Deep Video Inpainting

Dahun Kim* KAIST
Sanghyun Woo* KAIST
Joon-Young Lee Adobe Research
In So Kweon KAIST

Abstract

Video inpainting aims to fill spatio-temporal holes with plausible content in a video. Despite tremendous progress of deep neural networks for image inpainting, it is challenging to extend these methods to the video domain due to the additional time dimension. In this work, we propose a novel deep network architecture for fast video inpainting. Built upon an image-based encoder-decoder model, our framework is designed to collect and refine information from neighbor frames and synthesize still-unknown regions. At the same time, the output is enforced to be temporally consistent by a recurrent feedback and a temporal memory module. Compared with the state-of-the-art image inpainting algorithm, our method produces videos that are much more semantically correct and temporally smooth. In contrast to the prior video completion method which relies on time-consuming optimization, our method runs in near real-time while generating competitive video results. Finally, we applied our framework to video retargeting task, and obtain visually pleasing results.

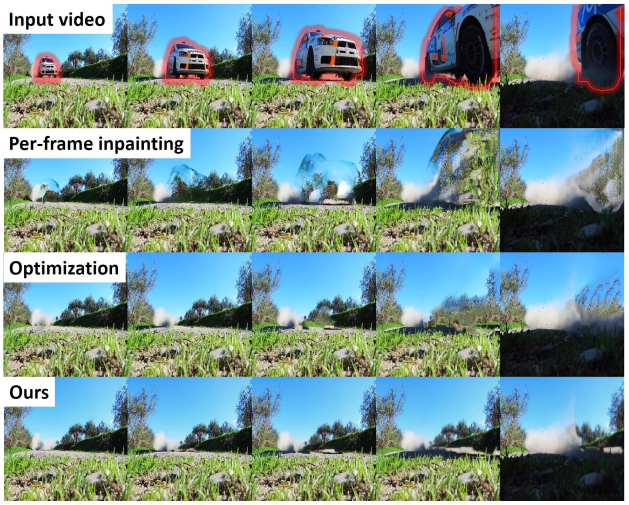


Figure 1. Input video with mask boundaries in red (row-1). Video inpainting results by per-frame image inpainting [33] (row-2), optimization-based method [11] (row-3), and our method (row-4). *Best viewed when zoomed-in*.

1. Introduction

Video inpainting can help numerous video editing and restoration tasks such as undesired object removal, scratch or damage restoration, and retargeting. More importatnly, and apart from its converntional demands, video inpainting can be used in combination with Augmented Reality (AR) for a greater visual experience; Removing existing items gives more opportunities before overlaying new elements in a scene. Therefore, as a Diminished Reality (DR) technology, it opens up new opportunities to be *paired with recent real-time / deep learning-based AR technologies*. Moreover, there are several semi-online streaming scenarios such as automatic content filtering and visual privacy filtering. Only a small wait will lead to a considerable latency, thus making the speed itself an important issue.

Despite tremendous progress on deep learning-based inpainting of a single image, it is still challenging to extend these methods to video domain due to the additional time dimension. The difficulties coming from complex motions and high requirement on temporal consistency make video inpainting a challenging problem. A straightforward way to perform video inpainting is to apply image inpainting on each frame individually. However, this ignores motion regularities coming from the video dynamics, and is thus incapable of estimating non-trivial appearance changes in image-space over time. Moreover, this scheme inevitably brings temporal inconsistencies and causes severe flickering artifacts. The second row in Fig. 1 shows an example of directly applying the state-of-the-art feed-forward image inpainting [33] in a frame-by-frame manner.

To address the temporal consistency, several methods have been developed to fill in the missing motion fields; using a greedy selection of local spatio-temporal patches [24], a per-frame diffusion-based technique [16], or an iterative optimization [11]. However, the first two methods treat flow estimation to be independent of color estimation [16, 24] and the last relies on time-consuming optimization [11] (3rd row in Fig. 1), which is effective but limits their practicality and flexibility in general scenarios.

*Both authors have contributed equally to this work.

One might attempt to maintain temporal consistency by applying a post-processing method. Recently, Lai *et al*. [14] proposed a deep CNN model that takes both original and per-frame processed videos as input and produces a temporally consistent video. However, their method is only applicable when those two input videos have a pixel-wise correspondences (*e.g.*colorization), which is not the case for video inpainting.

In this paper, we investigate whether a feed-forward deep network can be adapted to the video inpainting task. Specifically, we attempt to train a model with two core functions: 1) temporal feature aggregation and 2) temporal consistency preserving. For the **temporal feature aggregation**, we cast the video inpainting task as a sequential multi-to-single frame inpainting problem. In particular, we introduce a novel 3D-2D feed-forward network which is built upon a 2D-based (image based) encoder-decoder model. The network is designed to collect and refine potential hints from neighbor frames and synthesize semantically-coherent video content in space and time. For the **temporal consistency**, we propose to use a recurrent feedback and a memory layer (*e.g.* convoutional LSTM [28]). In addition, we use a flow loss to learn a warping of the previously synthesized frame and a warping loss to enforce both short-term and long-term consistency in results. Finally, we come up with a single, unified deep CNN model called **VINet**.

We conduct extensive experiments to validate the contributions of our design choices. We show that our multi-to-single frame formulation produces videos that are much more accurate and visually pleasing than the method of [33]. An example result of our method is shown in the last row of Fig. 1. Our model sequentially processes video frames of arbitrary length and requires no optical flow computation at the test time, thus runs at a near real-time rate.

Contribution. In summary, our contribution is as follow.

1. We cast video inpainting as a sequential multi-to-single frame inpainting task and present a novel deep 3D-2D encoder-decoder network. Our method effectively gathers features from neighbor frames and synthesizes missing content based on them.

2. We use a recurrent feedback and a memory layer for the temporal stability. Along with the effective network design, we enforce strong temporal consistency via two losses: flow loss and warping loss.

3. Up to our knowledge, it is the first work to provide a single, unified deep network for the general video inpainting task. We conduct extensive subjective and objective evaluations and show its efficacy. Moreover, we apply our method to video retargeting and super-resolution tasks, demonstrating favorable results.

2. Related Work

Significant progress has been made on image inpainting [1,3,8,12,15,18,30–33], to a point of where commercial solutions are now available [2]. However, video inpainting algorithms have been under-investigated. This is due to the additional time dimension which introduces major challenges such as severe viewpoint changes, temporal consistency preserving, and high computational complexity. Most recent methods found in the literature address these issues using either object-based or patch-based approaches.

In object-based methods, a pre-processing is required to split a video into foreground objects and background, and it is followed by an independent reconstruction and merging step at the end of algorithms. Previous efforts which fall under this category are homography-based algorithms that are based on the graph-cut [9,10]. However, the major limitation of these object-based methods is that the synthesized content has to be copied from the visible regions. Therefore, these methods are mostly vulnerable to abrupt appearance changes such as scale variations, *e.g.* when an object moves away from the camera.

In patch-based methods, the patches from known regions are used to fill in a mask region. For example, Patwardhan *et al*. [19, 20] extend the well-known texture synthesis technique [8] to video inpainting. However, these methods either assume static cameras [19] or constrained camera motion [20] and are based on a greedy patch-filling process where the early errors are inevitably propagated, yielding globally inconsistent outputs.

To ensure the global consistency, patch-based algorithms have been cast as a global optimization problem. Wexler *et al*. [27] present a method that optimizes a global energy minimization problem for 3D spatio-temporal patches by alternating between patch search and reconstruction steps. Newson *et al*. [17] extend this by developing a spatio-temporal version of PatchMatch [2] to strengthen the temporal coherence and speed up the patch matching. Recently, Huang *et al*. [11] modify the energy term of [27] by adding an optical flow term to enforce temporal consistency. Although these methods are effective, their biggest limitations are high computational complexity and the absolute dependence upon the pre-computed optical flow which cannot be guaranteed to be accurate in complex sequences.

To tackle these issues, we propose a deep learning based method for video inpainting. To better exploit temporal information coming from multiple frames and be highly efficient, we construct a 3D-2D encoder-decoder model, that can provide traceable features revealed from the video dynamics. It takes total 6 frames as input; 5 source frames and 1 reference frame (*i.e.*the frame to be inpainted). We learn the feature flow between frames to deal with both hole-filling and coherence. The still-unknown regions are synthesized in a semantically natural way based on the sur-

rounding context. We argue that our method provides a better prospect than the previous optimization-based techniques in that deep CNNs are excellent at learning spatial semantics and temporal dynamics from an ever-growing vast amount of video data. To our best knowledge, this is the first work that deeply addresses the general video inpainting problem via a deep CNN model.

3. Method

3.1. Problem Formulation

Video inpainting aims to fill in arbitrary missing regions in video frames $X_1^T := \{X_1, X_2, ..., X_T\}$. The reconstructed regions should be either accurate as in the ground-truth frames $Y_1^T := \{Y_1, Y_2, ..., Y_T\}$ and consistent in space and time. We formulate the video inpainting problem as learning a mapping function from X_1^T to the output $\hat{Y}_1^T := \{\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_T\}$ such that the conditional distribution $p(\hat{Y}_1^T|X_1^T)$ is identical to $p(Y_1^T|X_1^T)$. Through matching the conditional distributions, the network learns to generate realistic and temporally-consistent output sequences. To simplify the problem, we make a Markov assumption where we factorize the conditional distribution to a product form. In this form, the naive *frame-by-frame* inpainting can be formulated as

$$p(\hat{Y_1^T}|X_1^T) = \prod_{t=1}^{T} p(\hat{Y}_t|X_t). \quad (1)$$

However, to obtain visually pleasing video results, we argue that the generation of t-th frame $\hat{Y}_t$ should be consistent with 1) spatio-temporal neighbor frames X_{t-N}^{t+N} where N denotes a temporal radius, 2) the previously generated frame $\hat{Y}_{t-1}$ and 3) all previous history encoded in a recurrent memory M_t. Thus, we propose to learn the conditional distribution of

$$p(\hat{Y_1^T}|X_1^T) = \prod_{t=1}^{T} p(\hat{Y}_t|X_{t-N}^{t+N}, \hat{Y}_{t-1}, M_t). \quad (2)$$

In our experiments, we set N to 2, taking two lagging and two leading frames to recover the current frame. We sample frames with a temporal stride 3, such that $X_{t-N}^{t+N} := \{X_{t-6}, X_{t-3}, X_t, X_{t+3}, X_{t+6}\}$. We want to recover the current frame by both aggregating information from neighbor frames and synthesizing totally blind regions jointly. At the same time, the output is enforced to be temporally consistent with the past predictions by the recurrent feedback ($\hat{Y}_{t-1}$) and the memory (M_t). We train a deep network D to model the conditional distribution $p(\hat{Y}_t|X_{t-N}^{t+N}, \hat{Y}_{t-1}, M_t)$ as $\hat{Y}_t = D(X_{t-N}^{t+N}, \hat{Y}_{t-1}, M_t)$. We obtain the final output $\hat{Y}_1^T$ by applying the function D in an autoregressive manner. Our multi-to-single frame formulation outperforms a single-frame baseline and even produces results comparable with the optimization-based method, as described in Sec. 4.

3.2. Network Design

Our full model (VINet) jointly learns to inpaint the video and maintain temporal consistency. The overview of VINet is illustrated in Fig. 2.

3.2.1 Multi-to-Single Frame Video Inpainting

In videos, the occluded or removed parts in a frame are often revealed in the past/future frames as the objects move and the viewpoint changes. If such hints exist in the temporal radius, those disclosed content can be borrowed to recover the current frame. Otherwise, the still-unknown regions should be synthesized. To achieve this, we construct our model as an encoder-decoder network that learns such temporal feature aggregation and single-frame inpainting simultaneously. The network is designed to be fully convolutional, which can handle arbitrary size input.

Source and reference encoders. The encoder is a multiple-tower network with source and reference streams. The source stream takes past and future frames with the inpainting masks as input. For the reference stream, the current frame and its inpainting mask are provided. We concatenate the image frames and the masks along the channel axis, and feed into the encoder. In practice, we use a 6-tower encoder: 5 source streams with weight-sharing that take two lagging (X_{t-6}, X_{t-3}) and two leading frames (X_{t+3}, X_{t+6}), and the previously generated frame ($\hat{Y}_{t-1}$), and 1 reference stream. The source features that are non-overlapping with the reference features can be borrowed to inpaint the missing regions by the following feature flow learning and learnable feature composition.

Feature flow learning. Before directly combining the source and reference features, we propose to explicitly align the feature points. This strategy helps our model easily borrow traceable features from the neighbor frames. To achieve this, we insert flow sub-networks to estimate the flows between the source and reference feature maps in four different spatial scales (1/8, 1/4, 1/2, and 1). We adopt the coarse-to-fine structure of PWCNet [25]. The explicit flow supervision is only given at the finest scale (*i.e.* 1) and *only between* the consecutive two frames, where we extract the pseudo-ground-truth flow $W_{t \Rightarrow t-1}$ between Y_t and Y_{t-1} using FlowNet2 [13].

Learnable Feature Composition. Given the aligned feature maps from the five source streams, they are concatenated along the time dimension and fed into a $5 \times 3 \times 3$ (THW) convolution layer that produces a spatio-temporally aggregated feature map $F_{s'}$ with the time dimension of 1. This is designed to dynamically select source feature points across the time axis, by highlighting the features complementary to the reference features and ignoring otherwise. For each 4 scales, we employ a mask sub-network to com-

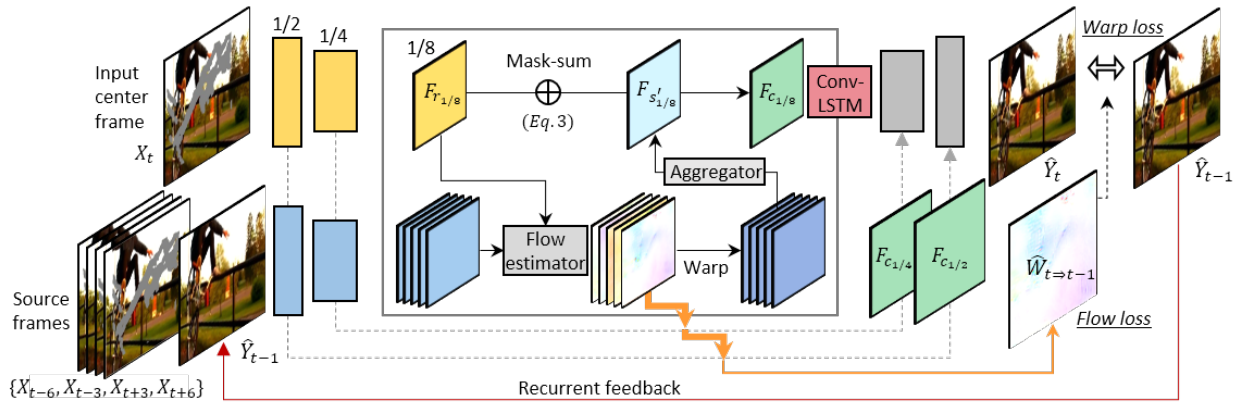


Figure 2. **The overview of VINet.** Our network takes in multiple frames (X_{t-6}, X_{t-3}, X_t, X_{t+3}, X_{t+6}) and the previously generated frame ($\hat{Y}_{t-1}$), and generates the inpainted frame ($\hat{Y}_t$) as well as the flow map ($\hat{W}_{t\Rightarrow t-1}$). We employ both flow sub-networks and mask sub-networks at 4 scales (1/8, 1/4, 1/2, and 1) to aggregate and synthesize feature points progressively. For temporal consistency, we use a recurrent feedback and a temporal memory layer (ConvLSTM) along with two losses: flow loss and warp loss. The orange arrows denote the $\times 2$ upsampling for residual flow learning as in [25] for 5 streams, while the thinner orange arrow is for only the stream from $\hat{Y}_{t-1}$. The mask sub-networks are omitted in the figure for the simplicity.

bine the aggregated feature map $F_{s'}$ with the reference feature map F_r. The mask sub-network consists of three convolution layers and takes the absolute difference of the two feature maps $|F_{s'} - F_r|$ as input and produces single channel composition mask m, as suggested in [6]. By using the mask, we can gradually combine the warped features and the reference features. At the scale of 1/8, the composition is done by

$$F_{c_{1/8}} = (1 - m_{1/8}) \odot F_{r_{1/8}} + m_{1/8} \odot F_{s'_{1/8}}, \quad (3)$$

where $\odot$ is the element-wise product operator.

Decoder. To pass image details to the decoder, we employ skip connections as in U-net [23]. To prevent the concern raised by [32] that skip connections contain zero values at the masked region, our skip-connections pass the composite features similarly to Eq. (3), as

$$F_{c_{1/4}} = (1 - m_{1/4}) \odot F_{r_{1/4}} + m_{1/4} \odot F_{s'_{1/4}}, \quad (4)$$

$$F_{c_{1/2}} = (1 - m_{1/2}) \odot F_{r_{1/2}} + m_{1/2} \odot F_{s'_{1/2}}. \quad (5)$$

At the finest scale, the estimated optical flow $\hat{W}_{t\Rightarrow t-1}$ is used to warp the previous output $\hat{Y}_{t-1}$ to the current raw output $\hat{Y'}_t$. We then blend this warped image and the raw output with the composition mask m_1, to obtain our final output $\hat{Y}_t$ as

$$\hat{Y}_t = (1 - m_1) \odot \hat{Y'}_t + m_1 \odot \hat{W}_{t\Rightarrow t-1}(\hat{Y}_{t-1}). \quad (6)$$

3.2.2 Recurrence and Memory

To strongly enforce the temporal coherence on the video output, we propose to use the recurrent feedback loop ($\hat{Y}_{t-1}$) and the temporal memory layer (M_t) as formulated in Eq. (2).

Our formulation encourages the current output to be conditional to the previous output frame. The knowledge from the previous output encourages the traceable features to be kept unchanged, while the untraceable (*e.g.* occlusion) points to be synthesized. This not only helps the output to be consistent along the motion trajectories but also avoids ghosting artifacts at occlusions or motion discontinuities.

While the recurrent feedback connects the consecutive frames, filling in the large holes requires more long-term (*e.g.* 5 frames) knowledge. At this point, the temporal memory layer can help to connect internal features from different time steps in the long term. We adopt a convolutional LSTM (ConvLSTM) layer and a warping loss as suggested in [14]. In particular, we feed the composite feature F_c at the scale 1/8 to the ConvLSTM at every time step.

3.3. Losses

We train our network to minimize the following loss function,

$$\mathcal{L} = \lambda_R \mathcal{L}_R + \lambda_F \mathcal{L}_F + \lambda_W \mathcal{L}_W, \quad (7)$$

where $\mathcal{L}_R$ is the reconstruction loss, $\mathcal{L}_F$ is the flow estimation loss, and $\mathcal{L}_W$ is the warping loss. The balancing weights $\lambda_R, \lambda_F, \lambda_W$ are set to 1, 10, 1 respectively throughout the experiments. For the temporal losses $\mathcal{L}_F$ and $\mathcal{L}_W$, we set the number of recurrences as 5 ($T = 5$).

$\mathcal{L}_R$ consists of two terms, $\mathcal{L}_1$ and $\mathcal{L}_{ssim}$,

$$\mathcal{L}_1 = \left\| \hat{Y}_t - Y_t \right\|_1, \tag{8}$$

$$\mathcal{L}_{ssim} = \left(\frac{(2\mu_{\hat{Y}_t}\mu_{Y_t} + c_1)(2\sigma_{\hat{Y}_t Y_t} + c_2)}{(\mu^2_{\hat{Y}_t} + \mu^2_{Y_t} + c_1)(\sigma^2_{\hat{Y}_t} + \sigma^2_{Y_t} + c_2)}\right), \tag{9}$$

$$\mathcal{L}_R = \mathcal{L}_1 + \mathcal{L}_{ssim}, \tag{10}$$

where $\hat{Y}_t, Y_t$ denote the predicted frame and the ground-truth frame respectively. μ, σ denote the average, variance, respectively. c_1, c_2 denote the stabilization constants which are respectively set to $0.01^2, 0.03^2$.

The flow loss $\mathcal{L}_F$ is defined as

$$\sum_{t=2}^{T} \left(\left\| W_{t\Rightarrow t-1} - \hat{W}_{t\Rightarrow t-1} \right\|_1 + \left\| Y_t - \hat{W}_{t\Rightarrow t-1}(Y_{t-1}) \right\|_1 \right), \tag{11}$$

where $W_{t\Rightarrow t-1}$ is the pseudo-ground-truth backward flow between the target frames, Y_t and Y_{t-1}, extracted by FlowNet2 [13]. In Eq. (11), the first term is the endpoint error between the groundturth and the estimated flow, and the second is the warping error when the flow is used to warp the previous target frame to the next target frame.

The warping loss $\mathcal{L}_W$ includes $\mathcal{L}_{st}$ and $\mathcal{L}_{lt}$ as,

$$\mathcal{L}_{st} = \sum_{t=2}^{T} M_{t\Rightarrow t-1} \left\| \hat{Y}_t - W_{t\Rightarrow t-1}(Y_{t-1}) \right\|_1, \tag{12}$$

$$\mathcal{L}_{lt} = \sum_{t=2}^{T} M_{t\Rightarrow 1} \left\| \hat{Y}_t - W_{t\Rightarrow 1}(Y_1) \right\|_1, \tag{13}$$

$$\mathcal{L}_W = \mathcal{L}_{st} + \mathcal{L}_{lt}. \tag{14}$$

We follow the protocol in [14] that uses FlowNet2 [13] to obtain $M_{t\Rightarrow t-1}$ and W_{t-1}, which respectively denote the binary occlusion mask and the backward optical flow between the target frames Y_t and Y_{t-1}. We adopt both short-term and long-term temporal losses. Note that we use ground-truth target frames in the warping operation since the synthesizing ability is imperfect during training.

3.4. Two-Stage Training

We employ a two-stage training scheme that gradually learns the core functionalities for video inpainting; 1) We first train the model without the recurrent feedback and memory to focus on learning the temporal feature aggregation. At this stage, we only use the reconstruction loss $\mathcal{L}_R$; 2) We then add the recurrent feedback and the ConvLSTM layer, and fine-tune the model using the full loss function (Eq. (7)) for temporally coherent predictions. We use videos in the Youtube-VOS dataset [29] as ground-truth for the training. It is a large-scale dataset for video object segmentation containing 4000+ YouTube videos with 70+ common objects. All video frames are resized to 256×256 pixels for training and testing.

Video mask dataset. In general video inpainting, the spatio-temporal holes consist in diverse motion and shape changes. To simulate this complexity during training, we create the following four types of video masks.

1. Random square: We randomly mask a square box in each frame. The visible regions each of input frames are mostly complementary so that the network can clearly learn how to align, copy, and paste neighboring feature points.
2. Flying square: The motion of the inpainting holes is rather regularized than random in real scenarios. To simulate such regularity, we shift a square by a uniform step size in one direction across the input frames.
3. Arbitrary mask: To simulate diverse hole shapes and sizes, we use the irregular mask dataset [15] which consists of random streaks and holes of arbitrary shapes. During training, we apply random transformations (translation, rotation, scaling, sheering).
4. Video object mask: In the context of the video object removal task, masks with the most realistic appearance and motion can be obtained from video object segmentation datasets. We use the foreground segmentation masks of the YouTube-VOS dataset [29].

3.5. Inference

We assume that the inpainting masks for all video frames are given. To avoid any data overlap between training and testing, we obtain object masks from the DAVIS dataset [21, 22], the public benchmark dataset for video object segmentation. It contains dynamic scenes, complex camera movements, motion blur effects, and large occlusions. The inpainting mask is constructed by dilating the ground-truth segmentation mask. Our method processes frames recursively in a sliding window manner.

3.6. Implementation Details

Our model is implemented using Pytorch v0.4, CUDNN v7.0, CUDA v9.0. It run on the hardware with Intel(R) Xeon(R) (2.10GHz) CPU and NVIDIA GTX 1080 Ti GPU. The model runs at 12.5 fps on a GPU for frames of 256×256 pixels. We use Adam optimizer with β = (0.9, 0.999) and a fixed learning rate 1e-4. We train our model from scatch. The first and second training stage takes about 1 day each using four NVIDIA GTX 1080 Ti GPUs.

4. Experiments

In this section, we conduct experiments to analyze our two major design choices. Specifically, we visualize the learned multi-to-single mechanism and show the impact of the added recurrence and memory. We then evaluate our video results both quantitatively and qualitatively,

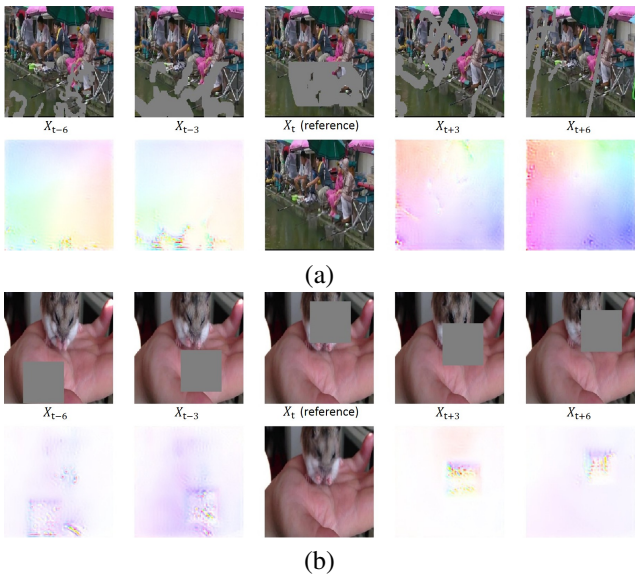


Figure 3. **Visualization of the learned feature composition.** Input frames are on the odd rows, and corresponding feature flows referential to the center, and the inpainted frame are on the even rows. Our network successfully aligns and integrates the source features to fill in the large and complex hole in the reference frame.

compared with the state-of-the-art baselines. Finally, we demonstrate the applicability of our framework on video retargeting and video super-resolution tasks.

Baselines. We compare our approach to two state-of-the-art baselines in the literature by running their test codes with our testing videos and masks.

- Yu *et al*. [33]: A feed-forward CNN based method, which is designed for single image inpainting. We processes videos frame-by-frame without using any temporal information.
- Huang *et al*. [11]: An optimization-based video completion method, which jointly estimates global flow and color. It requires on-the-fly optical flow computation and is extremely time-consuming.

4.1. Visualization of Learned Feature Composition

Fig. 3 shows that the proposed model explicitly borrows visible neighbor features to synthesize the missing content. For the visualization, we take the model of the first training stage and plot the learned feature flow from each of the four source streams to the reference stream, at 128×128 pixel resolution. We observe that even with a large and complex hole in the reference (center) frame, our network is able to align the source feature maps to the reference and integrate them to fill in the hole. Even without an explicit flow supervision, our flow sub-network is able to warp the feature points in visible regions and shrink the unhelpful zero features in masked regions. Moreover, these potential hints are adjusted according to the spatio-temporal semantics, rather than copied-and-pasted in a fixed manner. One example is shown in Fig. 3-(b) where the eyes of the hamster are synthesized *half-closed*.

	DAVIS masks on Sintel frames
Frame-by-frame [33]	0.0429
Optimization [11]	0.0343
VINet (agg. only)	0.0383
VINet (agg. + T.C.)	**0.0015**

Table 1. **Flow warping errors.** We evaluate the flow warping errors on the Sintel dataset using 21 videos and ground truth flows.

	DAVIS masks on DAVIS frames
Frame-by-frame [33]	0.0080
Optimization [11]	0.0053
VINet (agg. only)	0.0073
VINet (agg. + T.C.)	**0.0046**

Table 2. **FID scores.** We evaluate the FID scores on the DAVIS dataset using 20 videos.

4.2. Improvement on Temporal Consistency

We compare the temporal consistencies of our video results before and after adding the recurrent feedback and the convLSTM. To validate the effectiveness of our method, we also compare with the two representative baselines mentioned above [11, 33]. Since the Sintel dataset [4] provides ground-truth optical flows, we use it to quantitatively measure the *flow warping errors* [14]. We use the object masks in the DAVIS dataset [21, 22] as our inpainting mask sequences. We take 32 frames each from 21 videos in Sintel to constitute our inputs and experiment for five trials. For each trial, we randomly select 21 videos of length 32+ from DAVIS to create corresponding mask sequences and keep them unchanged for all the methods.

In Table. 1, we report the flow warping errors averaged over the videos and trials. It shows that our full model outperforms other baselines by large margins. Even the global (heavy) optimization method [11] performs marginally better than our 1st-stage method and has a much larger error than our full model. Not surprisingly, Yu *et al*.'s method turns out to be the least temporally consistent. Note that the error of our full model is reduced by a factor of 10 after adding the recurrent feedback and the convLSTM layer, implying that they significantly improve the temporal stability in the short and long term.

4.3. Spatio-Temporal Video Quality

Wang *et al*. [26] proposed a video version of the inception score (FID) to quantitatively evaluate the quality of video generation. We take this metric to evaluate the quality of video inpainting as it measures the spatio-temporal

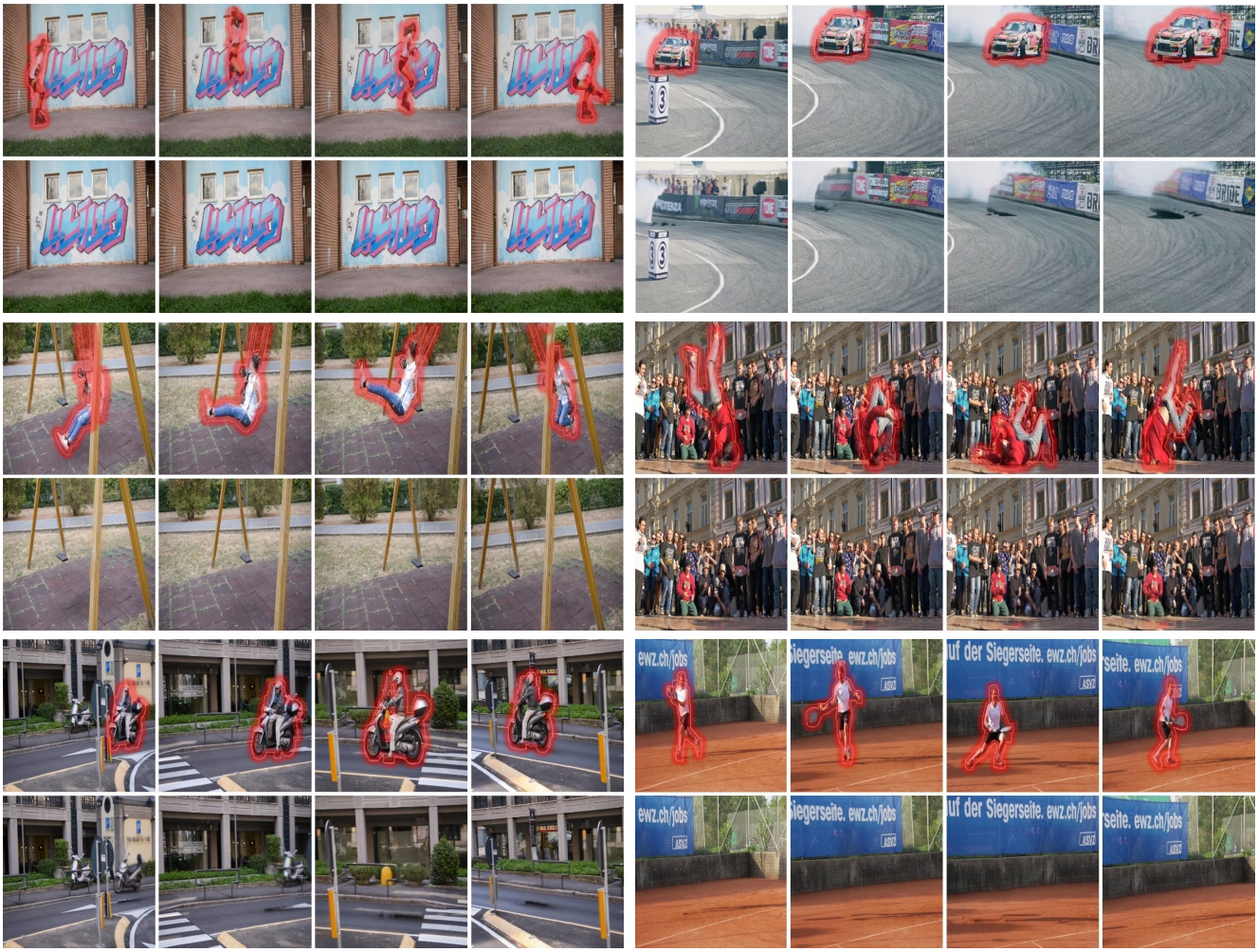


Figure 4. **Object removal from DAVIS video sequences.** For each input sequence, we show representative frames with mask boundaries in red. We show the inpainted results using our method in even rows.

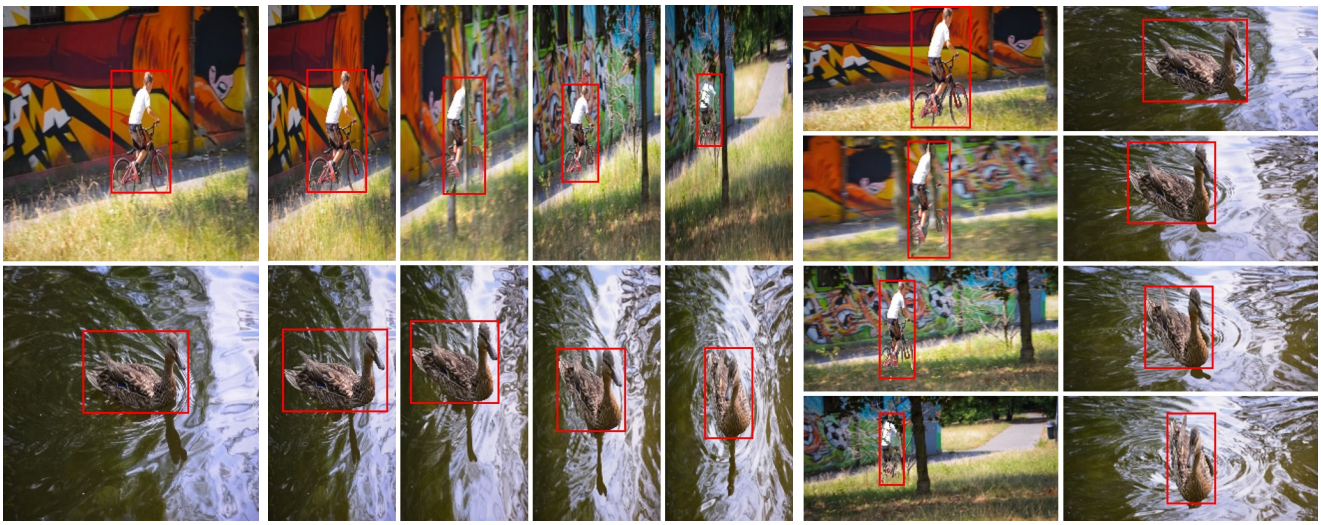
quality in a perceptual level. As in [26], we follow the protocol that uses the I3D network [5] pretrained on a video recognition task to measure the distance between the spatio-temporal features extracted from the output videos and the ground-truth videos.

For this experiment, we take 20 videos in the DAVIS dataset. For each video, we ensure to choose a different video out of the other 19 videos to make a mask sequence, so that we have the setting where our algorithm is supposed to recover the original videos rather than remove any parts. We use the first 64 frames for both input and mask videos. We run five trials as in Sec. 4.2 and average the FID scores over the videos and trials. Table. 2 summarizes the results. Our method has the smallest FID among the compared methods. This implies that our method achieves both better visual quality and temporal consistency.

4.4. User Study on Video Object Removal

We apply our approach to remove dynamically moving objects in videos. We use 24 videos from the DAVIS dataset [21, 22] of which the names are listed in Fig. 6. Examples of our results are in Fig. 4. We perform a human subjective test for evaluating the visual quality of inpainted videos. We compare our method with the strong optimization baseline [11] which is specifically aimed for the video completion task.

In each testing case, we show the original input video, our removal result and the result of Huang *et al*. on the same screen. The order of the two removal video results is shuffled. To ensure that a user has enough time to distinguish the difference and make a careful judge, we play all the video results once at the original speed and then once at $0.5\times$ speed. Also, a user allows seeing videos multiple times. Each participant is asked to choose a preferred



(a) First input frame (b) Horizontally shrunk frames (c) Vertically shrunk frames

Figure 5. **Extension to video retargeting.** (a) Original first frame. (b) Horizontally shrunk frames. (c) Vertically shrunk frames.

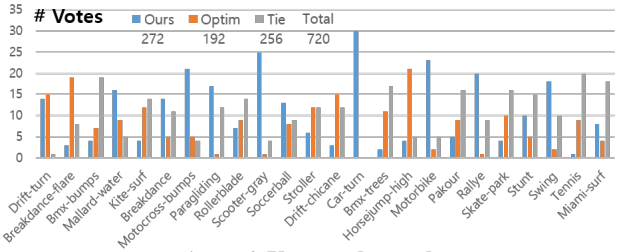


Figure 6. **User study results**.

result or tie. A total of 30 users participated in this study. We specifically ask each participant to check for both image quality and temporal consistency. The user study results are summarized in Fig. 6. It shows that, while there are different preferences across video samples, our method is preferred more often by the participants.

4.5. Application to Video Retargeting

Video retargeting aims to adjust the aspect ratio (or size) of frames to fit the target aspect ratio while maintaining salient content in a video. We propose to solve video retargeting by *removing and then adding*, which is a potential pipeline where our framework would run in combination with other AR (*i.e.*overlaying) technologies. Specifically, we first remove the salient content by inpainting the background, resize the inpainted frames into the target aspect ratio, and then overlay the salient content after the desirable rescaling. To simplify the settings, we target to horizontally or vertically shrink the frames while keeping the original aspect ratio of the moving object. The saliency masks can be automatically estimated, for example, by a feed-forward CNN [7], however we assume a more constrained scenario where the saliency masks are given as the object segmentation masks for all frames. Our method yields little warble and jittering over time and produces natural video sequences. Fig. 5 shows examples of the retargeted frames.

4.6. Limitation

We observe color saturation artifacts when there is a large and long occlusion in a video. The discrepancy error of the synthesized color propagates over time, causing inaccurate warping. The regions that have not been revealed in the temporal radius is synthesized blurry. Also, due to the limited memory footprint, we only experimented with 256×256 px frames.

5. Conclusion

In this paper, we propose a novel framework for video inpainting. Based on the multi-to-single encoder-decoder network, our model learns to aggregate and align the feature maps from neighbor frames to inpaint videos. We use the recurrent feedback and the temporal memory to encourage temporally coherent output. Our extensive experiments demonstrate that our method achieves superior visual quality than the state-of-the-art image inpainting solution and performs favorably against an optimization method both qualitatively and quantitatively. Despite some limitations, we argue that a well-posed feed-forward network has a great potential to avoid computation-heavy optimization method and boosts its applicability in many related vision tasks.

Acknowledgements Dahun Kim was partially supported by Global Ph.D. Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018H1A2A1062075).

References

[1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001. 2

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24, 2009. 2

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000. 2

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012. 6

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017. 7

[6] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua. Coherent online video style transfer. In *Proc. Intl. Conf. Computer Vision (ICCV)*, 2017. 4

[7] D. Cho, J. Park, T.-H. Oh, Y.-W. Tai, and I. S. Kweon. Weakly-and self-supervised learning for content-aware deep image retargeting. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4568–4577. IEEE, 2017. 8

[8] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *iccv*, page 1033. IEEE, 1999. 2

[9] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. In *European Conference on Computer Vision*, pages 682–695. Springer, 2012. 2

[10] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt. How not to be seenobject removal from videos of crowded scenes. In *Computer Graphics Forum*, volume 31, pages 219–228. Wiley Online Library, 2012. 2

[11] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)*, 35(6):196, 2016. 1, 2, 6, 7

[12] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 2

[13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. 3, 5

[14] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang. Learning blind video temporal consistency. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2018. 2, 4, 5, 6

[15] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018. 2, 5

[16] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on pattern analysis and Machine Intelligence*, 28(7):1150–1163, 2006. 1

[17] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014. 2

[18] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2

[19] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–69. IEEE, 2005. 2

[20] K. A. Patwardhan, G. Sapiro, and M. Bertalmío. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007. 2

[21] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 5, 6, 7

[22] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 5, 6, 7

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[24] T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang. Video completion by motion field transfer. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 411–418. IEEE, 2006. 1

[25] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 3, 4

[26] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 6, 7

[27] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *null*, pages 120–127. IEEE, 2004. 2

[28] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2

[29] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. *arXiv preprint arXiv:1809.00461*, 2018. 5

[30] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, volume 1, page 3, 2017. 2

[31] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017. 2

[32] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2, 4

[33] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention, 2018. 1, 2, 6