Short-Long-Term Propagation-based Video Inpainting

Shibo Li, Shuyuan Zhu, Yuzhou Huang, Shuaicheng Liu, Bing Zeng
University of Electronic Science and Technology of China, Chengdu, China

Muhammad Ali Imran, Qammer H. Abbasi, Jonathan Cooper
University of Glasgow, Glasgow, UK

***Abstract*—In this paper, we propose a new method to inpaint videos with removed regions. Our method was developed based upon combining both short-term propagation-based inpainting (STPI) and long-term propagation-based inpainting (LTPI) modules. The STPI module is designed to in-fill an image from a single frame with local reference information, whilst the LTPI module uses multiple STPI modules to inpaint the whole video, giving a high temporal consistency and a low complexity. With both of the proposed modules, the correlated spatio-temporal information of frames can be propagated throughout the video, offering reliable short-long-term source information for inpainting. The experimental results demonstrate that our proposed method provides better results when compared with the state-of-the-art.**

■ **VIDEO INPAINTING** is used to in-fill missing regions of a given video after any undesired objects or visual artefacts have been removed. The technique can also be used to recover damaged video contents (including historical films), making the technique particularly useful for video editing and restoration. Video inpainting was initially developed based on image inpainting that has been successfully applied to the masked face completion and analysis [1-3]. Over recent years, a variety of methods have been proposed to implement video inpainting, including image patch-based approaches, combining propagation-based methods with deep learning-based algorithms.

The patch-based approaches [4, 5] are designed to fill the missing regions by using the available patches collected spatially or temporally from known regions of the video. In general, the filling of the missing regions can be implemented in either a greedy fashion [4] or a global fashion [5]. The methods effectively process the non-stationary inpainting scenes, although searching for such patches always results in rather high computational complexity, which limits their speed and the range of applications.

In contrast, propagation-based methods [5, 6] have been developed based on the spatio-temporal correlation of video frames. With the guidance of optical flow or homography, the source information collected for inpainting is propagated throughout the video so that the empty areas can be filled by the composed content with a high temporal coherence. The performance of these approaches depends upon the propagation efficiency. Consequently, the mechanisms of how to implement effective information propagation remains a key issue for the design of the propagation-based inpainting.

Recently, deep learning-based methods, including the application of convolutional neural networks (CNNs) [7, 8] and the attention-based ones [9, 10], have demonstrated impressive inpainting results for videos. Many of CNN-based methods design the end-to-end schemes and apply the 3D convolution to fuse spatial-temporal features for the synthesis of contents. However, these approaches often produce coarse textural detail due to the lack of the effective alignment of features. Attention-based methods have also been developed based upon the spatio-temporal context aggregation module to compose content, although, these approaches cannot produce fine-grained textures and thus cannot process complicated video scenes. Note that the deep learning-based approaches suffer from high computational cost and complexity, especially at the training stage. Moreover, the robustness of these methods are not as good as excepted. Their performance highly depends upon the training dataset.

Although a number of different methods have been proposed to implement effective inpainting for videos, there still exists key challenges in the design of a reliable inpainting scheme. Firstly, the camera motion induces parallax in the video scenes, which makes that in many cases, the captured video contains both foreground and background. Secondly, the generated content is not always of sufficient quality due to the lack of spatial and temporal coherency between adjacent frames. Finally, collecting reference information from the whole video often results in high computational complexity.

In order to tackle these problems, we propose a combined module comprising short-long-term propagation-based inpainting (SLTPI) to fill the missing areas of the video with removed objects. The proposed SLTPI is composed by both short-term propagation-based inpainting (STPI) and long-term propagation-based inpainting (LTPI). Specifically, the STPI module is designed to fill a single frame with the reference information obtained from its adjacent neighbors. In STPI, a depth-guided mesh-warping model with an illumination adaptation algorithm and a progressive fusion algorithm are developed to fill the missing region with high quality information. The LTPI module is constructed based on STPI but uses more reference information from more distant frames. Moreover, it is designed to inpaint the whole video by propagating the spatio-temporal information of frames through the video. In LTPI, the intra group of pictures (GOP) inpainting and the inter GOP inpainting are both developed to reduce the computational complexity.

RELATED WORK

Before deep learning was applied to video inpainting, most inpainting approaches were developed by filling the missing regions with the available patches collected spatially or temporally from the known regions, as so-called patch-based methods, which can be implemented in either a greedy or a global fashion. The greedy-based solutions [4] are used to fill the regions pixel by pixel, but often produce inconsistent results. To solve this problem, Huang *et al.* [5] introduced a global objective function with optical flow to optimize the patch searching and accordingly enforce temporal coherence of the result.

Propagation-based inpainting approaches focus on how to complete the whole video by propagating the correlated spatio-temporal information of frames through the video. For instance, Huang *et al.* [5] formulated a propagation-based algorithm for a joint color and flow optimization problem. By solving this, appropriate content was synthesized. However, this method produces over-smooth flow, which often results in blurred regions and boundaries in the most complicated scenes. To obtain the reliable flow for the construction of missing regions, Gao *et al.* [6] completed the flow estimation with edge guidance to improve inpainting performance.

Most recently, deep learning-based video inpainting has become increasingly popular. Lee *et al.* [7] constructed encoder-decoder models to aggregate information from multiple frames to inpaint the target frame. Oh *et al.* [9] proposed an asymmetric attention block for progressive region filling. Ouyang *et al.* [8] applied an internal learning to process challenging or complex scenarios which contain ambiguous backgrounds or long-term occlusion. Li *et al.* [10] proposed an end-to-end video inpainting by propagating features with the guidance of flow. Moreover, Zhang *et al.* [11] constructed a flow-guided transformer model for video inpainting.

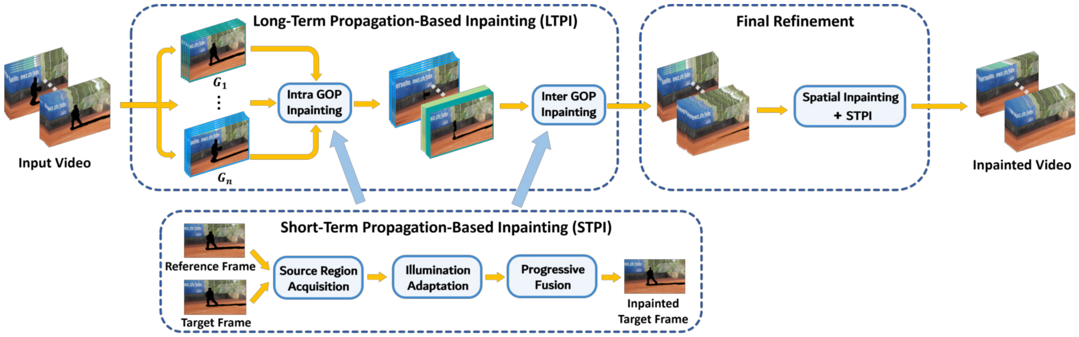


Figure 1. Schematic showing the pipeline of our proposed method. Firstly, the input video is divided into several GOPs. Then, the intra and inter GOP inpainting are sequentially performed to compose the LTPI module and both are constructed based on the STPI module. The STPI module consists of source region acquisition, illumination adaptation and progressive fusion, which are employed to obtain the reliable source region and seamlessly transfer it to target region. The final refinement, which is composed by spatial inpainting and STPI, is carried out if the frames cannot be completely filled by LTPI.

Overview

The pipeline of SLTPI is illustrated in Fig. 1, where both the STPI and LTPI modules collect the correlated spatio-temporal information to fill the missing areas of video frames. More specifically, the STPI module is designed to inpaint a single frame with the reference information provided by its adjacent neighbors. The LTPI module is designed to inpaint the whole video, with the aim of reducing complexity and maintaining high temporal consistency. The pipeline of activity is constructed based on STPI but can obtain more reference information from the long-distance frames. In addition, the final refinement adopted in SLTPI is used to inpaint the frames which cannot be completely filled by LTPI.

The STPI module consists of source region acquisition, illumination adaptation and progressive fusion. To acquire reliable source regions, we designed the depth-guided mesh-warping model to predict the motion for missing regions, which can be divided into single-layer and multi-layer alignments. Then we adujust the illumination of source regions to adapt to the target frame and seamlessly transfer them to target frame with progressive fusion.

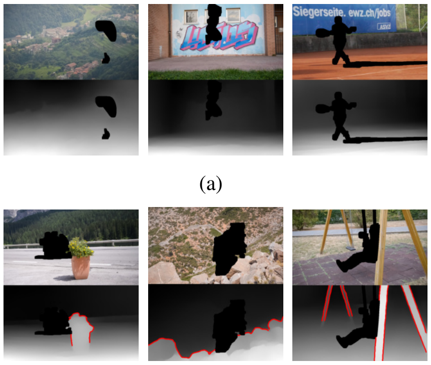
The LTPI module is implemented by progressively applying STPI to video frames. In LTPI, all the frames of a video are firstly divided into GOPs. Then, STPI is performed on the frames of each GOP along both the forward and backward directions, achieving the intra GOP inpainting. Subsequently, STPI is applied to the frames of two adjacent GOPs, implementing the inter GOP inpainting. If some frames of a video cannot be completely inpainted by LTPI, the spatial inpainting coupled with STPI is performed to complete them, which accordingly achieves the final refinement.

Short-term Propagation-based Inpainting (STPI)

The STPI module consists of source region acquisition, illumination adaptation and progressive fusion. For source region acquisition, we use the single-layer and multi-layer alignment-based techniques to obtain source region according to whether the frame includes multiple layers.

Source Region Acquisition

In our proposed STPI module, the inpainting of a target frame relies upon the source information collected from its neighboring reference frames. However, varied motion for different layers often induces misalignment between frames, often resulting in difficult acquisition of reliable information for inpainting. To tackle this problem, we firstly align the reference frame F_r to the target frame F_t. Then, with the user-specified mask, we obtain the source region from the aligned F_r to fill the missing region of F_t. In addition, we design two source region acquisition methods based on the single-layer alignment and multi-layer alignment for two scenarios. Different alignment is adopted according to the continuity


(a)

(b)

Figure 2. Examples of single-layer and multi-layer scenes from DAVIS dataset. The discontinuous depth boundary detected by Canny is highlighted in red color. (a) Single-layer scenes. First row: video frames. Second row: depth maps. (b) Multi-layer scenes. First row: video frames. Second row: depth maps.

of the depth map [12], where the existence of discontinuous object edges in the depth map indicates the existence of different layers in the frame. In our work, a Canny edge detector [13] is used to detect the discontinues edge in the depth, as illustrated in Fig. 2.

Single-layer alignment-based source region acquisition For the video whose frames do not contain obvious foreground and background layers, i.e., composed by only one layer, we design a single-layer alignment-based method to acquire the source region.

Single-layer alignment: For the single-layer scenario, we adopt the mesh-warping model [14] to align the reference and the target frames. This model warps a frame with local homography for alignment and introduces mesh grids to optimize the homography. Moreover, it does not separate the frame into different planes, which makes it applicable for the alignment of frames with single layer and continuous depth.

To construct the mesh-warping model, the matched features of F_t and F_r should be obtained first. Before generating features, we initially fill the missing regions of F_t and F_r by using the inward interpolation and the available boundary pixels. This processing constructs a smooth region to avoid the matched features of F_t and F_r falling around the boundaries of missing regions. After the initialization, we generate the SURF features [15] of both F_t and F_r. Then, we gather the matched features of these frames and apply an RANSAC algorithm [16] to remove undesired features. Finally, we use the remaining features to construct the mesh-warping model, which produces optimal local homography and accordingly guarantees a low warping loss for frame alignment.

In the mesh-warping model, assuming that $\hat{F}_r$ is a warped reference frame, let $\hat{V}_i = [\hat{v}_i^1, \hat{v}_i^2, \hat{v}_i^3, \hat{v}_i^4]^T$ represent the vertex vector of a grid cell of $\hat{F}_r$. The optimal warping is determined by minimizing

$$E(\hat{V}) = E_d(\hat{V}) + \eta E_s(\hat{V}) \tag{1}$$

where $\hat{V}$ is composed by all the warped grid vertices, E_d and E_s are the data term and similarity term, respectively, and η is the weighting factor.

By performing the bilinear combination on a mesh grid cell of $\hat{F}_r$, we obtain the feature $\hat{f}_i$ of $\hat{F}_r$ as $\hat{f}_i = c_i \hat{V}_i$, where $c_i = [c_i^1, c_i^2, c_i^3, c_i^4]$ is the bilinear weighting vector. The above minimization problem is quadratic and can be readily solved using a standard sparse linear solver as suggested in [14]. Then, the data term is defined

$$\begin{aligned} E_d(\hat{V}) &= \sum\nolimits_i \|\hat{f}_i - f_i\|_2^2 \\ &= \sum\nolimits_i \|c_i \hat{V}_i - f_i\|_2^2. \end{aligned} \tag{2}$$

The similarity term is defined to constrain the warping with small deformations. Note that each grid cell can be split into two triangles [14]. For each $\triangle v v_1 v_2$ whose vertices are v, v_1 and v_2, v can be represented by v_1 and v_2 in a local orthogonal coordinates system as

$$v = v_1 + (u_1 + u_2 R_{90})(v_2 - v_1) \tag{3}$$

where u_1 and u_2 are the coordinates of v in the local coordinate system and R_{90} is the 90° rotation matrix for v [14]. Based on Eq. (3), the similarity term is defined as

$$\begin{aligned} &E_s(\hat{V}) = \\ &\sum\nolimits_{\hat{v}} \|\hat{v} - [\hat{v}_1 + (u_1 + u_2 R_{90})(\hat{v}_2 - \hat{v}_1))]\|^2 \end{aligned} \tag{4}$$

where $\hat{v}$, $\hat{v}_1$ and $\hat{v}_2$ represent three vertices of the triangle in the optimized mesh grid. After substituting Eqs. (2) and (4) into Eq. (1), we can minimize the resulting $E(\hat{V})$ with a sparse

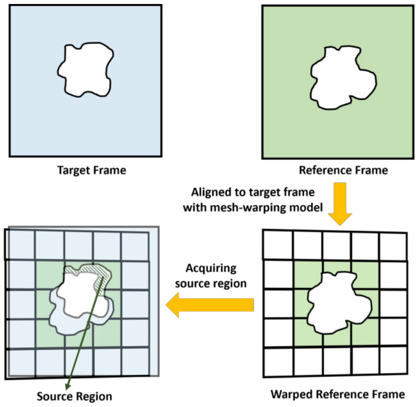


Figure 3. Source region acquisition with single-layer alignment.

linear system solver, which accordingly generates optimal mesh grid.

Once the optimized mesh grid is obtained, we can generate the local homography for each grid cell. To achieve a low complexity, we warp the local area rather than the whole reference frame to form the source region for the target region. The corresponding procedure is illustrated in Fig. 3. Firstly, based on the user-specified mask, we produce the warped reference sub-mask for each grid cell with its homography. Secondly, we form the warped reference mask $\hat{\Omega}_r$ with all the resulting sub-masks. Thirdly, we generate a binary mask Ω_o to determine whether we need to warp local area or not. In addition, Ω_o is used to obtain the reference information for inpainting from the warped local area. In this work, Ω_o is obtained as

$$\Omega_o = \Omega_t \odot (I - \hat{\Omega}_r) \tag{5}$$

where Ω_t is the target mask, $\odot$ is the element-wise product, and I is a mask whose elements are all ones.

Source region acquisition: Finally, if Ω_o is not an empty mask, i.e., the mask whose elements are all zeros, we will use it to obtain the source region from the local reference area. To obtain this area, we firstly form a regular mask, denoted Ω_R, with the grid cells of the given mask of the reference frame. Then, we get the reference region R_r

$$R_r = \Omega_R \odot F_r. \tag{6}$$

Next, we warp R_r with the local homographies and obtain the source region for inpainting as

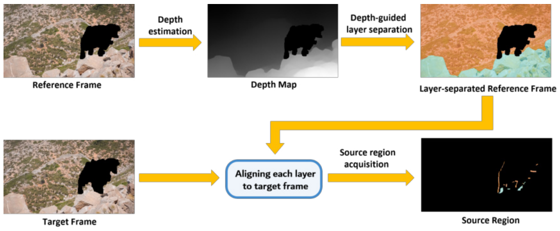


Figure 4. Source region acquisition with multi-layer alignment.

$$R_s = \Omega_o \odot warp(R_r). \tag{7}$$

If Ω_o is an empty mask, it indicates that we cannot get source region from neighboring frame. For this scenario, the LTPI module is adopted and it collects source region from long-distance frame for inpainting, which will be introduced in the construction of LTPI.

Multi-layer alignment-based source region acquisition method Single-layer alignment is used to obtain source region for inpainting the video whose frames do not contain obvious foreground and background, i.e., with continuous depth. It works well in the single-layer scenario because all the objects within a frame have similar motions. However, in the multi-layer scenario, the foreground and background produce a discontinuity in the depth, so resulting in different motions between them. If the single-layer alignment is applied to this scenario, the features of foreground are always treated as outliers and accordingly eliminated. Therefore, applying the single-layer alignment to multi-layer scenarios cannot achieve desired alignment. To tackle this problem, we separately warp different layers of the reference frame to implement the multi-layer alignment. With this alignment, we can obtain the reliable source region. The multi-layer alignment-based source region acquisition consists of depth estimation, depth-guided separation and source region acquisition, as illustrated in Fig. 4.

Multi-layer alignment: We adopt a monocular depth estimation method [12] in our work to obtain the depth map which is used to separate foreground and background.

We designed a depth-guided method to separate the foregrounds and backgrounds of frames. Depth map is obtained in depth-aided alignment decision. After separation, we aligned the corresponding layers of the reference and target frames to acquire the appropriate source information. To implement the layer separation, the frame is firstly split into a number of super-pixels. Then, according to the depth and color information, these super-pixels are clustered into two classes. With this classification, the frame is finally separated into foreground and background.

The super-pixel represents an irregular-shaped area which contains adjacent pixels with similar texture, color and depth. In our work, we firstly use the simple linear iterative clustering [17] to convert an $H \times W$ frame into an $H/n \times W/n$ super-pixel map, where n is the size of a super-pixel. Then, the mean values of both the color and depth of each super-pixel cluster are used to determine the classification of super-pixels, which makes a low complexity for layer separation. Based upon these mean values, a Meanshift method [18] is used to classify super-pixels.

In our work, we combine the color and depth of a super-pixel to form a classification parameter

$$t_i = |color_{mean} - color_i| + \lambda|depth_{mean} - depth_i| \tag{8}$$

where $color_{mean}$ and $depth_{mean}$ denote the mean values of color and depth of a cluster, respectively, $color_i$ and $depth_i$ are the color and depth values of a specific super-pixel, respectively, and λ is the depth weight. By comparing t_i with a given threshold T, we can accordingly divide all the super-pixels into two clusters. With these clusters, we separate the frame into foreground ($t_i > T$) and background ($t_i \leq T$). Furthermore, according to the resulting layers of a reference frame, we can divide the user-specified mask into the corresponding foreground and background masks.

Source region acquisition: After the depth-guided layer separation, the reference frame is divided into foreground and background. Then, we separately apply the single-layer-based acquisition to these layers to obtain the corresponding source regions for them. Finally, we combine these regions together to form source region for target frame.

To verify the effectiveness of multi-layer alignment-based acquisition, we applied the methods to the frames with multiple players and compared the result with those obtained using the single-layer alignment. The corresponding results, including the aligned reference frames, the alignment errors and the final inpainting results are all given in Fig. 5. The results presented in Fig. 5 indicate that using the multi-layer alignment-based methods provides reliable source information, which accordingly generates better inpainting results.

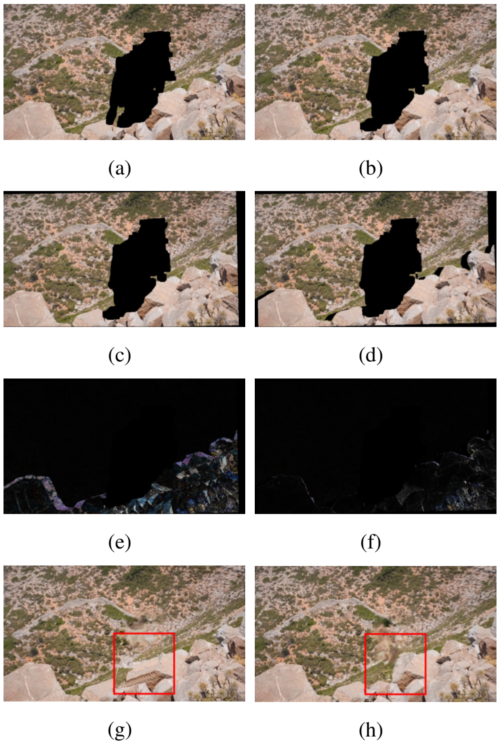


Figure 5. Results for the verification of multi-layer alignment. (a)-(b) Target frame and reference frame of *Goat* video from DAVIS dataset; (c) single-layer alignment result; (d) multi-layer alignment result; (e) single-layer alignment error; (f) multi-layer alignment error; (g) inpainting with single-layer alignment; and (h) inpainting with multi-layer alignment.

Illumination Adaptation

Any illumination change during video capture often induces brightness variation between frames. If we directly migrate the acquired source region to the target frame, it will result in apparent boundary effects in the inpainted frame. To tackle this problem, we designed an illumination adaptation algorithm and applied it to the obtained source region before transferring it to

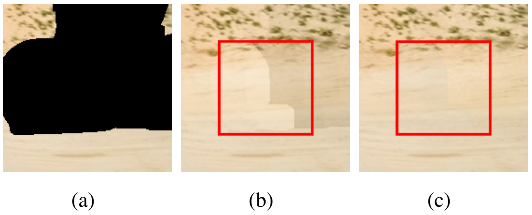

(a) (b) (c)

Figure 6. Results for the verification of illumination adaptation. (a) Selected local region for broken frame of *Bmp-bumps* video from DAVIS dataset; (b) with illumination adaptation; and (c) without illumination adaptation.

target frame. The illumination adaptation occurs in the LAB color space and the source region is converted from the RGB color space to the LAB space. After the color conversion, the L channel of the source region, denoted L_s, is adjusted as

$$\hat{L}_s = \alpha L_s + \beta\mu \tag{9}$$

where α is the scaling factor , β is the compensation term and $\mu = [1, ..., 1]^T$. In our method, the optimal (α,β) is determined by minimizing the difference between the source region L_s and the target region L_t in the LAB space. However, the target information is not available in the inpainting task. To solve this problem, we determined (α,β) based upon the common available information around L_t and L_s. Then, we used the resulting (α,β) to adjust L_s.

Let $\bar{L}_s$ and $\bar{L}_t$ represent the available neighboring areas for L_s and L_t, respectively. In our work, $\bar{L}_s$ and $\bar{L}_t$ are obtained by performing a specific mask Ω_L on the source region and the target region, respectively, where $\Omega_L = (I - \hat{\Omega}_r) \odot (I - \Omega_t)$. Meanwhile, due to the existence of local difference and warping error, some corresponding pixels of $\bar{L}_s$ and $\bar{L}_t$ are quite different to each others, which always degrades the accuracy to find optimal (α,β). To solve this problem, we used the absolute deviation-based method [19] to exclude these pixels. After that, the optimal (α,β) was determined by

$$(\hat{\alpha}, \hat{\beta}) = \underset{(\alpha,\beta)}{\arg\min} \, \|\bar{L}_t - (\alpha\bar{L}_s + \beta\mu)\|_2^2. \tag{10}$$

The closed-form solution to Eq. (10) is obtained by using the least squares estimation, i.e.,

$$\begin{cases} \hat{\alpha} = \dfrac{\bar{L}_s^T\bar{L}_t - \bar{L}_s^T\mu}{\bar{L}_s^T\bar{L}_s - (\bar{L}_s^T\mu)^2} \\ \hat{\beta} = \dfrac{(\bar{L}_s^T\bar{L}_s)(\bar{L}_t^T\mu) - (\bar{L}_s^T\bar{L}_t)(\bar{L}_s^T\mu)}{\bar{L}_s^T\bar{L}_s - (\bar{L}_s^T\mu)^2}. \end{cases} \tag{11}$$

We adjust L_s according to Eq. (9) and then convert it with the chrominance components to the RGB space. The converted result was finally used to fill the target region.

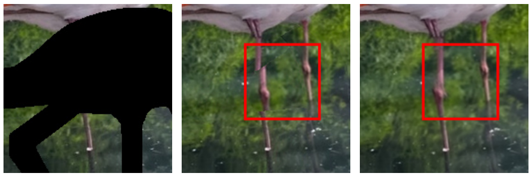
We present exemplar results in Fig. 6 to demonstrate the effectiveness of our proposed illumination adaptation, where the inpainting results with and without the adaptation are both given. According to these results, it is found that the illumination adaptation effectively reduces boundary effects and produces more pleasant inpainting result.

Progressive Fusion

To guarantee the continuity of the filled region and its neighboring area, we designed a progressive fusion algorithm to seamlessly transfer the neighboring area of the source region to the neighboring area of the target region. Specially, we combine those pixels around the boundary of the source region with the pixels at the same locations in the target region to fill the neighboring area of the target area. Let d denote the city block distance between a selected neighboring pixel $p_s(i,j)$ and the boundary of the source region. In this work, we select a number of neighboring pixels with different distances, i.e., $d = 1, 2, ..., d_{max}$, to progressively fuse them with the neighboring pixels of the target region, where d_{max} is the biggest range to collect pixels. With this distance, we define a weighting factor for each selected pixel as $\omega = 1 - d/d_{max}$. Then, we combine $p_s(i,j)$ with the corresponding pixel $p_t(i,j)$ in the target region to generate a fused pixel $\hat{p}_t(i,j)$

$$\hat{p}_t(i,j) = \omega \cdot p_s(i,j) + (1-\omega) \cdot p_t(i,j). \tag{12}$$

We demonstrated the effectiveness of the progressive fusion by comparing the inpainting results obtained with and without it. The corresponding results are given in Fig. 7. It is seen from Fig. 7 that adopting the progressive fusion in our method produces smoother result.



(a) (b) (c)

Figure 7. Results for the verification of progressive fusion. (a) Selected local region for broken frame of *Flamingo* video from DAVIS dataset; (b) with progressive fusion; and (c) without progressive fusion.

Long-term Propagation-based Inpainting

In this section, we designed the LTPI module based upon our proposed STPI to inpaint the whole video. This module can collect reference information from the long-distance frames. As a result, it may potentially obtain more available information for inpainting. Moreover, it was designed to rapidly deliver source information over frames and guarantee the temporal consistency of the inpainted video.

To implement LTPI, we firstly divided all the frames of a video into several GOPs, i.e., $\{G_1, G_2, ..., G_n\}$, where each GOP contains m frames. Then, we independently inpainted each G_i by using its inside reference information, which constructs the intra GOP inpainting. After all the GOPs were processed by the intra inpainting, we further filled the remained empty regions of the frames of G_i with the reference information offered by the other groups, which composed the inter GOP inpainting. Moreover, both the intra and inter GOP inpainting methods consisted of the forward propagation-based inpainting (FPI) and the backward propagation-based inpainting (BPI), where FPI and BPI were implemented based upon our proposed STPI. To implement FPI, the given frame $F^{(j-1)}$ was selected as the reference frame for its next frame $F^{(j)}$. In contrast, to implement BPI, $F^{(j+1)}$ was used as the reference frame for its previous frame $F^{(j)}$. We designed the GOP-based inpainting for LTPI so that we can avoid the propagation of inpainting errors over the whole video.

Intra GOP Inpainting

The intra GOP inpainting occurs inside each G_i, where FPI was firstly performed on the frames of G_i and then BPI was applied to them. To implement the intra FPI, our proposed STPI was used to fill the frames of G_i from the first to the last. In contrast, to implement the intra BPI, STPI was carried out from the last frame to the first frame of G_i. The implementation of the intra GOP inpainting is summarized in Algorithm 1.

Algorithm 1: Intra GOP inpainting

```
Input: Input video
Output: Inpainted GOPs
  $\{\hat{G}_1, \hat{G}_2, ..., \hat{G}_n\}$
Initialization: Dividing $S$ into GOPs
  $\{G_1, G_2, ..., G_n\}$
Processing frame $F_i^{(j)} \in G_i$:
for $i = 1 : n$ do
    Forward propagation-based inpainting
      (FPI):
    for $j = m : 2$ do
        Set $F_t = F_i^{(j)}$ and $F_r = F_i^{(j-1)}$;
        Update $F_i^{(j)}$:
          $F_i^{(j)} = STPI(F_t, F_r)$.
    end
    Backward propagation-based
      inpainting (BPI):
    for $j = 1 : m - 1$ do
        Set $F_t = F_i^{(j)}$ and $F_r = F_i^{(j+1)}$;
        Update $F_i^{(j)}$:
          $F_i^{(j)} = STPI(F_i^{(j)}, F_i^{(j+1)})$.
    end
    Composing $\hat{G}_i$ with the updated $F_i^{(j)}$.
end
```

Inter GOP Inpainting

After the intra GOP inpainting was applied to all the groups, each frame of a GOP has been fully or partially filled by using the source information collected from its neighboring frame(s). After that, if there were still empty areas in the frames of an inpainted GOP $\hat{G}_i$, the inter GOP inpainting was applied to it to fill the areas. In the inter GOP inpainting, the source information was collected from the other GOPs and propagated gradually from the close groups to the distant ones. This strategy guarantees that all of the available information of the adjacent frames but different groups can be used with high priority. It also avoided the accumulation and propagation

Algorithm 2: Inter GOP inpainting

```
Input: Intra GOP inpainted GOPs
  $\{\hat{G}_1, \hat{G}_2, ..., \hat{G}_n\}$
Output: Inpainted video
Processing $\{\hat{G}_1, \hat{G}_2, ..., \hat{G}_n\}$:
for $k = 1 : n - 1$ do
    for $i = n : -1 : 1 + k$ do
        if $\hat{G}_i$ is not completely inpainted
         then
            Update $\hat{G}_i$:
              $\hat{G}_i = FPI(\hat{G}_i, \hat{G}_{i-1})$
        else
            $Skip$
        end
    end
    for $i = 1 : n - k$ do
        if $\hat{G}_i$ is not completely inpainted
         then
            Update $\hat{G}_i$:
              $\hat{G}_i = BPI(\hat{G}_i, \hat{G}_{i+1})$
        else
            $Skip$
        end
    end
end
Composing video with the updated $\hat{G}_i$.
```

of inpainting errors over the whole video.

The inter GOP inpainting was implemented in an iterative manner. In each iteration, FPI was firstly performed on the specific frames of the video, and then BPI was carried out. Both FPI and BPI were applied to two adjacent GOPs, where one GOP offered reference frame for the inpainting of frames of the other. Specifically, FPI starts from the last two GOPs, i.e., $(\hat{G}_{n-1}, \hat{G}_n)$, while BPI starts from the first two GOPs, i.e., $(\hat{G}_1, \hat{G}_2)$. To fill an incomplete GOP, when FPI was carried out, the last frame of $\hat{G}_{i-1}$ and all the frames of $\hat{G}_i$ were firstly gathered according to the temporal order. Then, STPI was sequentially performed on these frames, i.e., from the first to the last frame. To implement the inter BPI, all the frames of $\hat{G}_i$ and the first frame of $\hat{G}_{i+1}$ should be collected firstly.

Subsequently, STPI was carried out from the last frame to the first one over the stored frames. After one iteration was accomplished, the first two adjacent GOPs will be removed from the FPI procedure in the next iteration and the last two adjacent GOPs will also be removed from the BPI processing in the next iteration. With the increment of iteration, the participated GOPs in both FPI and BPI were progressively reduced, which effectively avoided some repeated FPI and BPI operations over the same adjacent GOPs. By applying FPI and BPI to all the adjacent GOPs, the spatio-temporal correlated information of all the frames could be propagated over the whole video, which offers reliable reference for inpainting. Note that if $\hat{G}_i$ does not contain the incomplete frames, the inter GOP inpainting will be skipped. The detail to implement the inter GOP inpainting is summarized in Algorithm 2.

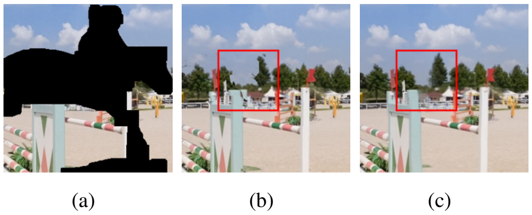

(a) (b) (c)

Figure 8. Results for the verification of inter GOP inpainting. (a) Selected local region for broken frame of *Horsejump-high* video from DAVIS dataset; (b) without inter GOP inpainting; and (c) with inter GOP inpainting.

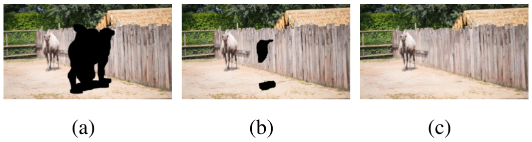

(a) (b) (c)

Figure 9. Results for the verification of final refinement. (a) Selected broken frame of *Camel* video from DAVIS dataset; (b) before final refinement; and (c) after final refinement.

We present results in Fig. 8 to demonstrate the effectiveness of the inter GOP inpainting by making a comparison between the results with and without this operation. One can see from Fig. 8 that the use of the inter GOP inpainting produces the good results, with propagation of the long-term source information over the video to reduce error accumulation.

Final Refinement

Our proposed propagation-based inpainting methods, i.e., STPI and LTPI, collect the source information from the other frames to fill the

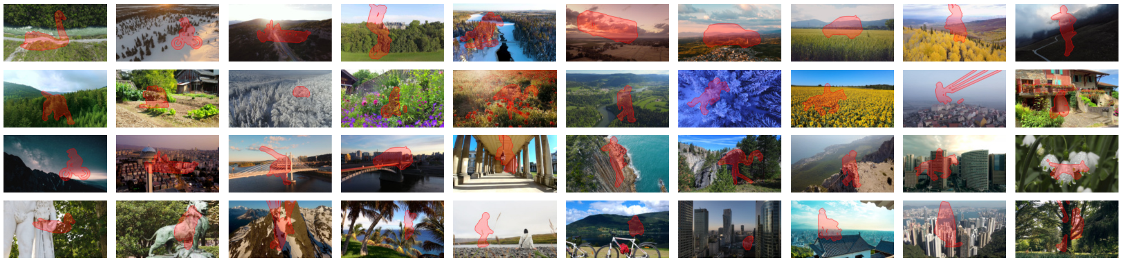


Figure 10. The Landscape dataset with object-like masks for quantitative experiments. The top two rows are single-layer scenes and the bottom two rows are multi-layer scenes.

missing regions of a given frame. If all the reference frames cannot provide enough information to completely fill the empty areas of some specific frames, we will employ the GAN-based spatial inpainting [20] coupled with the propagation-based inpainting to complete them, implementing the final refinement. The former inpainting is used to fill all the missing areas of a single frame with high visual quality and the latter one is applied to complete frames guaranteeing high temporal consistency and low complexity.

We demonstrated the effectiveness of the final refinement by comparing the inpainting results obtained before and after its application. These results are presented in Fig. 9 and they show that the final refinement completely fills all the missing areas of the target frame.

Experimental Results

Experimental Setup

We evaluated the performance of our proposed method by comparing it with the state-of-the-art methods, including three propagation-based methods, i.e., Huang's approach [5], FGVC [6] and FGT [11], and four deep learning-based methods, i.e., CPNet [7], OPN [9], IIVI [8], E2FGVI [10] and FGT [11]. Note that Huang's approach [5] is designed on the Matlab platform with CPU, while FGVC [6] and FGT [11] are developed based on the Pytorch platform with both CPU and GPU. The four deep learning-based methods are implemented on the Pytorch platform with GPU. In addition, our proposed method was developed based upon the Matlab platform with CPU except for the depth estimation and the final refinement which are implemented on the Pytorch platform with GPU. In this work, we adopt Matlab 2020b and Pytorch 1.2.0 with Inter(R)-Core(TM) i7-9700k 3.60GHz CPU and NVIDIA GTX 2080Ti 11GB GPU in the experiments.

Table 1. Quantitative Comparison of Different Methods on DAVIS dataset for video completion

Methods	Accuracy						Efficiency
	Square			Object			Runtime
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	(s/frame)
Huang [5]	31.53	0.9654	0.034	30.14	0.9509	0.043	4.03
CPNet [7]	30.43	0.9468	0.045	29.65	0.9331	0.051	0.40
OPN [9]	30.58	0.9492	0.047	29.57	0.9364	0.053	0.96
FGVC [6]	32.45	0.9709	0.028	31.16	0.9611	0.035	2.36
IIVI [8]	31.69	0.9628	0.031	31.22	0.958	0.037	110.15
E2FGVI [10]	33.78	0.9809	0.026	33.07	0.9777	0.028	0.16
FGT [11]	33.86	0.9831	0.024	32.89	0.9693	0.031	1.89
Ours	**34.04**	**0.9865**	**0.021**	**33.15**	**0.9806**	**0.025**	0.51

Quantitative Comparison

In quantitative comparison, we applied our proposed approach and the state-of-the-art methods, including Huang's method [5], CPNet [7], OPN [9], FGVC [6], IIVI [8], E2FGVI [10] and FGT [11], to two video datasets. The first one is the popular video segmentation dataset DAVIS and the second one is our collected dataset Landscapes that is composed by fourty videos without moving objects. We adopted three quantitative metrics, including PSNR (dB), SSIM, and LPIPS, to evaluate the efficiency of different methods.

Firstly, as previous work [6, 10, 11] did, we adopted 50 video clips from DAVIS dataset to evaluate quantitative performance and time efficiency. The resolutions of these videos are 240 × 432. We generated the stationary square masks and the temporally-varied irregular masks for video restoration scenario and object removal scenario, respectively. The quantitative evaluation results were given in Table. 1. It can be seen from Table. 1 that our method outperforms the state-of-the-arts on all the three quantitative metrics. Meanwhile, the runtime of different approaches are also presented in Table. 1. Although our

Table 2. Verification of Robustness for Different Resolution on Landscapes for object removal

Methods	PSNR/SSIM		
	480p	720p	1080p
Huang [5]	29.27/0.9644	30.30/0.9719	30.01/0.9673
CPNet [7]	29.21/0.9566	29.89/0.9660	30.68/0.9728
OPN [9]	28.32/0.9589	29.62/0.9657	-/-
FGVC [6]	33.24/0.9871	32.73/0.9875	32.33/0.9832
IIVI [8]	32.12/0.9782	31.84/0.9773	30.77/0.9745
E2FGVI [10]	33.40/0.9832	33.12/0.9827	-/-
FGT [11]	35.37/0.9901	35.24/0.9894	35.13/0.9878
Ours	**35.92/0.9913**	**35.58/0.9907**	**35.23/0.9896**

method was developed mostly based on CPU, its average processing time is comparable to that of the deep learning-based methods which were implemented based on GPU, which proves the time efficiency of our proposed approach.

Secondly, to verify the robustness of different methods, we applied them to the videos of our Landscapes dataset, where each video is transferred into three versions with three corresponding resolutions, i.e. 480p, 720p, and 1080p. We used the masks offered by DAVIS dataset to remove the content of video and fill the empty regions. Our Landscape dataset can provide ground-truth to evaluate object removal with object-like masks. Videos and masks from Landscape dataset are shown in Fig. 10. The corresponding results were offered in Table 2. OPN [9] and E2FGVI [10] cannot process the 1080p videos due to the limited GPU memory. One can see from Table 2 that our method achieves the best performance in different resolution scenarios with the given marks. This demonstrates its robustness to video resolution and removed content.

Qualitative Comparison

We carried out qualitative comparison experiments on DAVIS datset. These videos are also used in [5] for the quantitative comparison, where the user-specified object masks are given, and the videos contain both the single-layer scenes and multi-layer scenes. We demonstrate the inpainting performance of our method by performing it on these videos and make the qualitative comparison with state-of-the-art methods.

Firstly, we present some visual results in Fig. 11 to compare our method with the other methods. It is found from Fig. 11 that our method produces more pleasant results which contain smoother edges and clearer textures. However, the other methods, especially OPN [9], often generate blurred results and distorted semantic objects. All the results of our proposed method can be found in the website[1].

Table 3. Parameter Determination for Single-layer Alignment

	η=0.5	η=1.0	η=1.5	η=2.0
PSNR	35.81	**36.03**	35.72	35.68
SSIM	0.9901	**0.9922**	0.9892	0.9880

Table 4. Parameter Determination for Multi-layer Alignment

		PSNR (dB)	SSIM
n = 40	λ=0.5	35.54	0.9879
	λ=1.0	35.75	0.9903
	λ=1.5	35.68	0.9887
n = 80	λ=0.5	35.73	0.9896
	λ=1.0	**35.81**	**0.9904**
	λ=1.5	35.77	0.9902
n = 120	λ=0.5	35.41	0.9868
	λ=1.0	35.64	0.9974
	λ=1.5	35.28	0.9885

Secondly, besides of visual results, the temporal coherence evaluation adopted in [5] is used to compare the temporal consistency of the results obtained by using different methods. In this comparison, a slice of successive frames are selected and the spatio-temporal profile (the yellow line highlighted in frames) is given. We offer the temporal coherence results in Fig. 12. One can see from Fig. 12 that our result maintains the long-term temporal consistency and accordingly achieves better temporal coherence.

Thirdly, we carried out a user study to subjectively evaluate the inpainting results obtained with different methods. For each video, the source video and the inpainted videos were all displayed to ten participants and they were required to give a score from 1 to 5 to evaluate the quality of the result, where a higher score indicates a better result. The user study results are presented in Fig. 13 and these results show that most of our results have higher scores, which demonstrates the good performance of our proposed method.

Determination of Hyper-parameters

In our work, the hyper-parameters defined in the single-layer alignment, multi-layer alignment,

[1] https://drive.google.com/drive/folders/1Qcsn0cy36xRVcgKTLbeFxNpBPLBB-RIa

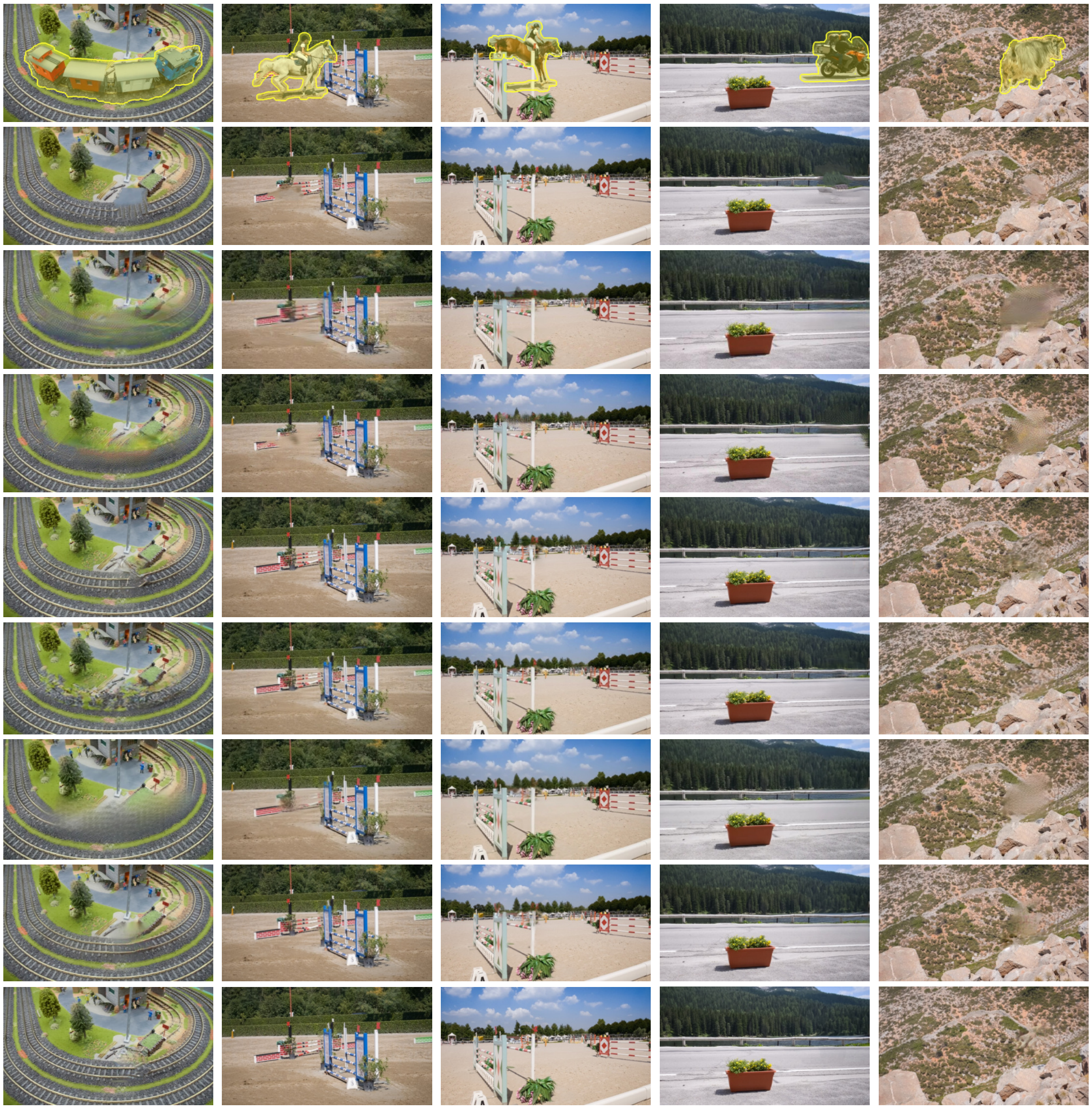


Figure 11. Qualitative comparison of inpainting results for some videos from DAVIS dataset. From left to right: *Train*, *Horsejump-Low*, *Horsejump-High*, *Motorbike* and *Goat*. From top to bottom: Mask, Huang [5], CPNet [7], OPN [9], FGVC [6], IIVI [8], E2FGVI [10], FGT [11] and ours.

progressive fusion were determined according to a number of preliminary experiments. These experiments were carried out on the videos from Landscape dataset with random masks using the 480p resolution.

For single-layer alignment To determine the parameter η used in the single-layer alignment, we conducted experiments with different η on 20 videos from Lanscape dataset, to select the most applicable parameter η. The average quantitative results over these videos are given in Table 3. Based upon the results presented in Table 3, we choose $\eta = 1.0$ for our work due to the best performance offered by it.

For multi-layer alignment We applied multi-layer alignment with various combinations of depth weight λ and super-pixel size n to broken landscape videos for the determination of them. The other 20 videos from Lanscape dataset were adopted in this experiment and the classification threshold T was empirically specified as $T = 30$. The average quantitative results over these videos

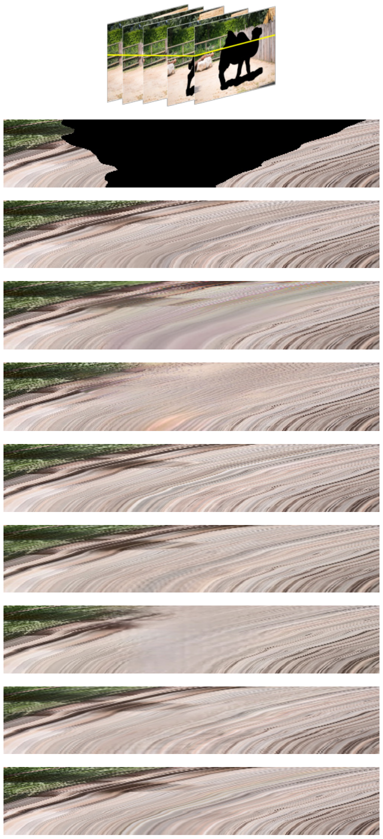


Figure 12. Comparison of temporal coherence. From top to bottom: Temporal slice, Mask, Huang [5], CP-Net [7], OPN [9], FGVC [6], IIVI [8], E2FGVI [10], FGT [11] and ours.

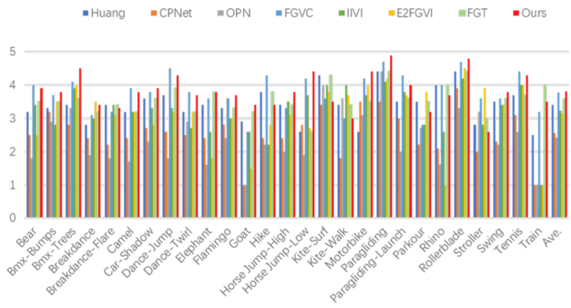


Figure 13. User study results evaluated on DAVIS videos.

are given in Table 4. According to the results presented in Table 4, we selected $n = 80$ and $\lambda = 1.0$ for our method due to the better results achieved by using such a combination.

Table 5. Parameter Determination for Progressive Fusion

d_{max}	10	20	30	40	50
PSNR	35.63	35.60	**35.92**	35.78	35.62
SSIM	0.9905	0.9896	**0.9913**	0.9889	0.9883

Table 6. Ablation Study Results of LTPI

Methods		PSNR	SSIM
Intra GOP inpainting	FPI	27.89	0.9412
	BPI	27.43	0.9383
	FPI+BPI	33.56	0.9731
Intra GOP inpainting +Inter GOP inpainting	FPI	29.63	0.9557
	BPI	29.35	0.9541
	FPI+BPI	**35.92**	**0.9913**

For progressive fusion We used different d_{max} in the experiment and chose the most applicable one to calculate the weighting factor ω for Eq. (12) used in the progressive fusion algorithm. The average quantitative results over all the broken landscape videos are given in Table 5. According to the results shown in Table 5, we finally adopt $d_{max} = 20$ in our work due to the superior performance.

Effectiveness of LTPI

We conducted ablation experiments to verify the effectiveness of our proposed LTPI module, where the performance of intra GOP inpainting and inter GOP inpainting was verified. Meanwhile, the effectiveness of FPI and BPI was also verified. These experiments were conducted on the landscape videos with the 480p resolution. The verification results are given in Table 6. It was found from Table 6 that completing videos with both the intra GOP inpainting and the inter GOP inpainting demonstrate impressive results and adopting FPI and BPI can bring significant performance gain.

Conclusions

In this paper, we propose a short-long-term propagation-based method to fill the missing areas of videos. Our proposed method was developed based upon integrating two inpainting modules, i.e., STPI and LTPI, where the STPI module fills a single frame with the reference information collected from local adjacent frames and the LTPI module inpaints the whole video by progressively applying STPI to all the frames. In both the modules, the correlated spatio-temporal information is propagated from one frame to another, guaranteeing a high temporal consistency. The experimental results, including qualitative studies with users as well as quantitative methods, using reference techniques [5-11] all demonstrate that our proposed method achieves better performance than those at the current state-of-the-art.

REFERENCES

1. S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 2682-2690. 2017.
2. Y. Li, S. Liu, J. Yang, and M. Yang, "Generative face completion," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 3911-3919. 2017.
3. S. Ge, C. Li, S. Zhao, and D. Zeng, "Occluded face recognition in the wild by identity-diversity inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol.30, no. 10, pp. 3387-3397, Jan. 2020.
4. K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2005.
5. J. Huang, S. Kang, N. Ahuja, and J. Kopf, "Temporally coherent completion of dynamic video," *ACM Trans. on Graphics*, vol. 35, no. 6, pp. 1-11, Nov. 2016.
6. C. Gao, A. Saraf, J. Huang, and J. Kopf, "Flow-edge guided video completion," in *Proc. Eur. Conf. Comput. Vis.*, pp. 713-729, 2020.
7. S. Lee, S. Oh, D. Won, and S. Kim, "Copy-and-paste networks for deep video inpainting," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2019.
8. H. Ouyang, T. Wang, and Q Chen, "Internal Video Inpainting by Implicit Long-range Propagation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 14579-14588, 2021.
9. S. W. Oh, S. Lee, J. Lee and Seon Joo Kim, "Onion-peel networks for deep video completion," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2019.
10. Z. Li, C. Z. Lu, J. Qin, C. L. Guo, and M. M. Cheng, "Towards an end-to-end framework for flow-guided video inpainting," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 17562-17571, 2022.
11. K. Zhang, J. Fu, and D. Liu. "Flow-guided transformer for video inpainting." in *Proc. Eur. Conf. Comput. Vis.*, pp. 74-90, 2022.
12. R. Ranftl, K. Lasinger, D. Hafner, K. Schindler and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, Early Access Article, DOI 10.1109/TPAMI.2020.3019967, 2020.
13. J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 679-698, 1986.
14. F. Liu, M. Gleicher, H. L. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," *ACM Trans. on Graphics*, vol. 28, no. 3, pp. 1-9, 2009.
15. H. Bay, T. Tinne, and L. V. Gool, "SURF: Speeded up robust features." in *Proc. Eur. Conf. Comput. Vis.*, May 2006.
16. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981
17. R. Achanta, S. Shaji, K. Smith, A. Lucchi, A. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274-2282, 2012.
18. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 24, no.5, pp. 603-619, 2002.
19. C. Leys, C. Ley, O. Klein, P. Bernard and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *J. Exp. Soc. Psychol.*, vol. 49, no. 4, pp. 764-766, 2013.
20. J. H. Yu, Z. Lin, J. M. Yang, X. H. Shen, X. Lu, and T. S Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2019.

Shibo Li received the B.Eng degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with both School of Information and Communication Engineering, UESTC and James Watt School of Engineering, University of Glasgow. His research interests include computer vision and wireless sensing. Contact him at 202011012230@std.uestc.edu.cn.

Shuyuan Zhu received the Ph.D. degree from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010. From 2010 to 2012, he worked at HKUST and Hong Kong Applied Science and Technology Research Institute Company Limited, respectively. In 2013, he joined University of Electronic Science and Technology of China and is currently a Professor with School of Information and Communication Engineering. Dr. Zhu's research interests include image/video compression and computer vision. He currently serves as an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology and received the Best Associate Editor Award in 2021. He received the Top 10% paper award at IEEE ICIP-2014 and the Best 10% paper award at VCIP-2016. He served as the committee member for IEEE ICME-2014, IEEE DSP-2015, VCIP-2016, PCM-2017, IEEE MIPR-2020 and IEEE ICIP-2021. Contact him at eezsy@uestc.edu.cn.

Yuzhou Huang received the B.Eng degree from the joint educational program of the University of Electronic Science and Technology of China, Chengdu, China, and the University of Glasgow, Glasgow, UK, in 2021. He is currently pursuing the Ph.D. degree in School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China. His research interests include computer vision and wireless communication. Contact him at 221019083@link.cuhk.edu.cn.

Shuaicheng Liu received the B.Eng degree from Sichuan University, Chengdu, China, in 2008, and the M.S. and Ph.D. degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. Since 2014, he has been an Associate Professor with the School of Information and Communication Engineering, Institute of Image Processing, University of Electronic Science and Technology of China. His research interests include computer vision and computer graphics. Contact him at liushuaicheng@uestc.edu.cn.

Bing Zeng received his B. Eng and M. Eng degrees from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and his Ph.D. degree from Tampere University of Technology, Tampere, Finland, in 1991. He worked as a postdoctoral fellow at University of Toronto from September 1991 to July 1992 and as a Researcher at Concordia University from August 1992 to January 1993. He then joined The Hong Kong University of Science and Technology (HKUST) and returned to UESTC in the summer of 2013. Currently, he focuses on the research of image/video processing, computer vision and image/video compression. During his tenure at HKUST and UESTC, he received about 20 research grants, filed 8 international patents, and published more than 260 papers. He received 2011 Best Paper Award of IEEE Transactions on Circuits and Systems for Video Technology (TCSVT). He served as an Associate Editor for IEEE TCSVT for 8 years and received the Best Associate Editor Award in 2011. He was elected as an IEEE Fellow in 2016 for contributions to image and video coding. Contact him at eezeng@uestc.edu.cn.

Muhammud Ali Imran Fellow IET, Fellow IEEE, Senior Fellow HEA is Dean University of Glasgow UESTC and a Professor of Wireless Communication Systems with research interests in self organised networks, wireless networked control systems, internet of things (IoT) and the wireless sensor systems. He heads the Communications, Sensing and Imaging (CSI) research group at University of Glasgow and is the Director of Glasgow-UESTC Centre for Educational Development and Innovation. He is an Affiliate Professor at the University of Oklahoma, USA and a visiting Professor at 5G Innovation Centre, University of Surrey, UK. He has over 20 years of combined academic and industry experience with several leading roles in multi-million pounds funded projects. He has filed 15 patents; has authored/co-authored over 400 journal and conference publications; has edited 7 books and authored more than 30 book chapters; has successfully supervised over 40 postgraduate students at Doctoral level. He has been a consultant to international projects and local companies in the area of self-organised networks. Contact him at Qammer.Abbasi@glasgow.ac.uk.

Qammer H. Abbasi is a Professor with the James Watt School of Engineering, University of Glasgow, U.K., deputy head for Communication Sensing and Imaging group. He has published 350+ leading international technical journal and peer reviewed conference papers and 10 books and received several recognitions for his research including URSI 2019 Young Scientist Awards, UK exceptional talent endorsement by Royal Academy of Engineering, Sensor 2021 Young Scientist Award , National talent pool award by Pakistan, International Young Scientist Award by NSFC China, National interest waiver by USA and 8 best paper awards. He is a committee member for IEEE APS Young professional, Subcommittee chair for IEEE YP Ambassador program, IEEE 1906.1.1 standard on nano communication, IEEE APS/SC WG P145, IET Antenna Propagation and healthcare network. Contact him at Muhammad.Imran@glasgow.ac.uk.

Jonathan Cooper received the Ph.D. degree in biosensor technologies from the University of Cranfield, UK. He holds The Wolfson Chair in biomedical engineering in the school of engineering, University of Glasgow. He is Emeritus Vice Principal. He has been elected as a Fellow of the Royal Academy of Engineering (UK's National Academy of Engineering) as well as a Fellow of the Royal Society of Edinburgh (Scotland's National Academy of Arts, Humanities and Sciences). Contact him at Jon.Cooper@glasgow.ac.uk.