第51卷第1期
2021年1月

中国科学技术大学学报
JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA

Vol. 51, No. 1
Jan. 2021

Received: 2020-12-15; Revised: 2020-12-23
doi:10.52396/JUST-2020-0022

MOVIE: Mesh oriented video inpainting network

LIU Sen, ZHANG Zhizheng, YU Tao, CHEN Zhibo*

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System,
University of Science and Technology of China, Hefei 230027, China
* Corresponding author. E-mail: chenzhibo@ustc.edu.cn

Abstract: Video inpainting aims to fill the holes across different frames upon limited spatio-temporal contexts. The existing schemes still suffer from achieving precise spatio-temporal coherence especially in hole areas due to inaccurate modeling of motion trajectories. In this paper, we introduce fexible shape-adaptive mesh as basic processing unit and mesh flow as motion representation, which has the capability of describing complex motions in hole areas more precisely and efficiently. We propose a Mesh Oriented Video Inpainting nEtwork, dubbed MOVIE, to estimate mesh flows then complete the hole region in the video. Specifically, we first design a mesh flow estimation module and a mesh flow completion module to estimate the mesh flow for visible contents and holes in a sequential way, which decouples the mesh flow estimation for visible and corrupted contents for easy optimization. A hybrid loss function is further introduced to optimize the flow estimation performance for the visible regions, the entire frames and the inpainted regions respectively. Then we design a polishing network to correct the distortion of the inpainted results caused by mesh flow transformation. Extensive experiments show that MOVIE not only achieves over four-times speed-up in completing the missing area, but also yields more promising results with much better inpainting quality in both quantitative and perceptual metrics.

Keywords: mesh flow; deep neural networks; video inpainting
CLC number: TU459　　**Document code**: A

1 Introduction

Video inpainting is of high importance for many professional video post-production applications, including video editing, scratch or damage video repair, logo or watermark removal in broadcast videos, etc[1-4]. The goal of video inpainting is to fill missed regions of a given video sequence with spatially and temporally consistent results. More challengeable than image inpainting in which only spatial consistency need be considered, improving aforementioned consistency requires us to not only exploit spatial contexts but also attach importance to exploiting the contents from nearby frames. To this end, solving the temporal misalignment problem for videos plays a dominant role.

For video data, temporal motions are complex due to local human or objects motions, global camera motions, and other environmental dynamics. The previous works which target video tasks such as video super-resolution and video stabilization[5-8] achieve promising results by explicitly taking advantage of motion information including optical flow, motion vector, homography, etc. Despite this, aligning features of adjacent frames in video inpainting is more challenging than other video tasks due to the pixel absence in the hole areas. Because the missed contents introduce noises and make the motion information not reliable as well.

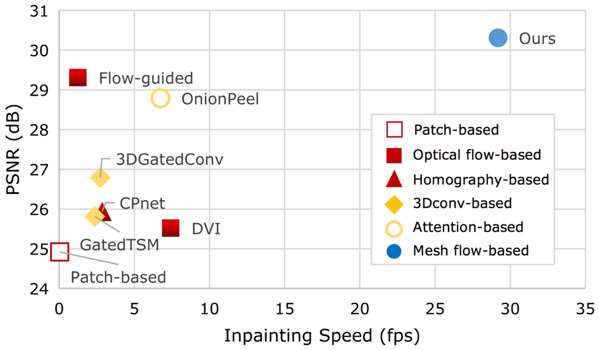

Figure 1. Comparison of the state-of-the-art methods in term of quality and speed on 37 video sequences with 2139 frames in total.

For the task of video inpainting, several temporal

Citation: LIU Sen, ZHANG Zhizheng, YU Tao, et al. MOVIE: Mesh oriented video inpainting network. J. Univ. Sci. Tech. China, 2021, 51(1): 1-11.

alignment solutions have been investigated, including patch matching methods, motion-based methods, 3D convolutional neural networks and attention-based neural networks. ① Patch matching methods select the most similar patch from adjacent frames for temporal coherence[9-12]. These methods may give rise to block artifact in the scenario with complex textures. ② Motion-based methods estimate motion information for copying corresponding contents from nearby frames, including optical flow and homography. Some works compute optical flows between two adjacent frames directly, which fail to provide precise flow prediction in the absence of pixel information[13-15]. A state-of-the-art optical flow-based work uses a sequence of flow maps from consecutive frames to complete the target optical flow[16]. However, the dense computation of pixel-level flow is a very time-consuming operation such that the efficiency is still under-explored. Homography-based methods use global affine transformation matrices to align two frames[13,17]. They only work well over plane motions or motions caused by camera rotations. ③3D convolutional neural networks use 3D filters to convolve the features from reference frames to target frame[18-20]. They have limited window size and suffer from high computation cost. ④ Attention-based neural networks compute similarities between the hole boundary pixels in the target and the non-hole pixels in the references[21] via attention module. They are unstable because the context information of the hole boundary are insufficient for similarity computation. So far, an efficient and accurate temporal alignment solution for video inpainting still remains under-investigated.

In addressing the temporal alignment problem, we propose to introduce flexible shape-adaptive mesh as basic processing unit and mesh flow as motion representation. We suggest that, in the case of videos with holes, mesh flow is more efficient and effective than other motion representations. First, mesh flow can represent more complex and coherent motions than homography and motion vector because it describes the motion trajectory of each pixel with multi-parameter model. Second, mesh flow can represent more accurate and robust motions in the hole area than optical flow because it can introduce richer spatial context information. Third, mesh flow is more computationally efficient than optical flow because it is a sparse motion field. More detailed theoretical analysis is illustrated in Section 3.1.

To take advantage of mesh flow towards more effective and efficient video inpainting, we propose a Mesh Oriented Video Inpainting nEtwork, called MOVIE, which consists of a sequential mesh flow estimation network and a polishing network. Since computing the mesh flow based on frames directly is unreliable and easily cause misalignments due to the existence of the holes in video, we design two sequential modules in the sequential mesh flow estimation network. Specifically, the mesh flow estimation module predicts mesh flows for the visible contents of the frames to guarantee the accuracy of the computed motions. Then the mesh flow completion module completes the mesh flow in the hole regions of the target frame by learning from a sequence of adjacent mesh flows. We design a hybrid loss function to optimize the flow estimation performance for the visible regions, the entire frames and the inpainted regions respectively, and train the sequential mesh flow estimation network in an end-to-end and self-supervised manner. For the polishing network, we align the frames with the estimated mesh flows in a propagate manner and feed them into the network for further refinement. The polishing network is trained to correct the distortion of the hole regions of each frame caused by mesh flow transformation. Experimental results reveal the superior of our method than the state-of-the-art schemes.

We summarize our contributions as follow:

(Ⅰ) We are the first one to propose to take advantage of mesh flows as motion representations in addressing the misalignment problem for video inpainting and demonstrate its superiority in both effectiveness and efficiency compared to other motion representations.

(Ⅱ) We propose a simple yet effective model and a hybrid loss function to better estimate mesh flows for the video with holes and better take advantage of the mesh flow for video inpainting.

(Ⅲ) We evaluate our method on various challenging videos and demonstrate our proposed approach can achieve impresive improvements in both effectiveness and efficiency compared to the state-of-the-art approaches.

2 related work

Recent years have witnessed remarkable progress in video inpainting by deep learning-based approaches. In this section, we provide an overview of the literature in terms of alignment techniques given their effectiveness in handling temporal consistency. We summarize the existing methods on video inpainting into four categories: patch matching methods, motion-based methods, 3D convolutional neural networks and attention-based neural networks.

2.1 Patch matching methods

Early works are mainly patch-based optimization methods[9-12]. They split each frame of the video into small patches and recover the hole region by pasting the most similar patch from other frames in the video. Since the patch-based alignment only describe simple translation motion for each patch, the block effects of the completed videos are obvious. Furthermore, the computation of patch similarity suffer from large search space, which make the completing process extremely slow.

2.2 Motion-based methods

Motion-based methods propose to estimate motion information first, then warp the content from nearby frames to target frame. The motion representations have been investigated in deep learning-based approaches are optical flow and homography.

Optical flow The widely used motion representation in deep learning based video inpainting is optical flow, which describes per-pixel motions between frames for warping the visible content of reference frames to the hole area of target frame. Five optical flow related works have been published, which explore various strategies to exploit optical flow field information[13-16,22]. Three works estimated flows only between the adjacent frames[13-15]. Chang et al. proposed a selective scheme to combine an optical flow warping model and an image-based inpainting model[15]. Ding et al. considered two branch optical flows generated from images and deep features separately[13]. Woo et al. used the computed flow field between the previous completed frame and the target frame as an auxiliary model to enforce temporal consistency[13]. However, estimating optical flow on the hole region directly is easy to lead incorrect flow prediction. Further, Kim et al. proposes to estimate flows of feature maps between the source and five reference frames in multi-scales, and complete the hole based on the aggregation of five aligned features[22]. Despite consideration of long-range frames, the flows are still computed on the hole area, which address the above issue with little success. Xu et al. proposed a Deep Flow Completion network to complete optical flow by watching a sequence of flow maps from consecutive frames, which further used to guide the propagation of pixels to fill up the missing regions in the video[16]. This strategy can provide more accurate optical flow estimation than previous approaches. Despite its significant performance improvements, the dense computation of pixel-level flow is a very time-consuming operation such that the efficiency is still under-explored.

Homography Two homography-based methods are proposed to predict global transformation parameters for aligning frames[13,17]. They both computed affine matrices between multiple reference frames and the target frame for the alignment, followed by an aggregation and refinement process. In the aggregation stage, Woo et al. proposed a non-local attention model to pick up the best matching patches in aligned frames[13], Lee et al. proposed a context matching module to assign weights for each aligned frames[17]. However, homography cannot be used to describe complex motions, limiting its application. Besides, to ensure long time dependency, they complete current frame by visiting the reference frames over a long-range distances, even the whole video shot. This strategy will result in intensive computational cost despite the simplicity of homography.

2.3 3D convolutional neural network

Several 3D convolutional networks are proposed to use 3D filters to convolve the features from reference frames to target frame, which equivalent to temporal alignment[18-20]. Wang et al. proposed a 3D-2D encoder-decoder network, which uses the output from 3D completion network to guide the 2D completion network[18]. Chang et al. did video inpainting with 3d gated convolutional and temporal patch discriminator[18], and further introduced a learnable gated temporal shift module to replace the computation-intensive 3D convolutional layer, which leads to a 3x reduction in computation[20]. However, the computation cost is still heavy. The temporal window size is limited in these 3D convolutional-based methods, hence they lack the ability to handle long time dependency challenge.

2.4 Attention-based neural network

Since attention module can be used for feature matching, Oh et al. proposed an asymmetric attention block to compute similarities between the hole boundary pixels in the target and the non-hole pixels in the references in a non-local manner[21]. The results are unstable regarding the complex situation and small dimension of the hole boundary.

3 Mesh oriented video inpainting network

3.1 Representations of motion information

Towards better understanding of mesh flow, we first formulate the concept of mesh flow and compare it with other motion representations.

Given a target frame $T(x, y)$ and a reference frame $R(x, y)$, we aim to find mapping functions $x' = f(x, y)$ and $y' = g(x, y)$ to minimize the following objective function:

$$E = \mathrm{d}\{T(x, y), R(x', y')\} \tag{1}$$

To compute mesh flow, we first need to partition the frame into non-overlapped regular blocks and treat each block as a basic processing unit mesh. Each mesh is a flexible shape-adaptive quadrilateral and can be arbitrarily transformed according to its four vertices. Then we compute motions of the pixels at the vertices of the mesh quadrilaterals, and interpolate the motions of other pixels based on the motions of the vertexes with bilinear interpolation kernel. The model of the

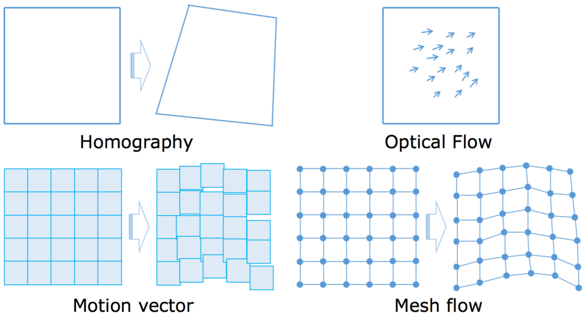

Figure 2. Motion representations, including homography, optical flow, motion vector and mesh flow.

motions of the vertices in the mesh quadrilaterals can be expressed as followed:

$$\left.\begin{aligned} f(x_v,\ y_v) &= x_v - d_x \\ g(x_v,\ y_v) &= y_v - d_y \end{aligned}\right\} \qquad (2)$$

where v denotes the vertices in the mesh quadrilaterals.

Mesh flow possesses several characteristics: ① mesh flow describes the motion trajectory of each pixel with multi-parameter model, namely, the motion of each pixel is computed with four nodal motions in its belonged mesh quadrilateral, ②mesh flow represents coherent motion trajectories across mesh quadrilaterals (see Figure 2), ③ mesh flow computes the motions of the vertices according to their four belonged quadrilaterals, which can introduce rich context information for each vertice, ④mesh flow is a sparse motion field.

Homography describes affine mapping between two images, which is formulated as below:

$$\left.\begin{aligned} f(x,y) &= a_0 + a_1 x + a_2 y \\ g(x,y) &= b_0 + b_1 x + b_2 y \end{aligned}\right\} \qquad (3)$$

where all the pixel share the same affine transformation parameters $a_0, a_1, a_2, b_0, b_1, b_2$.

Since homography can only represents camera moving over a stationary scene, it cannot handle complex scenarios with multi-object motions.

Optical flow computes motion information in pixel-level, the motion model of each pixel can be expressed as followed:

$$\left.\begin{aligned} f(x,\ y) &= x - d_x \\ g(x,\ y) &= y - d_y \end{aligned}\right\} \qquad (4)$$

It is worth noting that what we need are the motion trajectories of the pixels in the hole area. While optical flow cannot introduce rich context information for these pixels as mesh flow, the optical flow estimation in the hole area is not reliable. In addition, optical flow is a dense motion field, which makes the flow estimation computation-intensive and time-consuming.

Motion vector describes motion information in a patch-based manner. Since realistic motion in a patch may be more complicated than translation, motion vector cannot provide precise motion representation. Further, applying motion vectors as the description of motion information is easy to render blocking effect when predicting image due to discontinuity across block boundary (see Figure 2). In comparison, mesh flow can represent more complex non-linear motion transformation and produce more continuous results.

Overall, mesh flow is more efficient and effective compared with other motion representations for the video inpainting task, in which the video has holes.

3.2 Sequential mesh flow estimation network

Given a sequence of frames $\{I_t \mid t=1,\cdots,n\}$ with holes $\{H_t \mid t=1,\cdots,n\}$, our goal is to estimate the meshes $\{M_t \mid t=1,\cdots,n-1\}$ between frames wherein we also need take into account corrupted contents (corresponding to the holes).

Since the spatial information is not sufficient due to the holes in video, computing the meshes based on frames directly is unreliable and easily cause misalignments. Thus, we propose to estimate the mesh flow of each entire frame in a sequential manner. The first step is to compute the mesh flow corresponding to visible regions of frames. Afterwards, we estimate the mesh flow corresponding to the corrupted parts (i. e., holes) based on adjacent mesh flows. The framework is illustrated in Figure 3.

As the left part of Figure 3 shown, we first build a mesh estimation module to infer the mesh flows of visible regions between two adjacent frames I_{t-1} and I_t. The module encodes each frame with hole, $I_t \& (1-H_t)$ and $I_{t-1} \& (1-H_{t-1})$, then feeds the concatenation of the features into a sequence of residual blocks[23], and finally outputs the estimated mesh flow. Note that the final mesh flow M'_{t-1} is generated by multiplying the output mesh flow with the two holes $H_{t-1} \times H_t$. The size of mesh is decided by the size of video and partitioned block. If the size of video is $W * H$ and the size of block is $S_1 \times S_2$, then the size of mesh is $W/S_1 \times H/S_2$.

The right part of Figure 3 illustrates the process of completing invisible region of mesh flow. To complete the target mesh flow M'_{t-1}, we first collect N consecutive mesh flows $M'_{t-6}, \cdots, M'_{t-2}$ before the target mesh flow and N consecutive mesh flows $M'_t, \cdots, M'_{t+4}$ after it as references. Then we concatenate these $N\times 2$ mesh flows with the target mesh flow, and feed them into a residual block-based module to generate the final completed mesh flow. We set $N\times 2 = 10$ based on experimental analysis.

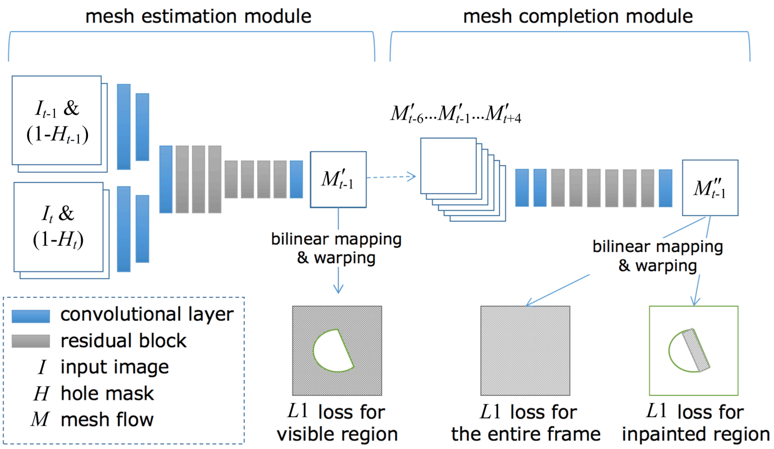
mesh estimation module
mesh completion module
I_{t-1} & $(1-H_{t-1})$
I_t & $(1-H_t)$
M'_{t-1}
$M'_{t-6}\cdots M'_{t-1}\cdots M'_{t+4}$
M''_{t-1}
bilinear mapping & warping
bilinear mapping & warping
convolutional layer
residual block
I input image
H hole mask
M mesh flow
$L1$ loss for visible region
$L1$ loss for the entire frame
$L1$ loss for inpainted region

Figure 3. Overview of the sequential mesh estimation network. The network consists of two modules: mesh estimation module and mesh completion module. The mesh estimation module predicts meshes for the visible regions of the frames, and the mesh completion module completes the missing area of mesh of the target frame by learning from a sequence of adjacent meshes.

3.3 The hybrid loss function for the sequential mesh flow estimation network

We optimize the sequential mesh flow estimation network in a self-supervised manner. Firstly, we upsample the mesh flows to pixel-level flow maps by interpolating from nodal motions with bilinear interpolation kernel. Then we align the frames based on the pixel-level flow maps, and optimize for minimizing the $\mathscr{L}_1$ distance between the target frames and the aligned frames.

For the first mesh flow estimation module, we adopt the $\mathscr{L}_1$ loss for visible regions. Since $(2\times N+1)$ mesh flows are needed for completing one mesh flow, we compute the average value of these $(2\times N+1)\backslash\ \mathscr{L}_1$ distances.

$$\mathscr{L}_{1(\text{visible region})} = \frac{\sum_{i=t-N-1}^{t+N-1} H_{i-1}\odot H_i\odot \| Vert\ I_i - \omega'(I_{i-1},\ \theta') \|_1}{2\times N+1} \tag{5}$$

where I denotes the input frame, H denotes the holes, ω' denotes the warping function with the predicted mesh flow θ'.

For the second mesh flow completion module, we propose two $\mathscr{L}_1$ loss functions, loss for the entire frame and loss for the inpainted regions.

$$\left.\begin{array}{l} \mathscr{L}_{1(\text{frame})} = \| I_t - \omega''(I_{t-1},\theta'') \|_1 \\ \mathscr{L}_{1(\text{inpainted region})} = \\ \omega''(H_{t-1},\theta'')\odot(1-H_t)\odot \| I_t - \omega''(I_{t-1},\ \theta'') \|_1 \end{array}\right\} \tag{6}$$

where I denotes the input frame, H denotes the holes, ω'' denotes the warping function with the final completed mesh flow θ''.

In summary, the hybrid loss function of the sequential mesh flow estimation network is as follows:

$$\mathscr{L} = \mathscr{L}_{1(\text{visible region})} + \mathscr{L}_{1(\text{the frame})} + \mathscr{L}_{1(\text{inpainted region})} \tag{7}$$

3.4 Polishing network

We illustrate the procedure of the polishing network in Figure 4. Given the mesh flows estimated by the sequential mesh flow estimation network, we perform the alignment procedure between two frames as described in section 3.3. To get the final aligned results, we perform the alignment operations for the frames (with the holes) pair by pair from the first two frames to the last two frames, and then repeat the same procedure backwardly. Then we concatenate the aligned frames and holes with the input frames and holes, and feed them into a residual block-based polishing network to generate the final refined output.

We illustrate the procedure of the polishing network in Figure 4. The whole procedure has three steps: forward propagation, backward propagation and refinement. In the first step, we first warp the visible content of the first frame to the hole of the second frame with mesh flow, and update the mask of the second frame. We do the same warping operation one by one from the first frame to the last frame. In the second step, we do the same procedure as the first step backwardly. In the final step, we concatenate the aligned frames and holes with the input frames and holes, and feed them into a residual block-based polishing network to generate the final refined output.

We design two loss functions specific to the inpainted region and the entire frame respectively to train the polishing network. Among them, $\mathscr{L}_1$ based loss function is designed for refining adopted contents in the hole regions of the frame, while the adversarial loss[24] is designed for making the completed contents more realistic and more consistent with visible regions.

$$\left.\begin{array}{l} \mathscr{L}_{1(\text{inpainted region})} = (1-H)\odot \| I - \widetilde{\omega}(I,\ \widetilde{\theta}) \|_1 \\ \mathscr{L}_{\theta a\mathrm{dv}\theta} = E_i[\theta\log\theta D(I)] + \\ E_{\widetilde{\omega}(I,\ \widetilde{\theta})}[\theta\log\theta(1-D(\widetilde{\omega}(I,\ \widetilde{\theta})))] \end{array}\right\} \tag{8}$$

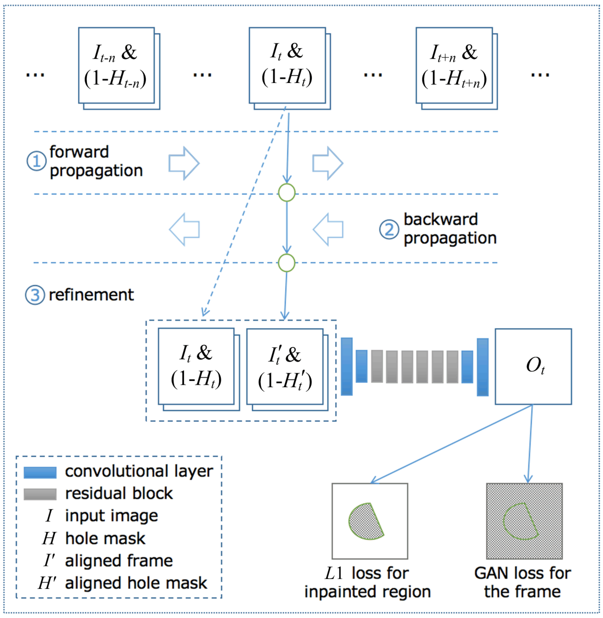

Figure 4. The process of video polishing. We first use the estimated mesh flows to warp the contents in a propagation manner forwardly, and then repeat the same procedure backwardly. Finally, we concatenate the aligned frames and holes with the input frames and holes, and feed them into a residual block-based polishing network to generate the final refined output.

where I denotes the input frame, H denotes the holes, $\widetilde{\omega}$ denotes the forward and backward warping function with the completed mesh flows $\widetilde{\theta}$.

In summary, the total loss of the polishing network is as follows:

$$\mathscr{L} = \mathscr{L}_{1(\text{inpainted region})} + \mathscr{L}_{\text{adv}} \tag{9}$$

3.5 Implementation detail

We train our method in two stage, first the sequential mesh flow estimation network, then the polishing network. The two models both run on hardware with the Intel(R) Xeon(R) CPU E5-2620 v4 and GeForce GTX 1080Ti GPUs.

We train the sequential mesh flow estimation network on 3471 videos in the YouTubVOS[25] dataset. For each sample, we select 12 frames with random frame step between 1 to 5 in one video. To collect the hole mask, we use the irregular mask dataset provided by an image inpainting work PartialConv[26], which contains 12000 mask files. We further augment the mask dataset to 480000 mask files by performing random translation, flipping and rotation. During training, we randomly select 12 masks for each sample.

We use the Adam Optimizer with $\beta = (0.9, 0.999)$ and learning rates $= 10^{-4}$. Using one GeForce GTX 1080Ti GPU, the early convergency takes about 8 hours, and the final convergency takes about one week (the PSNR improves about 1dB).

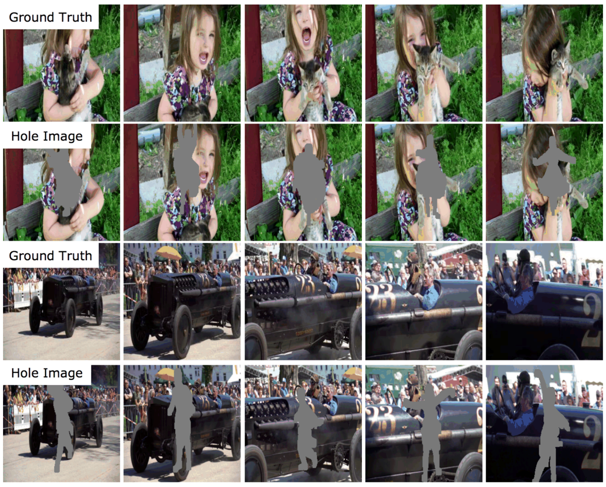

Figure 5. Video sequence for quantitative comparisons. The masks are selected from other videos in the DAVIS dataset.

Then we build training data for the polishing network by performing temporal alignment operations on the 3471 videos of the YouTubeVOS dataset. For each video, the alignment operation propagates forwardly and backwardly with the output of the sequential mesh flow estimation network. The random selected masks are also saved with the aligned results.

We use the Adam Optimizer with $\beta = (0.9, 0.999)$ and learning rates $= 10^{-4}$. Using one GeForce GTX 1080Ti GPU, the early convergency takes about 2 hours, and the final convergency takes about 3 days (the PSNR improves about 1dB).

4 Experiments

4.1 Evaluation datasets

To demonstrate the qualitative and quantitative performance of our proposed method MOVIE, we evaluate it on DAVIS[27,28] dataset, which consists of pixel-wise foreground object annotation.

For qualitative evaluation, we test on several video sequences with large motions and use the labeled pixel-wise foreground objects as holes. For quantitative evaluation, we randomly select 37 video sequences with 2139 frames in total in DAVIS dataset. Since the ground-truths of removed regions are not available while removing the objects directly in the video, we randomly select a mask sequence for each video from other videos in the DAVIS dataset. Figure 5 shows two samples of the test sequences, which contain large foreground objects in the hole region and the motions are complicated. We report the evaluation in terms of PSNR and SSIM, which are commonly used in video inpainting tasks. We also conduct ablation studies on these 37 video sequences. Inference speed is computed

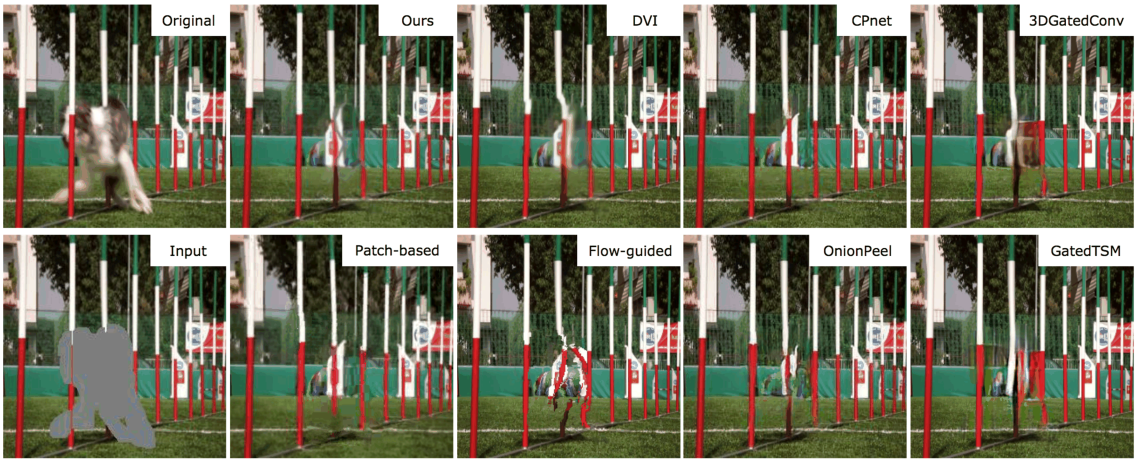

Figure 6. Qualitative results compared with the state-of-the-art methods on DAVIS dog-agility video sequence. Our method can complete the white and red pole with more precise and coherent structure.

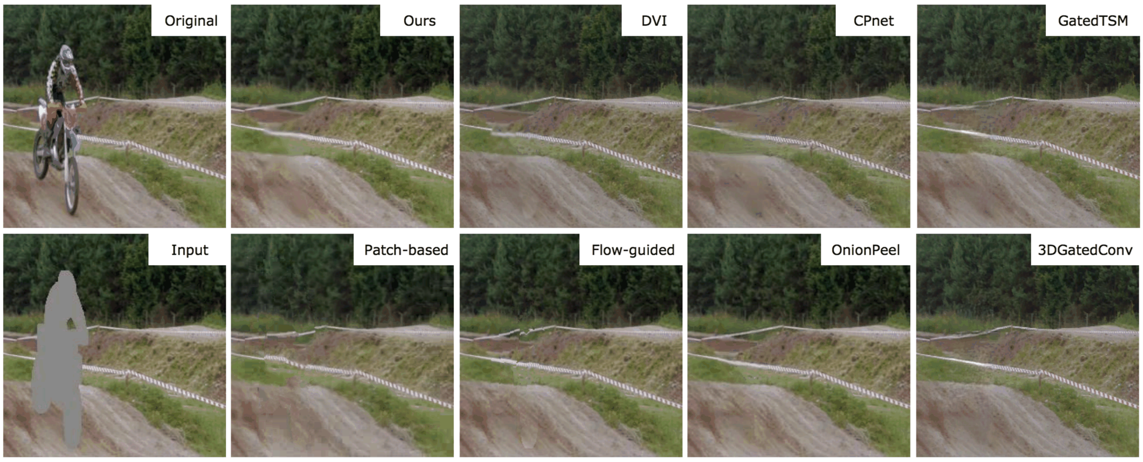

Figure 7. Qualitative results compared with the state-of-the-art methods on DAVIS motocross-bumps video sequence. Our method can complete the two white-stripe warning lines more continuously.

on a NVIDIA GTX 1080 Ti GPU for frames of 256x256 pixels.

4.2 Baselines

We compare our approach with the state-of-the-art approaches which can be categorized as follows:

Patch-based Patch-based[11] completes the hole of a video via patch-based similar matching.

Optical flow-based DVI[22] is a two-frame optical flow-based method, and Flow-guided[16] uses a sequence of flow maps from consecutive frames to complete the target optical flow.

Homography-based CPnet[17] does temporal alignment by computing affine matrices between two frames.

3D convolution-based GateTSM[20] and 3DGatedConv[18] both introduce new modules in 3D convolution layers for better performance and faster computation speed.

Attention-based OnionPeel[21] uses an asymmetric attention block to compute similarities between the hole boundary pixel in the target and the non-hole pixels in the references in a non-local manner.

4.3 Qualitative results

We illustrate the qualitative results in Figures 6 and 7, where the video sequence is a large motion video. As shown in the Figures, our method can complete the white and red pole with more precise and coherent structure. Our method is able to deal with the complicated situations, while the state-of-the-art methods have limitations in inpainting consistent results in which obvious artifacts can be observed. We illustrate the qualitative results in Figures 6 and 7, where these two video sequences are both large motion videos. As shown in the Figures, our method can complete the white and red pole with more precise and coherent structure (in Figure 6), and can complete the two white-stripe warning lines more continuously (in Figure 7). Our method is able to deal with these complicated situations, while the state-of-the-art methods have limitations in inpainting consistent results in which obvious artifacts can be observed.

The performance improvement would come from two aspects： the superiority of mesh flow and the design of the sequential mesh flow estimation network. Mesh flow can represent complex non-linear motion transformations and coherent motion trajectories across mesh quadrilaterals. Further, the design of the sequential mesh flow estimation network can guarantee the precise estimation of the mesh flow in the hole areas of the video.

4.4 Quantitative results

We show the quantitative results in Table 1. Our method produces significant improvement (more than 1dB PSNR) over the current state-of-the-arts on the challenging datasets which contain complex motions from foreground objects, and show speedups of up to about 4x against the fastest method and more than 20x against the best method.

The reconstruction values demonstrate that several methods fail to produce high reconstructed values due to the complex foreground motions in the hole region, including patch-based method[11], two-frame optical flow-based method DVI[22], homography-based method CPnet[17], 3D convolution-based methods GateTSM[20] and 3DGatedConv[18]. Their weaknesses have been analyzed in detail in section related work. Further, Flow-guided[16] and OnionPeel[21] can inpaint with higher PSNR but not reliable. Flow-guided[16] is easily affected by the noise introduced by the missed contents while estimates motion trajectories of the hole area in pixel-level. OnionPeel[21] is failed due to the small dimension of the hole boundary which used for computing similarities.

Table 1. Quantitative comparison with the state-of-the-art methods. Our method produces signicant improvement (more than 1dB PSNR) over the current state-of-the-art methods, and show speedups of up to about 4x against the fastest method and more than 20x against the best method.

Method	Type	PSNR	SSIM	Speed (fps)
Patch-based[11]	Patch-based	24.92	0.832	0.04
DVI[22]	Optical flow-based	25.52	0.860	7.42
GatedTSM[20]	3Dconv-based	25.81	0.867	2.38
CPnet[17]	Homography-based	25.94	0.869	2.86
3DGatedConv[18]	3Dconv-based	26.79	0.893	2.74
OnionPeel[21]	Attention-based	28.79	0.942	6.74
Flow-guided[16]	Optical flow-based	29.31	0.945	1.24
Ours	Mesh flow-based	**30.31**	**0.948**	**29.19**

In comparison, our method can handle complex motions more precisely and efficiently. The results show that mesh flow can be used to provide more precise temporal alignment with the well-designed sequential mesh flow estimation network. Meanwhile, our method is a computationally efficient solution.

4.5 Ablation study on the sequential mesh flow estimation network

In this section, we conduct a series of ablation studies to analyze the effectiveness of each component in the sequential mesh flow estimation network. Quantitative analyses are conducted on 37 video sequences described in Section 4.1 whose mask sequences are selected from other videos. We train each model in 50,000 iterations and optimize the models in the same training settings for fair compirason. The training process takes about 8 hours.

(Ⅰ) The effectiveness of the sequential mesh flow estimation： Our model estimates mesh flows in a sequential manner： first estimates mesh flows for visible contents of the frames, then completes the mesh flows of hole areas by learning from the adjacent mesh flows. To analyze the effectiveness of this sequential strategy, we compare with a direct mesh flow estimation model which estimates the mesh flows directly given a sequence of frames and holes. As illustrated in Figure 8, estimating mesh flow in sequential manner can generate more accurate mesh flows, while estimating mesh flows directly totally fails.

(Ⅱ) Ablation study on mesh size： As shown in Table 2, we analyze the influence of mesh size.

The results show that setting the mesh size to 8 can achieve better performance. The mesh size larger than 8 may fail to describe the complex motions in the holes. And the mesh size smaller than 8 cannot exploit sufficient information for aligning frames.

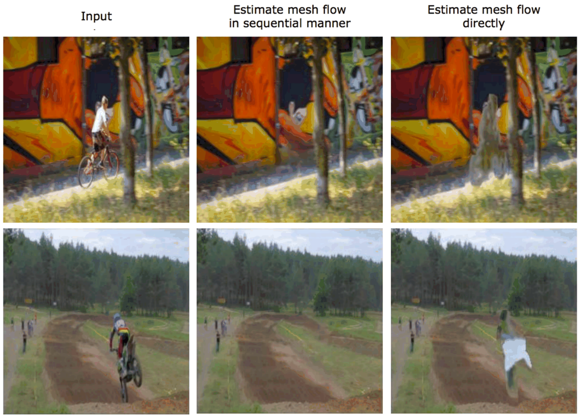
Input
Estimate mesh flow in sequential manner
Estimate mesh flow directly

Figure 8. Ablation study on the effectiveness of the sequential mesh flow estimation strategy.

Table 2. Ablation study on different mesh sizes.

mesh size	PSNR	SSIM
4	26.96	0.921
8	**28.25**	**0.930**
16	26.01	0.901
32	26.35	0.916

Table 3. Ablation study on number of references.

number of references	PSNR	SSIM
6	27.21	0.924
8	27.83	0.929
10	**28.25**	**0.930**

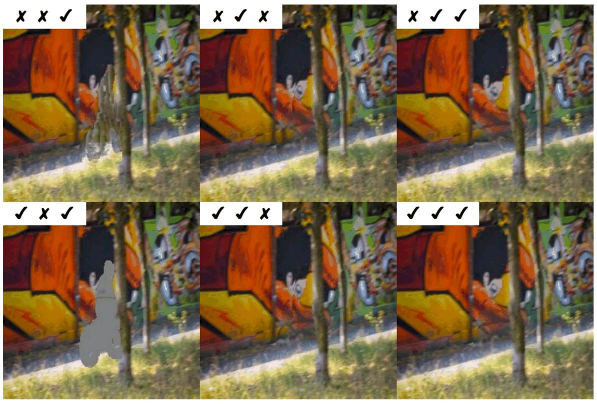

Figure 9. Ablation study on the hybrid loss function in the sequential mesh flow estimation network. The symbols in the left-top corner of each frame represent the three loss functions respectively: *L*1 loss for visible region, *L*1 loss for the entire frame, *L*1 loss for inpainted region.

(Ⅲ) Ablation study on number of references: We further analyze the influence of number of references. Note that the number of mesh flows larger than 12 will lead to out-of-memory error, hence we set the number of mesh flows up to 10. As shown in Table 3, larger number can lead to better performance.

(Ⅳ) Ablation study on the hybrid loss function: To evaluate the hybrid loss function of the sequential mesh flow estimation network, we train the model in different combination settings of the three loss functions in the hybrid loss function. As shown in Table 4, each of the three loss functions make positive contribution to the final performance.

The aligned results are illustrated in Figure 9. Specifically, the two results in the left column indicate that the results are totally failed without $\mathscr{L}1$ loss for the entire frame. The comparison between the results in the right column and the middle column shows that $\mathscr{L}_1$ loss for inpainted region can lead to more consistent texture. And the effectiveness of *L*1 loss for visible region is illustrated in the comparison between the two results in the right column.

Table 4. Ablation study on the hybrid loss function of the sequential mesh flow estimation network.

*L*1 loss for visible region	*L*1 loss for the entire frame	*L*1 loss for inpainted region	PSNR	SSIM
×	×	√	26.19	0.914
×	√	×	28.68	0.934
×	√	√	28.75	0.936
√	×	√	23.07	0.881
√	√	×	28.75	0.936
√	√	√	28.89	0.936

Table 5. Ablation study on Polishing Network. The polishing network can achieve 1dB PSNR improvement.

polishing network	PSNR	SSIM
with	29.25	0.945
w/o	30.31	0.948

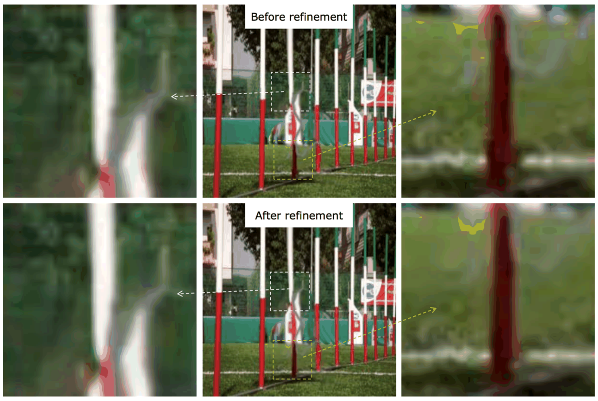

Figure 10. Ablation study on the polishing network. The polishing network can smooth the artifact of the result and make it more visually plausible.

4.6 Ablation study on the polishing network

In this section, we conduct ablation study regarding the polishing network. The results in Table 5 indicate that the polishing network can achieve 1dB PSNR improvement. Figure 10 shows that the polishing network can smooth the artifact of the result and make it more visually plausible.

5 Conclusion

In this paper, we propose an efficient and effective method for video inpainting. In essence, our main idea is to introduce mesh flow as a more proper representation of motion information so as to better target the temporal misalignment problem for video inpainting. Specifically, We design a sequential mesh flow estimation network which firstly predicts mesh flow only for visible regions of frames, then completes the holes of mesh flow by learning from the adjacent mesh flows. We further design a polishing network to polish the aligned results. Experiment results show that our method yields more promising results with higher inpainting quality in both quantitative and perceptual metrics, and achieves four-time speed-up at least in completing the missing area.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China(61571413,61632001) .

Conflict of interest

The authors declare no conflict of interest.

Author information

SEN Liu received the B. S. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013. Currently, he is working towards the PhD degree at School of Information Science and Technology, University of Science and Technology of China. His area of interests includes artificial intelligence, deep learning, video coding, computer vision and pattern recognition and reinforcement learning.

CHEN Zhibo (M'01-SM'11) received the B. S., and PhD degree from Department of Electrical Engineering Tsinghua University in 1998 and 2003, respectively. He is now a professor in University of Science and Technology of China. His research interests include image and video compression, visual quality of experience assessment, immersive media computing and intelligent media computing. He has more than 100 publications and more than 50 granted EU and US patent applications. He is IEEE senior member, Secretary (Chair-Elect) of IEEE Visual Signal Processing and Communications Committee. He was TPC chair of IEEE PCS 2019 and organization committee member of ICIP 2017 and ICME 2013, served as Track chair in IEEE ISCAS and Area chair in IEEE VCIP.

ZHANG Zhizheng (S'19) received the B. S. degree in electronic information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently pursuing the PhD degree in the University of Science and Technology of China, Hefei, China. His current research interests include reinforcement learning, few-shot learning, and intelligent media computing.

YU Tao is currently pursuing the PhD degree with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. He received the B. S. degree in Electronics and Information Engineering in Anhui University in 2018. His research interests include computer vision, image processing and reinforcement learning.

References

[1] HuoX, Tan J. A novel non-linear method of automatic video scratch removal. Fourth International Conference on Digital Home. Guangzhou, China: IEEE, 2012: 39-45.

[2] Jin X, Su Y, Zou L, et al. Video logo removal detection based on sparse representation. Multimedia Tools and Applications, 2018, 77(22): 29303-29322.

[3] Qin C, He Z, Yao H, et al. Visible watermark removal scheme based on reversible data hiding and image inpainting. Signal Processing: Image Communication, 2018, 60: 160-172.

[4] Le T T, Almansa A, Gousseau Y, et al. Object removal from complex videos using a few annotations. Computational Visual Media, 2019, 5(3): 267-291.

[5] Callico G M , Lopez S, Sosa O, et al. Analysis of fast block matching motion estimation algorithms for video super-resolution systems. IEEE Transactions on Consumer Electronics, 2008, 54(3): 1430-1438.

[6] Wang L, Guo Y, Liu L,et al. Deep video super-resolution using hr optical flow estimation. IEEE Transactions on Image Processing, 2020, 29: 4323-4336.

[7] Liu S, Yuan L, Tan P,et al. Steadyflow: Spatially smooth optical flow for video stabilization. IEEE Conference on Computer Vision and Pattern Recognition. Columbus, USA: IEEE, 2014: 4209-4216.

[8] Lim A, Ramesh B, Yang Y,et al. Real-time optical flow-based video stabilization for unmanned aerial vehicles. Journal of Real-time Image Processing, 2019, 16(6): 1975-1985.

[9] Granados M, Tompkin J, Kim K,et al. How not to be seen-object removal from videos of crowded scenes. Comput. Graph. Forum, 2012, 31(2): 219-228.

[10] Wexler Y, Shechtman E, Irani M. Space-time completion of video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(3): 463-476.

[11] Newson A, Almansa A, Fradet M,et al. Video inpainting of complex scenes. Siam Journal on Imaging Sciences, 2014, 7(4): 1993-2019.

[12] Huang J, Kang S B, Ahuja N,et al. Temporally coherent completion of dynamic video. ACM Transactions on Graphics,2016,35(6):196.

[13] Woo S, Kim D, Park K,et al. Align-andattend network for globally and locally coherent video inpainting. 2019, arXiv:1905.13066.

[14] Chang Y L, Liu Z Y, Hsu W. Vornet: Spatio-temporally consistent video inpainting for object removal. Proceedings of the IEEE conference on computer vision and pattern recognition workshops. Long Beach, USA: IEEE, 2019: 00229.

[15] Ding Y, Wang C, Huang H,et al. Framerecurrent video

inpainting by robust optical flow inference. 2019, arXiv: 1905.02882.

［16］ Xu R, Li X, Zhou B, et al. Deep flow-guided video inpainting. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA: IEEE, 2019: 3723-3732.

［17］ Lee S, Oh S W, Won D, et al. Copy-and-paste networks for deep video inpainting. Proceedings of the IEEE International Conference on Computer Vision. Seoul, South Korea: ACM, 2019: 4413-4421.

［18］ Chang Y L, Liu Z Y, Lee K Y, et al. Free-form video inpainting with 3D gated convolution and temporal patchgan. Proceedings of the IEEE International Conference on Computer Vision. Seoul, South Korea: ACM, 2019: 9066-9075.

［19］ Wang C, Huang H, Han X, et al. Video inpainting by jointly learning temporal structure and spatial details. Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA: IEEE, 2019, 33: 5232-5239.

［20］ Chang Y L, Liu Z Y, Lee K Y, et al. Learnable gated temporal shift module for deep video inpainting. 2019, arXiv:1907.01131.

［21］ Oh S W, Lee S, Lee J Y, et al. Onion-peel networks for deep video completion. in Proceedings of the IEEE International Conference on Computer Vision. Seoul, South Korea: ACM, 2019: 4403-4412.

［22］ Kim D, Woo S, Lee J Y, et al. Deep video inpainting. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA: IEEE, 2019: 5792-5801.

［23］ He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA: IEEE, 2016: 770-778.

［24］ Goodfellow I, Pougetabadie J, Mirza M, et al. Generative adversarial nets. Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014, 2: 2672-2680.

［25］ Xu N, Yang L, Fan Y, et al. Youtube-vos: A large-scale video object segmentation benchmark. Computer Vision and Pattern Recognition, 2018, arXiv:1809.03327.

［26］ Liu G, Reda F A, Shih K J, et al. Image inpainting for irregular holes using partial convolutions. Computer Vision and Pattern Recognition, 2018: 89-105.

［27］ Perazzi F, Ponttuset J, Mcwilliams B, et al. A benchmark dataset and evaluation methodology for video object segmentation. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, IEEE, 2016: 724-732.

［28］ Ponttuset J, Perazzi F, Caelles S, et al. The 2017 Davis challenge on video object segmentation. Computer Vision and Pattern Recognition, 2017, arXiv:1704.00675.

基于网格流的视频修补网络

刘森，张直政，俞涛，陈志波*

中国科学技术大学中国科学院空间信息处理与应用重点实验室，安徽合肥 230027

摘要：视频修补的目的是基于视频帧之间的时空域上下文信息修补空洞. 现有的方法由于不能准确地对运动轨迹进行建模经常导致修补结果无法保持良好的时空一致性. 为此引入灵活的形状自适应网格作为基本处理单元，将网格流用于运动表示，提出了一个基于网格流的视频修补网络，通过先预测网络流再添补空洞区域的方式对视频进行修补. 具体地，首先设计了一个网格流预测模块用于预测视频中可见内容的网格流的预测和一个网格流修补模块用于修补视频中空洞区域的网格流，通过这种方式将网格流的预测和修补解耦以达到更容易训练优化的目的. 我们进一步设计了一个混合损失函数用于同时优化可见区域、修补区域和整个视频帧范围的网格流预测结果. 为修正经过网格流变换引起的失真现象，最后设计了一个修补优化网络. 大量试验结果证明，本文提出的方法不仅从主观评判和客观指标得到相比于现有方法更好的修补结果，而且相比于现有最快的方法达到了 4 倍的速度提升.

关键词：视频修补；网格流；深度神经网络