Proposal-based Video Completion

Yuan-Ting Hu[1], Heng Wang[2], Nicolas Ballas[3],
Kristen Grauman[3,4], and Alexander G. Schwing[1]

[1]University of Illinois Urbana-Champaign [2]Facebook AI
[3]Facebook AI Research [4]University of Texas at Austin

Abstract. Video inpainting is an important technique for a wide variety of applications from video content editing to video restoration. Early approaches follow image inpainting paradigms, but are challenged by complex camera motion and non-rigid deformations. To address these challenges flow-guided propagation techniques have been proposed. However, computation of flow is non-trivial for unobserved regions and propagation across a whole video sequence is computationally demanding. In contrast, in this paper, we propose a video inpainting algorithm based on proposals: we use 3D convolutions to obtain an initial inpainting estimate which is subsequently refined by fusing a generated set of proposals. Different from existing approaches for video inpainting, and inspired by well-explored mechanisms for object detection, we argue that proposals provide a rich source of information that permits combining similarly looking patches that may be spatially and temporally far from the region to be inpainted. We validate the effectiveness of our method on the challenging YouTube VOS and DAVIS datasets using different settings and demonstrate results outperforming state-of-the-art on standard metrics.

1 Introduction

Inpainting missing regions in a given image or video is a longstanding and important computer vision task with applications, *e.g.*, in image/video restoration. Not surprisingly, a significant amount of work has been devoted, particularly to image inpainting [6, 5, 4, 11, 1, 16, 15, 21, 40, 30, 37, 41], while leveraging temporal coherence for video completion has become increasingly popular more recently [9, 10, 24, 31, 14, 17, 39, 33, 3, 18, 42, 20, 2, 25].

Early video completion methods follow classical image inpainting techniques: missing regions are completed one frame at a time by finding patches which match or, more recently, via deep nets applied independently per frame. These methods are challenged by complex camera motion, non-rigid object deformations, motion blur, and the fact that retrieving a compelling patch is often computationally expensive. Moreover, in those methods, temporal artifacts occur if deep nets are applied independently per frame. To address those concerns, very recently, an optical flow guided propagation method has been demonstrated successfully [39]. However, computation of flow is non-trivial in unobserved regions. Besides, propagation across a whole video sequence is computationally demanding, preventing application of such methods on hardware with limited resources,

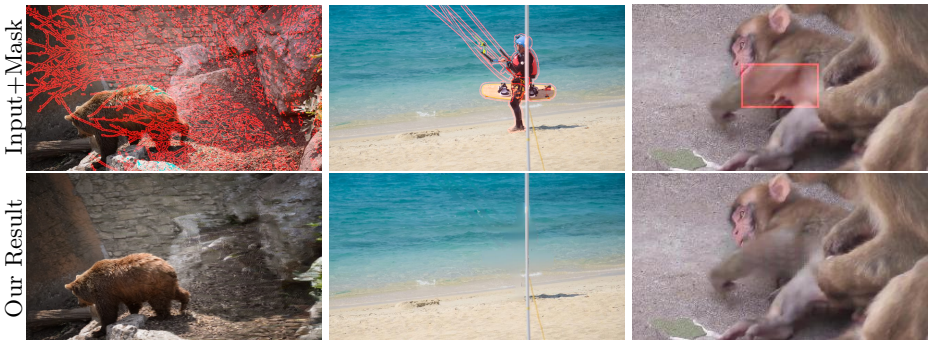


Fig. 1. Video inpainting in challenging scenarios, such as complex motion, cluttered background and large missing regions. We highlight the missing region in red in the first row and show the results of our method in the second row. We consider three different scenarios: (left) arbitrary region inpainting; (middle) object removal; (right) fixed region inpainting.

like mobile devices. Moreover, even with optical flow available, we are generally not able to infer parts missing due to occlusions.

Importantly, all the aforementioned methods directly infer an inpainted result. While direct inference is conceptually straightforward to implement via deep nets, it emphasizes local spatial and temporal consistency over more global context, as filters in deep nets often have a limited receptive field. To counter this bias, here, inspired by the recent success of proposal based techniques for object detection [12, 28], we develop a proposal-based approach.

Concretely, our suggested method first infers a coarse-grained inpainting. This inpainting is subsequently refined by constructing a set of proposals for each frame. Global spatio-temporal consistency is encouraged as proposals are contiguous regions which are fused via a parametric mechanism. The proposals are obtained via a top-k matching of (1) features of observed pixels with (2) features of the coarse-grained inpainting for the missing pixels. Different from existing work, this permits to effectively combine non-local (spatially and temporally) cues and leads to appealing results illustrated in Fig. 1.

To compare with existing methods [41, 39, 25, 20, 18], we provide extensive experiments on the challenging YouTube VOS [38] and DAVIS [26] datasets using two settings: fixed region inpainting and moving object removal. We demonstrate that our proposal-based approach achieves more accurate results than state-of-the-art methods. Moreover, it is over 55$\times$ faster than FGI [39] at inference time (0.69 *vs*. 37.63 seconds per frame) on the DAVIS dataset on fixed region inpainting, as it does not rely on optical flow-based propagation.

2 Related Work

Inpainting in images and video data has been a long standing problem in computer vision. In this section, we describe works most related to our method.

Image Inpainting. Recent efforts in image inpainting have shifted to designing of deep neural nets which fill holes of arbitrary shapes (free-from inpainting). Among them, partial convolution [21] and gated convolution [41] go beyond the standard convolution operator, and are proposed to better utilize the binary mask during convolution for inpainting. Besides operator level inventions, Edge-connect [23] first hallucinates edges for the missing regions, then uses the edges to preserve object boundaries in inpainted results. Similarly, StructureFlow [29] first recovers an edge-preserved smoothed version of the original image, then synthesizes texture for the smoothed regions.

DeepFill [40] proposes to match features between missing and known regions, and reconstructs missing pixels using the similarity scores from feature matching with known pixels. Iizuka *et al*. [15] fuse both global context and local texture information by training different discriminators for generative adversarial networks [8].

Different from the aforementioned works, our approach focuses on video inpainting, and leverages both spatial and temporal cues. Inspired by [29] we design a novel multiple stage framework for video inpainting. We first infer an initial coarse grained version, then refine the initial result based on proposals. The proposal component in our method is related to PatchMatch [1], extending it to be learning-based and end-to-end trainable.

Video Inpainting. Unlike image inpainting, video inpainting imposes new challenges of generating temporally consistent results. Chang *et al*. [2] extend gated convolutions [41] to 3D for free-form video inpainting and propose a temporal PatchGAN loss to enhance temporal consistency. Wang *et al*. [33] adopt an approach where a temporal network operates on low-resolution input to ensure temporal consistency and a spatial network recovers details using 2D convolution at a higher resolution. Onion-Peel networks [25] design an asymmetric attention block that computes similarities between the hole boundary pixels in the target and the non-hole pixels in the references in a non-local manner [34].

Optical flow provides dense correspondence between frames. It has been used to extrapolate unknown pixels in video inpainting [14]. Xu *et al*. [39] propose to first inpaint optical flow, which is arguably an easier task than inpainting the original video. Occluded missing pixels are then filled by an image inpainting method [40]. Unlike Xu *et al*. [39], Zhang *et al*. [42] simultaneously inpaint both RGB and optical flow with an internal learning approach that is inspired by the 'Deep Image Prior' [32]. VINet [17] also estimates both RGB and optical flow for the missing pixels. Temporal consistency is encouraged via a recurrent feedback and a ConvLSTM. Instead of computing optical flow explicitly, Copy-and-Paste networks [20] estimate affine transformations to align pixels across frames, and use a context matching module to fuse pixels from multiple reference frames to a target frame.

Our approach differs from the aforementioned ones in that we introduce the concept of 'inpainting proposals.' Bounding box proposals or anchor boxes have been widely used in object detection [12, 28]. However, to the best of our knowledge, proposals have not been considered for tasks like image or video

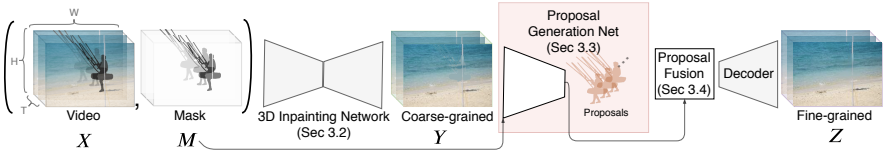


Fig. 2. Illustration of the proposed method. Our method takes as input the video and the mask and employs the developed 3D inpainting network to obtain a coarse-grained inpainting. We subsequently match pixels within the coarse-grained inpainting to observed pixels, creating a pool of proposals that can be used to inpaint the unobserved parts of the video. Finally we fuse the proposals via an attention mechanism and upsample via a decoder to obtain the final fine-grained inpainting.

inpainting. In this paper, we demonstrate that proposals enable pooling non-local patches with similar content very effectively and result in more spatially and temporally consistent inpainting results.

Proposals. Proposals have taken a primary role in object detection: a region-based convolutional neural net (R-CNN) approach [7] evaluates a conv-net for a computationally manageable number of candidate regions of interest (RoI). Each RoI is assessed independently regarding a variety of metrics like 'object-ness,' 'class,' *etc*. Extensions like RoI pooling [7, 13], region proposal networks [28] and RoI alignment [12] have further improved the efficacy of proposals.

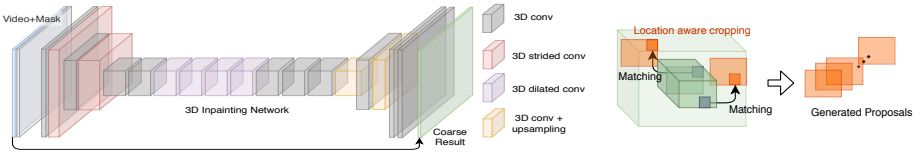
Inspired by the success of proposals in object detection, we introduce the concept of proposals to video inpainting. As mentioned before, we think proposals are ideal to quickly pool spatially and temporally non-local information in the form of similar patches. We provide details of the developed approach subsequently.

3 Proposal-based Video Completion

Given a video $X = (x_1, \ldots, x_T)$ composed of T frames with a resolution of $W \times H$, and corresponding masks $M = (m_1, \ldots, m_T) \in \{0, 1\}^{T \times W \times H}$ specifying the missing region for each frame, we want to recover the RGB values for pixels that are missing. Let x_t^i denote the RGB value of frame x_t at location i, where $i = (u, v) \in \mathbb{R}^2$, $u \in \{1, \ldots, W\}, v \in \{1, \ldots, H\}$ and let the mask $m_t^i = 1$ indicate pixels that are missing and need to be recovered. Unlike prior works, we propose to first create a set of 'inpainting proposals,' which are most similar to the missing regions. We then design a novel attention mechanism to fuse the proposals from different spatial-temporal locations to fill the missing regions. In this section, we first provide an overview of our proposal-based approach, then detail each component of our approach respectively.

3.1 Overview

The developed approach is outlined in Fig. 2. Again, we are given a video X and the corresponding mask M which indicates the location of the missing regions.



(a) 3D inpainting network (b) Proposal generation

Fig. 3. Illustration of (a) the 3D inpainting network and (b) the proposal generation module. Note that every tensor in (a) is in fact 4D ($W \times H \times T \times C$), and C is the number of channels. To avoid generating overly blurry results, we do not apply temporal striding in the 3D inpainting network. To simplify, we ignore the temporal dimension when visualizing the 4D tensors. For proposal generation (b), we match every unobserved pixel (dark blue square) with every observed pixel to find the top-k candidates (dark orange square). The surrounding area following the shape of the missing region is subsequently extracted and added to the pool of proposals.

We first apply our 3D inpainting network to recover a coarse-grained result Y. We then generate inpainting proposals based on the coarse-grained result and finally fuse the proposals via a classifier to generate the inpainting result Z.

Our approach is based on a 3D inpainting network, a proposal generation mechanism and a classifier to fuse the extracted set of proposals. All three components are jointly trained. We provide an overview of each component next before discussing their details in subsequent sections.

3D inpainting network: We first inpaint the input video X with missing regions specified in M via a 3D inpainting network which has an encoder-decoder structure, shown in Fig. 3 (a). To cope with 3D video data, the 3D inpainting network utilizes 3D gated convolutions [41, 2] to better integrate the information from the binary mask. We also apply dilation to 3D gated convolutions instead of using larger kernel sizes to reduce the computational cost. To reduce blurryness, our 3D network only downsamples spatially by a factor of four, and keeps the temporal length T unchanged through the whole network.

Despite our dedicated design choices, the coarse inpainting results from the 3D inpainting network still tend to be blurry. To improve this initial estimate we develop a novel proposal generation network, which refines the inpainting result and yields the final result Z after upsampling via a decoder. We provide more details subsequently.

Proposal generation: Candidates for parts of a missing region of a particular frame may appear anywhere in the video, *i.e.*, good candidates are not necessarily in adjacent frames and are not necessarily spatially close neither. Our key idea is to inpaint the missing region by attending to the observed pixels in the video, looking for candidate patches which fit the coarse inpainting. This permits to effectively combine non-local information represented in the form of candidate patches.

Intuitively, by generating proposals, the temporal and spatial information can be propagated from the observed parts of the video to the missing region. To this end, we develop the 'proposal generation network' to generate a set

of inpainting proposals $\mathcal{P}_t$ by matching features between observed pixels and unknown pixels of the coarse result, as shown in Fig. 3 (b). More specifically, for every unobserved pixel in frame x_t we match features of its coarse-grained estimate to features of any observed pixel in the given video. For every pixel we retain the top-k matching candidates as well as their surrounding pixels as indicated by the mask m_t.

We want to emphasize: a proposal is not locally confined to a single pixel. Very much in contrast, a single proposal can be used to inpaint all missing pixels in a frame.

Proposal fusion: We fuse a pool of proposals $\mathcal{P}_t$ at time t generated by the proposal generation network and compute the final inpainting Z based on the fused result. For this we train a classifier to produce a categorical distribution over all the proposals in order to fuse them. This mechanism establishes dependencies between the missing region and the observed region in a non-local way. The classifier module in our method is a 3-layer CNN to predict a probability distribution over all the proposals. The obtained categorical distribution is used to fuse the proposals.

The fused proposals across all points in time are fed into a decoder to generate the final inpainting Z. Note that the proposal generation network combined with the classifier permits attending to regions that can be spatially and temporally far apart albeit containing similar, useful fine-grained context for inpainting.

In the following, we describe in detail each of the three components of the proposed method.

3.2 3D Inpainting Network

We first inpaint a given video X which contains missing regions specified in M by using a 3D inpainting network. As illustrated in Fig. 3 (a), the architecture of our 3D inpainting network consists of layers of 3D gated convolutions [41, 2] with striding and dilation at different layers. In total, there are 19 layers in the 3D inpainting network to keep the computational cost low. We only do spatial striding twice to reduce the resolution from $W \times H$ to $W/4 \times H/4$ and keep the temporal length T unchanged in all the layers. Upsampling in the decoder is done with bilinear interpolation instead of deconvolution.

The coarse result Y is obtained by fusing the input video X with the raw output of the 3D inpainting network $\bar{Y}$ via

$$Y = M \odot \bar{Y} + (\mathbb{1} - M) \odot X, \tag{1}$$

where M is the mask indicating whether or not a pixel is observed, '$\odot$' denotes element-wise multiplication and $\mathbb{1}$ is the all-ones tensor. The coarse inpainting result Y obtained via this 3D inpainting network tends to be blurry. To rectify this we develop the 'proposal generation network' which we discuss next.

3.3 Proposal Generation

We describe how to generate the 'inpainting proposals' in the following. To generate proposals, we match pixels of the inpainted region in the coarse-grained

result, *i.e.*, pixels which were initially unobserved, to observed regions at any spatial and temporal locations in the input video. Consequently, candidates at any spatial and temporal distance are treated equally. Hence, we consider a much more global context which differs from prior approaches for video inpainting [39] and image inpainting [40].

Our approach hence performs spatially and temporally non-local matching. There are three components in the proposed 'proposal generation network': feature extraction, matching and generating of proposals as discussed next.

Feature extraction: We first extract features from the coarse-grained inpainting Y which are then used for matching. To be more specific, we compute features $F = g(Y)$ from the coarse-grained result Y via a deep net g, in our case an 8-layer CNN. Note that the spatial downsampling factor of g is 8. Therefore the feature map F is of dimension $\mathbb{R}^{T\times w\times h\times c}$, where w is $W/8$ and h is $H/8$. Again, no temporal down-sampling is employed. Let f_t be the feature map at time t. Matching is performed at the resolution of $h \times w$ instead of $H \times W$ such that matching can be performed more efficiently. We downsample the mask m_t as well with a factor of 8 to get $\bar{m}_t$ which indicates the missing pixels at the $h \times w$ resolution. We subsequently use the computed features for matching.

Top-k matching: After feature extraction, we find matches between pixels for which the original video did not provide an RGB value, and pixels for which the RGB value was observed. For this we use the obtained features. To be more specific, let

$$\mathcal{U}_t = \{(i,t)|\bar{m}_t^i = 1, \forall i\}$$

denote the unobserved pixels at time t which are required to be inpainted. Further, let

$$\mathcal{O} = \Big\{(j,t')|\bar{m}_{t'}^j = 0, \forall i, \forall t' \in \{1,\dots,T\}\Big\}$$

refer to all the observed pixels across all times.

To find the top-k matches, we first compute the similarity map $S_t \in \mathbb{R}^{|\mathcal{U}_t|\times|\mathcal{O}|}$ via

$$S_t(a,b) = d(f_a, f_b)$$

for all unobserved pixels $a = (i,t) \in \mathcal{U}_t$ at time t, and for all observed pixels $b = (j,t') \in \mathcal{O}$ irrespective of time and location. Note, given $a = (i,t)$, f_a refers to the feature at location i in f_t. So does f_b for $b = (j,t')$. We use $d(f_a, f_b)$ to denote the similarity between two features. We use the classical cosine distance as the distance function d in our implementation and leave exploration of more complex distance functions to future work.

Based on the similarity map S_t, we select the top-k matches $\mathcal{K}_a$ as follows:

$$\mathcal{K}_a = \{b|b \in \mathrm{Top}_k(S_t(a,b))\}.$$

Hence, for a given unobserved pixel $a \in \mathcal{U}_t$, $\mathcal{K}_a$ consists of only the top-k matches across all spatial locations and across all frames. The set $\mathcal{K}_a$ is then used to generate the set of proposals $\mathcal{P}_t$.

Generating proposals: We illustrate our method to generate proposals in Fig. 3 (b). After finding the top-k matches for every $a \in \mathcal{U}_t$, for each match, we

crop its surrounding region and generate a proposal p_a which refers to a set of feature vectors. We emphasize that each proposal p_a can be used to inpaint all missing pixels in frame x_t. Finally, we use these proposals to construct the set of proposals $\mathcal{P}_t$. Formally, we obtain

$$\mathcal{P}_t = \{p | p = \text{crop}(f_{t'}, j, i), \forall a = (i,t) \in \mathcal{U}_t, \forall b = (j,t') \in \mathcal{K}_a\}\,.$$

Note that the crop operation is location-aware. Concretely, our method does not crop the rectangle with its center location at j but instead crops a rectangle with the size of the missing region, while keeping the relative location of j inside the cropped rectangle identical to the relative location of i inside the missing region.

Hence, note that we don't only use a top-k match locally for the corresponding pixel. Very much in contrast, we use the locally computed top-k match to extract a proposal which provides information for all the missing pixels at time t. This is crucial as it permits our method to create many viable candidates, each of which can be used to inpaint all missing pixels at once.

Subsequently we detail how we propose to compute the final inpainting Z.

3.4 Proposals fusion

After generating the set of proposals $\mathcal{P}_t$ for the frame at time t, we fuse them via a classifier. Let p_n be one of the proposals in $\mathcal{P}_t$, *i.e.*, $p_n \in \mathcal{P}_t$ and let $p_{\{1,\dots,|\mathcal{P}_t|\}}$ be the concatenation of all $|\mathcal{P}_t|$ proposals. Recall that f_t is the feature map at time t obtained from the coarse-grained inpainting. For each unobserved pixel $(i,t) \in \mathcal{U}_t$, we compute the categorical distribution $A_i \in [0,1]^{|\mathcal{P}_t|}$ over all the proposals, via a classifier C with soft-max for normalization, *i.e.*,

$$A_{i,n} = \frac{\exp\{C(p_{\{1,\dots,|\mathcal{P}_t|\}}, f_t)_{i,n}\}}{\sum_{n'} \exp\{C(p_{\{1,\dots,|\mathcal{P}_t|\}}, f_t)_{i,n'}\}}.$$

The classifier C operates on the concatenation of all the proposals as well as the feature map f_t. We then fuse the proposals using distribution A_i to obtain the attended feature p_t^i for pixel i at time t via

$$p_t^i = \sum_{n=1}^{|\mathcal{P}_t|} A_{i,n} \cdot p_n^i.$$

Here, p_n^i refers to the feature at location i in proposal p_n. The fused feature map $P = \{(p_t^i)\}_{\forall i,t}$ is padded such that it has the same size as F. We subsequently concatenate the feature map P with the extracted coarse-grained features F and employ a decoder to compute the inpainting result $\bar{Z}$. To obtain the final result Z, we merge $\bar{Z}$ with the input X following Eq. (1).

3.5 Training

The described approach is trained end-to-end. For training of the proposal generation mechanism we construct a dataset which contains the ground-truth completion Z^*, the coarse-grained result Y and the final result Z. We jointly learn

end-to-end the parameters of the 3D inpainting network (Sec. 3.2), the parameters of the classifier (Sec. 3.4), and the decoder, by optimizing the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{L1}} + \lambda_1 \mathcal{L}_{\text{G}} + \lambda_2 \mathcal{L}_{\text{CE}}. \tag{2}$$

The objective consists of a pixel-wise L1 error $\mathcal{L}_{\text{L1}}$, an adversarial loss $\mathcal{L}_{\text{G}}$ and a cross-entropy loss $\mathcal{L}_{\text{CE}}$ for fusing the proposals. Hyper-parameters λ_1 and λ_2 are used to adjust the impact of the different loss components. We describe details for each of the individual loss terms in the following.

Reconstruction loss $\mathcal{L}_{\text{L1}}$: The L1 error is used for penalizing if the inpainting result deviates from the ground-truth. We penalize both the coarse-grained result as well as the refined result using the L1 loss, *i.e.*, we use

$$\begin{aligned}\mathcal{L}_{\text{L1}} = \|M \odot (Z^* - \bar{Y})\|_1 + \gamma \|(\mathbb{1} - M) \odot (Z^* - \bar{Y})\|_1 \\ + \|M \odot (Z^* - \bar{Z})\|_1 + \gamma \|(\mathbb{1} - M) \odot (Z^* - \bar{Z})\|_1.\end{aligned} \tag{3}$$

The hyper-parameter γ controls the loss occurring due to reconstruction of observed parts of the image.

Adversarial loss $\mathcal{L}_{\text{G}}$: An adversarial loss is commonly used for inpainting [40, 41, 2]. Let D be the discriminator and G be the inpainting network, *i.e.*, $Z = G(X, M)$ where X and M are input video and masks. As suggested in [41], we use a fully convolutional network as the discriminator. Note that the output of the discriminator is a tensor rather than a scalar. We compute the adversarial loss on each of the elements in the output tensor using a discriminator and accumulate by averaging.

The inpainting network minimizes the below objective to fool the discriminator, *i.e.*,

$$\mathcal{L}_{\text{G}} = -\mathbb{E}_{X,M}\big[D(G(X, M))\big].$$

The discriminator is trained to differentiate the inpainting result from the real video. We optimize the discriminator using the following objective containing a hinge-loss activation function:

$$\mathcal{L}_{\text{D}} = \mathbb{E}_{X,M}\big[\max\big(0, \mathbb{1} - D(G(X, M))\big)\big] + \mathbb{E}_{Z^*}\big[\max\big(0, \mathbb{1} + D(Z^*)\big)\big],$$

where Z^* is the ground truth video.

For stable training, we apply spectral normalization [22] to the discriminator. The discriminator is a fully convolutional net (FCN) with 6 layers of 3D convolutions. Because of our use of FCNs, the size of the input video isn't fixed during training.

Cross entropy loss $\mathcal{L}_{\text{CE}}$: An important component of the proposed method is the classifier C to compute the categorical distribution over all proposals as described in Sec. 3.4. To train the classifier, we obtain the labels by first extracting features using the feature extractor g described in Sec. 3.3 to get the feature map z_t of the ground-truth completion, *i.e.*, $z_t = g(Z^*)$. From it we obtain the ground truth distribution A_i^* via

$$A^*_{i,n} = \begin{cases} 1 & \text{if } n = \arg\max_{n'} \|z_t^i - p_n^i\|, \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, we compare the feature of the ground-truth completion to the proposals, find the one that best matches with the ground truth, and use this as the training label. After obtaining the training label, we minimize the cross-entropy loss between the predicted distribution A and the label A^*, *i.e.*, we use

$$\mathcal{L}_{\text{CE}} = -\sum_{i \in \mathcal{U}_t} \sum_{n=1}^{|\mathcal{P}_t|} A^*_{i,n} \log A_{i,n}.$$

3.6 Implementation Details

To optimize the objective given in Eq. (2) w.r.t. the network parameters, we use the Adam optimizer [19] with a learning rate of 1e-4. We first train the 3D inpainting network with only the L1 loss objective for 6,000 iterations with a batch size of 64. Subsequently we train the entire framework using all objectives for another 6,000 iterations. We use $\lambda_1 = 1$, $\lambda_2 = 0.05$, $k = 1$ and, $T = 8$. The inference time of our method on the DAVIS dataset (resolution 854×480) is around 0.69 seconds per frame using one NVIDIA V100 GPU.

4 Experimental Results

We evaluate the proposed approach on two datasets, following the experimental setup of prior work [39]. For completeness we first provide experimental settings before discussing our results.

4.1 Experimental setting

We first describe the datasets which we use for experiments before discussing metrics for comparison with baselines and mask generation.

Datasets: We use the DAVIS [26] and YouTube VOS [38] datasets which are both widely used in video inpainting [39, 36, 17]. DAVIS [26, 27] consists of 150 videos in total, providing high-quality pixel-level annotations for foreground objects. We follow the evaluation protocol in [39], and use 60 videos from the dev split for training and the remaining 90 videos where we have object masks for testing. YouTube VOS [38] is a much larger dataset with more than 4000 videos in total. It is a more challenging dataset as its videos are much longer than DAVIS (140 *vs*. 68.9 frames per video on average), have higher resolution, and cover a large variety of different scenarios. We use the training set which contains 3,471 videos to learn the parameters of our proposed proposal-based video-inpainting method, and evaluate on the test set which contains 541 videos. We use the 507 videos in the validation set for choosing hyperparameters.

Evaluation metrics: To measure the similarity between the inpainted videos and the ground truth, we use three metrics: structural similarity (SSIM) [35], peak signal-to-noise ratio (PSNR) and learned perceptual image patch similarity (LPIPS) [43]. We compute the three metrics on the entire video and also on the

Table 1. Results of our method compared to baselines on the YouTube VOS test set [38].

	Runtime	Inpainted region only			Entire frame		
	(per frame)	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓
Deepfill2 (ICCV'19) [41]	0.32 s	0.336	9.228	0.448	0.958	21.271	0.034
FGI (CVPR'19) [39]	112.32 s	0.355	10.890	0.409	0.959	22.934	0.032
OPN (ICCV'19) [25]	9.05 s	0.437	12.242	0.394	0.964	24.286	0.029
CPN (ICCV'19) [20]	1.40 s	0.412	11.795	0.478	0.962	23.845	0.036
VINet (CVPR'19) [18]	**0.18 s**	0.348	10.338	0.549	0.958	22.381	0.043
Ours	0.87 s	**0.445**	**13.292**	**0.388**	**0.969**	**25.821**	**0.030**

Table 2. Results of our method compared to baselines on the DAVIS dataset [26].

	Runtime	Inpainted region only			Entire frame		
	(per frame)	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓
Deepfill2 (ICCV'19) [41]	**0.09 s**	0.237	8.899	0.435	0.951	20.970	0.035
FGI (CVPR'19) [39]	37.63 s	0.341	10.974	**0.353**	0.958	23.045	**0.028**
OPN (ICCV'19) [25]	4.17 s	0.344	11.930	0.379	0.958	24.002	0.029
CPN (ICCV'19) [20]	0.53 s	0.316	11.338	0.507	0.956	23.409	0.040
VINet (CVPR'19) [18]	0.18 s	0.254	9.388	0.570	0.951	21.459	0.045
Ours	0.69 s	**0.348**	**12.453**	0.381	**0.959**	**24.511**	0.031

inpainted region only. This permits to assess the quality of both the entire frame and the inpainted region.

We evaluate the method on two inpainting scenarios, *i.e.*, fixed region inpainting and object removal.

4.2 Fixed Region Inpainting

Fixed region inpainting is a common task [39] in video inpainting to study the ability of completing a fixed missing region. Though the mask is fixed, this task is very challenging as the fixed regions can cover a large portion of the video and often break the irregular object boundaries. Following the setup in [39], given an input video of resolution $W \times H$, we mask out a rectangular region of size $W/4 \times H/4$ at the center of the frame. In the following we first compare quantitatively to the state-of-the-art (SOTA) using the aforementioned metrics before providing and discussing qualitative results on both YouTube VOS and DAVIS.

Comparison to SOTA: We provide a comparison to state-of-the-art video inpainting methods on YouTube VOS in Tab. 1. We evaluate SSIM, PSNR (higher is better) and LPIPS (lower is better) on both the inpainted region only and the entire frames (*i.e.*, the entire inpainted result Z).

From the results reported in Tab. 1 we observe that the proposed approach significantly improves all metrics. The improvements are slightly more pronounced when looking at the inpainted region only.

We conduct a similar evaluation on the DAVIS dataset and provide results in Tab. 2. We observe the recently proposed flow guided inpainting (FGI) [39]

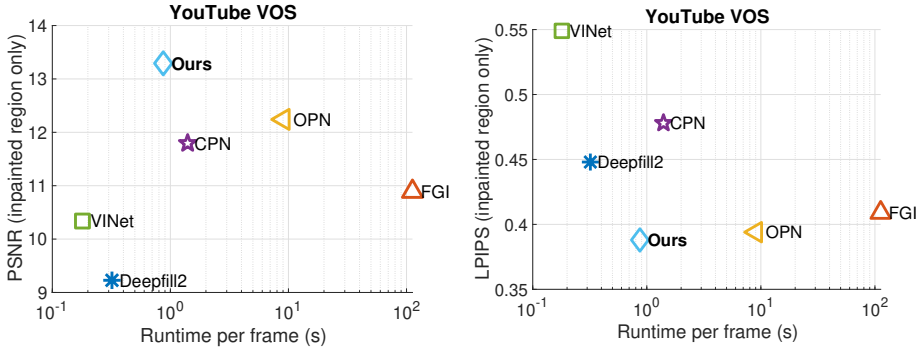


Fig. 4. Comparison of runtime *vs*. performance in PSNR (left) and LPIPS (right) with all methods on the YouTube VOS dataset. Note that for PSNR higher is better, and for LPIPS lower is better. Our method achieves better performance while running efficiently compared to baselines.

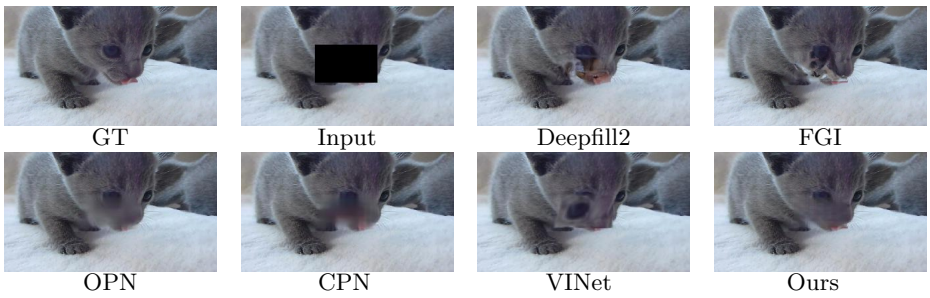


Fig. 5. Video inpainting results of our method compared to the baselines.

to be a competitive baseline. While our proposal based video completion falls short on the LPIPS metric, we observe improvements on PSNR and SSIM. Note that the runtime of FGI is much higher than ours as it's an iterative method and doesn't scale well with the length of the video.

Runtime: We report the average runtime of each method on the YouTube VOS and DAVIS datasets in Tab. 1 and Tab. 2. Our method is the second fastest one among the state-of-the-art video inpainting approaches on YouTube VOS and the third fastest on DAVIS. Note that the flow based approach FGI usually requires tens of iterations of propagation and is therefore time consuming. We plot the runtime *vs*. PSNR and LPIPS in Fig. 4. Our method achieves better performance while running efficiently compared to baselines.

Qualitative results: We provide qualitative results for video inpainting of our method and existing baselines in Fig. 4.2. For challenging cases which exhibit significant appearance changes we observe accurate video completion results. Deepfill2 [41] is an image-based baseline. The completion is less smooth since no temporal information is taken into account. FGI [39] largely relies on flow

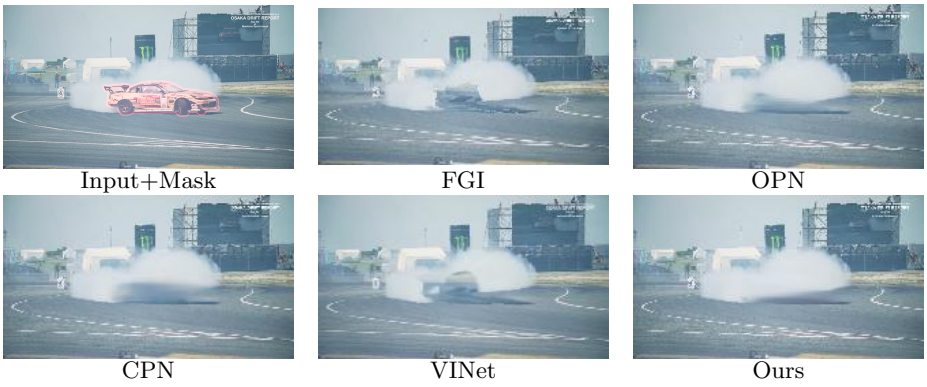


Fig. 6. Object removal results of our method compared to video inpainting baselines on the sequence `drift-chicane` of the DAVIS dataset.

Table 3. Ablation study of our method on YouTube VOS validation and test sets [38].

3D Inpainting Net	Proposal	Classifier	Performance on YouTube VOS Val			Performance on YouTube VOS Test		
			SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓
✓			0.421	14.33	0.415	0.418	11.80	0.458
✓	✓		0.432	14.41	0.377	0.422	11.84	0.433
✓	✓	✓	**0.459**	**16.22**	**0.301**	**0.445**	**13.29**	**0.388**

which is tricky especially when motion is complex. We observe other baselines to produce overly smooth results (CPN [20]) and unrealistic completion (OPN [25] and VINet [18]). We observe encouraging completions despite the fact that the proposed approach can be extended in many different directions.

4.3 Video Object Removal

We study applicability of the proposed method to object removal in videos. We use the DAVIS dataset [26] for this study as the dataset provides accurate object segmentations which specify the region to inpaint. In Fig. 6, we show the object removal results of our approach compared to the state-of-the-art video inpainting baselines, FGI [39], OPN [25], CPN [20] and VINet [18] on the DAVIS dataset. Compared to the baselines we observe our method to work well on object removal in videos, producing realistic results. Our method can inpaint arbitrary-shaped masks as shown in the first row of Fig. 1. More results can be found in the supplementary.

4.4 Ablation Study

To better understand the impact of individual components of the proposed approach we report results of an ablation study in Tab. 3. We use the YouTube VOS validation and test sets and the fixed region inpainting setup in this experiment.

We report the metrics computed on the inpainted region only. Specifically, we analyze the accuracy of our 3D inpainting net (discussed in Sec. 3.2). Using the proposal generation mechanism (discussed in Sec. 3.3) and fusing the results via a single convolution with learnable parameters reduces SSIM and PSNR metrics while it improves LPIPS. Finally, by combining the three developed parts, *i.e.*, 3D inpainting, proposal generation and the classifier (discussed in Sec. 3.4) we achieve the most accurate results. We observe the performance improvements to generalize to the test set.

4.5 Failure Cases

The proposed approach is challenged by thin structures and small objects with a large missing region ratio. This can be observed in Fig. 7.

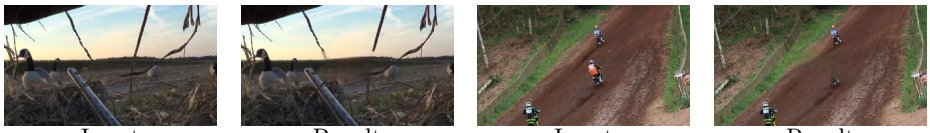


Fig. 7. Two failure cases. Input frame on the left, inpainting on the right.

5 Conclusion

We develop a proposal-based 3D video completion method. Different from prior work, we argue that proposals accurately summarize spatially and temporally non-local candidates that could be used for inpainting. To compute those proposals we first employ a developed 3D inpainting network which yields an initial coarse-grained estimate. To fuse the proposals we develop a classifier-based prediction mechanism. Despite the simplicity of the proposed method, we show on challenging datasets that the use of proposals indeed leads to accurate results. Going forward, we think better strategies to fuse the proposals and more intricate ways to match candidates can lead to even bigger improvements.

Acknowledgements: This work is supported in part by NSF under Grant No. 1718221 and MRI #1725729, UIUC, Samsung, 3M, and Cisco Systems Inc. (Gift Award CG 1377144). We thank Cisco for access to the Arcetri cluster.

References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. In: ACM TOG (2009)
2. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: Proc. ICCV (2019)
3. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Learnable gated temporal shift module for deep video inpainting". In: Proc. BMVC (2019)
4. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE TIP (2004)
5. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proc. Computer graphics and interactive techniques (2001)
6. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proc. ICCV (1999)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. CVPR (2014)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. NeurIPS (2014)
9. Granados, M., Kim, K.I., Tompkin, J., Kautz, J., Theobalt, C.: Background inpainting for videos with dynamic objects and a free-moving camera. In: Proc. ECCV (2012)
10. Granados, M., Tompkin, J., Kim, K., Grau, O., Kautz, J., Theobalt, C.: How not to be seenobject removal from videos of crowded scenes. In: Computer Graphics Forum (2012)
11. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM TOG (2007)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proc. ICCV (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE TPAMI (2015)
14. Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Temporally coherent completion of dynamic video. ACM TOG (2016)
15. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM TOG (2017)
16. Ilan, S., Shamir, A.: A survey on data-driven video completion. In: Computer Graphics Forum (2015)
17. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep blind video decaptioning by temporal aggregation and recurrence. In: Proc. CVPR (2019)
18. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep video inpainting. In: Proc. CVPR (2019)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Lee, S., Oh, S.W., Won, D., Kim, S.J.: Copy-and-paste networks for deep video inpainting. In: Proc. ICCV (2019)
21. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proc. ECCV (2018)
22. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
23. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)

24. Newson, A., Almansa, A., Fradet, M., Gousseau, Y., Pérez, P.: Video inpainting of complex scenes. SIAM Journal on Imaging Sciences (2014)
25. Oh, S.W., Lee, S., Lee, J.Y., Kim, S.J.: Onion-peel networks for deep video completion. In: Proc. ICCV (2019)
26. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proc. CVPR. pp. 724–732 (2016)
27. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv:1704.00675 (2017)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proc. NeurIPS (2015)
29. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: Structureflow: Image inpainting via structure-aware appearance flow. In: Proc. ICCV (2019)
30. Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Jay Kuo, C.C.: Contextual-based image inpainting: Infer, match, and translate. In: Proc. ECCV (2018)
31. Strobel, M., Diebold, J., Cremers, D.: Flow and color inpainting for video completion. In: German Conference on Pattern Recognition (2014)
32. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: Proc. CVPR (2018)
33. Wang, C., Huang, H., Han, X., Wang, J.: Video inpainting by jointly learning temporal structure and spatial details. arXiv preprint arXiv:1806.08482 (2018)
34. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proc. CVPR (2018)
35. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE TIP (2004)
36. Woo, S., Kim, D., Park, K., Lee, J.Y., Kweon, I.S.: Align-and-attend network for globally and locally coherent video inpainting. arXiv preprint arXiv:1905.13066 (2019)
37. Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: Proc. CVPR (2019)
38. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018)
39. Xu, R., Li, X., Zhou, B., Loy, C.C.: Deep flow-guided video inpainting. arXiv preprint arXiv:1905.02884 (2019)
40. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proc. CVPR (2018)
41. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proc. ICCV (2019)
42. Zhang, H., Mai, L., Xu, N., Wang, Z., Collomosse, J., Jin, H.: An internal learning approach to video inpainting. In: Proc. ICCV (2019)
43. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. CVPR (2018)