d, blurring artifacts and color variation happen, respectively. However, as shown in Fig. 4e, we can output a non-flickering video with our method.

Dehazing To measure the video flickering removal performance when we utilize the image-based dehazing method to original videos, image-based dehazing method [29] has been applied to a video *Travel* frame by frame, then flickers occur (Fig. 5b). Because the methods of Lang [20] and Bonneel [7] lose sight of the importance of spatial coherence, their methods cannot remove the flickers. Since we take spatiotemporal coherence into account when reconstructing an output frame from multiple frames, as shown in Fig. 5e, our output video is temporal consistent.

Style transfer To measure the video flickering removal performance when we utilize the image-based style transfer method to original videos, image-based style transfer method has been applied to a video *Oldman* frame by frame, then flickers occur (Fig. 6b). From the results of Lang [20] and Bonneel [7] in Fig. 6c and d, it is obvious that their methods cannot

Fig. 5 To measure the deflickering performance when we utilize the image-based dehazing method to original videos, image-based dehazing method has been applied to a video *Travel* frame by frame, leading to the flickers (**b**) (please see the color variation emphasized in the red box). Utilizing the method of Lang [20] and Bonneel [7] cannot remove the flickering artifact completely, which can be seen from the results in (**c**) and (**d**). Our proposed video deflickering approach outputs non-flickering video (**e**)

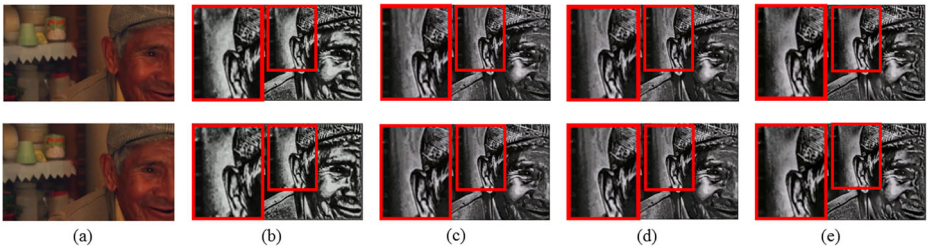


Fig. 6 To measure the deflickering performance when we utilize the image-based style transfer method to original videos, image-based style transfer has been applied to a video *Oldman* frame by frame, and then variation of the shadow area behind the right ear of the old man shows the temporal inconsistency. **b** Utilizing the method of Lang [20] and Bonneel [7] cannot remove the flickering artifact completely (**c**). The method of Bonneel [7] cannot eliminate the temporal inconsistency (please see the variation of the area behind the right ear on the close-up view). However, our proposed method can obtain temporally consistent output video (**e**)

remove the temporal inconsistency. However, as the results of our proposed method shown in Fig. 6e, we can output temporally consistent output videos.

5.2 Objective measurement

When evaluating the performance of our proposed video flickering removal method, we utilize the evaluation metric TCM (Temporal Consistency Model) proposed in [31] which takes both the original input video and the output videos into account. If successive frames of original video is temporally consistent, we expect a smoother output video. In a word, if the value of TCM is larger, the performance of removing flickers will be better. Figure 7 demonstrates the efficiency of our proposed video flickering removal method.

To measure the quality of the output video obtained with different video video flickering removal video flickering removal methods, we calculate the peak signal to noise ratio (PSNR). Figure 8 shows the psnr value of *Interview* video, *Cable* video, *Chicken* video, *CheckingEmail* video, *Travel* video and *Oldman* video. Because high psnr value ensures a

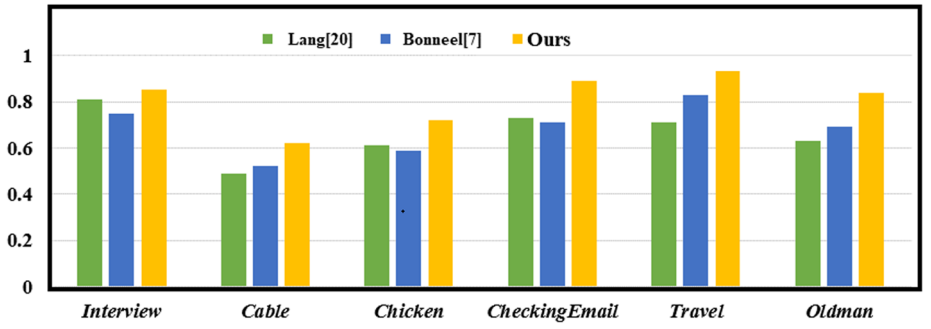


Fig. 7 Performance comparison by the TCM metric [31] on the *Interview* video, *Cable* video, *Chicken* video, *CheckingEmail* video, *Travel* video and *Oldman* video. Since a higher value of TCM always indicates a better algorithm performance, the result of the TCM metric comparison demonstrates the effectiveness of our proposed video temporal consistency approach

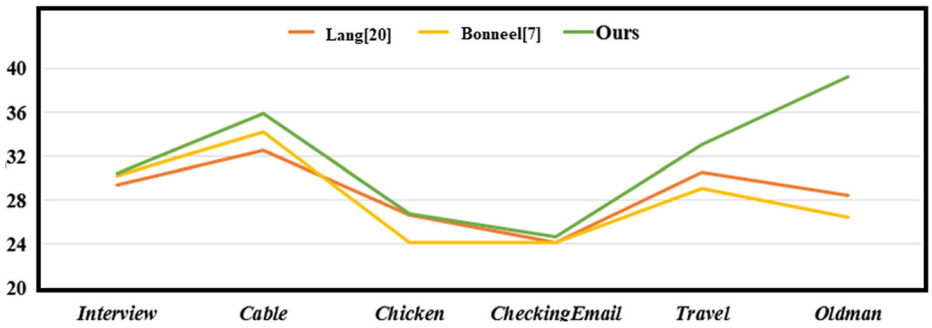


Fig. 8 The psnr value of output videos including *Interview* video, *Cable* video, *Chicken* video, *CheckingEmail* video, *Travel* video and *Oldman* video, which are generated from different temporal coherence methods including Lang [20] and Bonneel [7]

Table 1 Performance comparison of RMSE

Videos	**Lang** [20]	**Bonneel** [7]	**Ours**
Interview	2.9426	2.9398	2.6205
Cable	3.9990	3.8327	3.0594
Chicken	4.9952	4.8920	4.8889
CheckingEmail	3.9895	3.8727	3.2724
Travel	2.9988	2.8997	2.8108
Oldman	3.7821	3.8923	3.2321

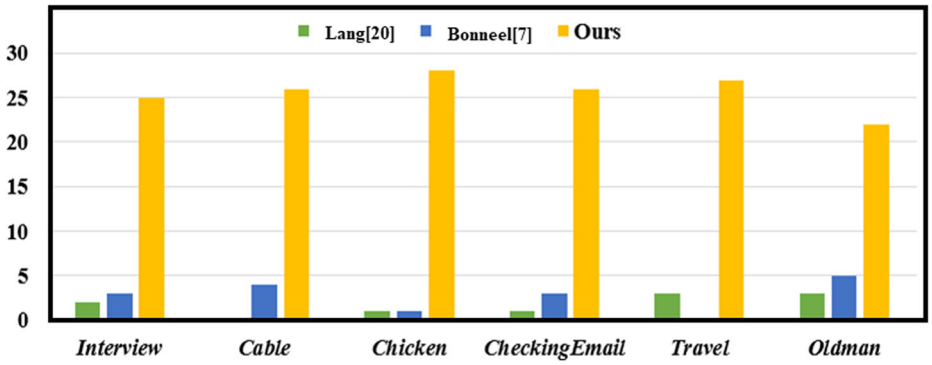


Fig. 9 The user study to evaluate the video deflickering performance of video temporal coherence methods, including Lang [20], Bonneel [7] and ours, on *Interview* video, *Cable* video, *Chicken* video, *CheckingEmail* video, *Travel* video and *Oldman* video

Table 2 Comparison of runtime in seconds

Video	Length	Size	Lang [20]	Bonneel [7]	Ours CPU	Ours GPU
Interview	200	960×540	187	64	57	0.53
Cable	77	1024×576	81	37	27	0.31
Chicken	92	1024×576	84	41	29	0.27
CheckingEmail	124	1280×720	241	87	64	0.58
Travel	153	1024×576	174	64	54	0.55
Oldman	133	1024×576	148	51	41	0.48

high-quality output video, it can be observed from Fig. 8 that the quality of our output video outperforms that of Lang [20] and Bonneel [7]. In addition, we compute the RMSE value between neighboring frames to weight the efficiency of deflickers, and lower RMSE value can describe a higher video temporal coherence. As shown in Table 1, the RMSE value of videos obtained via our approach is the lowest, which demonstrates the efficiency of our multiple frames based our proposed video flickering removal approach.

5.3 User study

To measure the video flickering removal performance from the view of users, 30 volunteers has been participated in our research. Firstly, we displayed videos with flickering ors non-flickers to them. Then output videos with video deflickering methods are shown to them, and volunteers choose the non-flickering video. As shown in Fig. 9, our video flickering removal approach outperforms the deflickering methods of Lang [20] and Bonneel [7].

5.4 Algorithmic efficiency

With proposed algorithmic optimization, run time of our approach is shown in Table 2. It can be noted that our video flicker removal approach outperforms [20] and [7]. To accelerate the speed of our whole algorithm, we reproduced our approach on a computer equipped with a Nvidia GTX 90 GPU with 8GB memory and the bandwidth between the PC memory and the GPU is 8GB/s. The whole proposed algorithm is implemented in C++ and we implement our parallel part experiment with CUDA. With algorithmic optimization and GPU acceleration, we evaluate the efficiency for the CPU version and GPU optimized version of our algorithm under different videos respectively in Table 2. Experimental results demonstrate that the efficiency after GPU acceleration has been improved a lot.

6 Conclusion and future work

To address flickering artifacts in videos, a multiple frames based video flicker removal approach is proposed in this paper. Since an object or a scene point may not always appear consecutively, utilizing an adjacent frame to reconstruct the flickering frame is not always sensible. We have proposed utilizing multiple frames to reconstruct the flickering frames to reduce the warp inaccuracy. To maintain the spatial coherence in a video and keep the variation of neighboring regions consistent, each original frame has been segmented into

superpixel regions using SLIC, and then we enforce spatial coherence on these superpixel regions. In addition, the temporal coherence and spatial continuity are also taken into account when reconstructing non-flickering video frames. Since the advantage of multiple corresponding frames and spatiotemporal coherence are utilized, our proposed video flickering removal approach outperforms other video deflickering methods. Experimental results demonstrate the efficiency of our proposed deflickering method. With using our proposed algorithmic optimization and GPU acceleration, the running time of our method also can compete with other state-of-arts video temporal coherence methods. In the future, we intend to take the flickering original video and processed video as input. Compared with utilizing the non-flickering original video and flickering processed video as input and extracting the temporal constancy information from the non-flickering original video, taking the flickering original video and processed video as input is challenging and interesting. In this way, whether the original video is flickering or not, we can always remove the flickering artifacts without inferring temporal information from the original video.

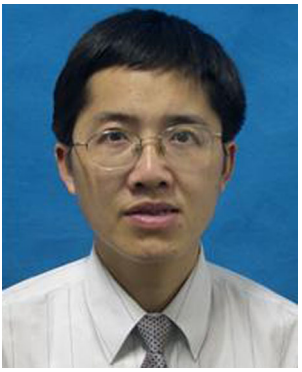
Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grant 61672228, Grant 61872241, Grant 61572316, and Grant 61370174, in part by the National Key Research and Development Program of China under Grant 2017YFE0104000 and Grant 2016YFC1300302, in part by the Macau Science and Technology Development Fund under Grant 0027/2018/A1, in part by the Science and Technology Commission of Shanghai Municipality under Grant 18410750700, Grant 17411952600, and Grant 16DZ0501100, and in part by the Shanghai Automotive Industry Science and Technology Development Foundation under Grant 1837.

References

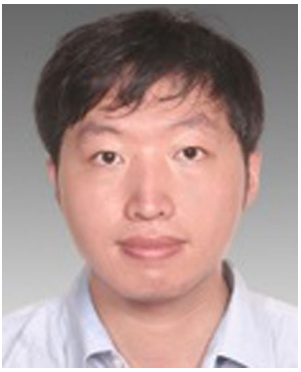
1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
2. Aydin T, Stefanoski N, Croci S, Gross M, Smolic A (2014) Temporally coherent local tone mapping of HDR video. ACM Trans Graph 33(6):196:1–196:13
3. Bell S, Bala K, Snavely N (2014) Intrinsic images in the wild. ACM Trans Graph 33(4):159:1–159:12
4. Bhattacharya S, Venkatesh KS, Gupta S (2016) Restoration of scene flicker using video decomposition. In: International conference on signal processing, computing and control, pp 396–400
5. Bonneel N, Sunkavalli K, Paris S, Pfister H (2013) Example-based video color grading. ACM Trans Graph 32(4):39:1–39:12
6. Bonneel N, Sunkavalli K, Tompkin J, Sun D, Paris S, Pfister H (2014) Interactive intrinsic video editing. ACM Trans Graph 33(6):197:1–197:10
7. Bonneel N, Tompkin J, Sunkavalli K, Sun D, Paris S, Pfister H (2015) Blind video temporal consistency. ACM Trans Graph 34(6):196:1–196:9
8. Dan BG, Lischinski D (2013) Optimizing color consistency in photo collections. ACM Trans Graph 32(4):38
9. Dong X, Bonev B, Zhu Y, Yuille AL (2015) Region-based temporally consistent video post-processing. In: IEEE conference on computer vision and pattern recognition, pp 714–722
10. Durand F, Dorsey J (2002) Fast bilateral filtering for the display of high-dynamic-range images. In: Conference on computer graphics and interactive techniques, pp 257–266
11. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–451
12. Farbman Z, Lischinski D (2011) Tonal stabilization of video. ACM Trans Graph 30(4):89:1–89:10
13. Gong W, Wang W, Li W, Tang S (2014) Temporal consistency based method for blind video deblurring. In: International conference on pattern recognition, pp 861–864
14. Guthier B, Kopf S, Eble M, Effelsberg W (2011) Flicker reduction in tone mapped high dynamic range video. In: Proceedings of the SPIE, vol 7866, pp 1–15
15. Hsu CY, Lu CS, Pei SC (2007) Video halftoning preserving temporal consistency. In: IEEE international conference on multimedia and expo, pp 1938–1941
16. Hsu E, Mertens T, Paris S, Avidan S, Durand F (2008) Light mixture estimation for spatially varying white balance. ACM Trans Graph 27(3):70:1–70:7

17. Huang CR, Chiu KC, Chen CS (2011) Temporal color consistency-based video reproduction for dichromats. IEEE Trans Multimedia 13(5):950–960
18. Kalantari NK, Shechtman E, Barnes C, Darabi S, Goldman DB, Sen P (2013) Patch-based high dynamic range video. ACM Trans Graph 32(6):202:1–202:8
19. Kanj A, Talbot H, Luparello RR (2017) Flicker removal and superpixel-based motion tracking for high speed videos. In: 2017 IEEE international conference on image processing (ICIP), pp 245–249
20. Lang M, Wang O, Aydin T, Smolic A, Gross M (2012) Practical temporal consistency for image-based graphics applications. ACM Trans Graph 31(4):34:1–34:8
21. Liu C, Yuen J, Torralba A (2011) SIFT flow: dense correspondence across scenes and its applications. IEEE Trans Pattern Anal Mach Intell 33(5):978–994
22. Liu Y, Nie L, Han L, Zhang L, Rosenblum DS (2015) Action2Activity: recognizing complex activities from sensor data. In: International conference on artificial intelligence, pp 1617–1623
23. Liu Y, Nie L, Liu L, Rosenblum DS (2016) From action to activity: sensor-based activity recognition. Neurocomputing 181:108–115
24. Liu Y, Zhang L, Nie L, Yan Y, Rosenblum DS (2016) Fortune teller: predicting your career path. In: AAAI conference on artificial intelligence, pp 201–207
25. Mantiuk R, Daly S, Kerofsky L (2008) Display adaptive tone mapping. ACM Trans Graph 27(3):1–10
26. Oskam T, Hornung A, Sumner RW, Gross M (2012) Fast and stable color balancing for images and augmented reality. In: Second international conference on 3d imaging, modeling, processing, visualization and transmission, pp 49–56
27. Reso M, Jachalsky J, Rosenhahn B, Ostermann J (2018) Occlusion-aware method for temporally consistent superpixels. IEEE Trans Pattern Anal Mach Intell PP(99):1–1
28. Shin DK, Yong MK, Park KT, Lee DS, Choi W, Moon YS (2014) Video dehazing without flicker artifacts using adaptive temporal average. In: The IEEE international symposium on consumer electronics, pp 1–2
29. Tang K, Yang J, Wang J (2014) Investigating haze-relevant features in a learning framework for image dehazing. In: IEEE conference on computer vision and pattern recognition, pp 295–302
30. Wang CM, Huang YH, Huang ML (2006) An effective algorithm for image sequence color transfer. Math Comput Model 44(7):608–627
31. Yao CH, Chang CY, Chien SY (2017) Occlusion-aware video temporal consistency. In: ACM multimedia, pp 777–785
32. Ye G, Garces E, Liu Y, Dai Q, Gutierrez D (2014) Intrinsic video and applications. ACM Trans Graph 33(4):80:1–80:11
33. Zeng H, Ma KK (2012) Content-adaptive temporal consistency enhancement for depth video. In: IEEE international conference on image processing, pp 3017–3020
34. Zhao X, Ding W, Liu C, Li H (2018) Haze removal for unmanned aerial vehicle aerial video based on spatial-temporal coherence optimisation. IET Image Process 12(1):88–97

Chao Li received her B.Eng. degree from Anqing Normal University, China. She is currently a postgraduate student in the Department of Computer Science and Engineering, East China University of Science and Technology. Her research interests include illumination-aware video composition video consistency, superpixel optimization, image/video processing and computer vision.



Zhihua Chen received his Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006. He is currently a Full Professor in the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China. His research interests include image/video processing and computer vision.



Bin Sheng received his BA degree in English and BE degree in computer science from Huazhong University of Science and Technology, and MS degree in software engineering from University of Macau, and PhD Degree in computer science from The Chinese University of Hong Kong. He is currently an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include virtual reality and computer graphics.

Ping Li received his Ph.D. degree from The Chinese University of Hong Kong. He is currently an assistant professor at Macau University of Science and Technology. His research interests include image/video stylization, big data visualization, GPU acceleration, and creative media. He has one image/video processing national invention patent, and has excellent research project reported worldwide by *ACM TechNews*.

Gaoqi He born in 1974. He is currently an associate professor at the Department of Computer Science and Engineering at East China University of Science and Technology. His research interests include video processing, crowd simulation, virtual reality and augmented reality.

Affiliations

Chao Li[1] · Zhihua Chen[1] · Bin Sheng[2,3] · Ping Li[4] · Gaoqi He[1]

✉ Bin Sheng
shengbin@sjtu.edu.cn

Chao Li
lichao_smile@163.com

Ping Li
pli@must.edu.mo

Gaoqi He
hegaoqi@ecust.edu.cn

1 Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China

2 Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

3 MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China

4 Faculty of Information Technology, Macau University of Science and Technology, Macau, China