Flow-Guided Video Inpainting with Scene Templates

Dong Lao   Peihao Zhu   Peter Wonka   Ganesh Sundaramoorthi

KAUST, Saudi Arabia

{dong.lao, peihao.zhu, peter.wonka, ganesh.sundaramoorthi}@kaust.edu.sa

Abstract

We consider the problem of filling in missing spatio-temporal regions of a video. We provide a novel flow-based solution by introducing a generative model of images in relation to the scene *(without missing regions) and mappings from the scene to images. We use the model to jointly infer the* scene template*, a 2D representation of the scene, and the mappings. This ensures consistency of the frame-to-frame flows generated to the underlying scene, reducing geometric distortions in flow based inpainting. The template is mapped to the missing regions in the video by a new (L^2-L^1) interpolation scheme, creating crisp inpaintings and reducing common blur and distortion artifacts. We show on two benchmark datasets that our approach out-performs state-of-the-art quantitatively and in user studies.*[1]

1. Introduction

Video inpainting is the problem of filling spatial-temporal regions, i.e., masked regions, with content that naturally blends with the remaining parts of the video. This is useful in video editing tasks, including removing watermarks or unwanted objects and video restoration. As videos exhibit temporal regularity, to inpaint a given frame, it is natural to use data from other frames, as the data in other frames may correspond to parts of the scene behind the masked region. Many state-of-the-art methods for video inpainting are flow-guided [7, 29, 19, 5], which take the approach of copying unmasked data from other frames into the masked region of a given frame by using optical flow.

While these approaches inpaint with plausible data from the scene through other frames, unlike single image inpainting methods (e.g. [3, 35]) that attempt to halluncinate image values in the masked region from other regions in the image or learned through datasets, they are highly dependent on the quality of the optical flow. Even though optical flow has advanced significantly with the progress of deep learning to the point that recent methods on benchmark optical flow
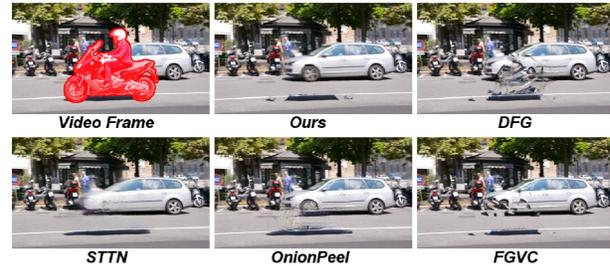
Figure 1: **Comparison with state-of-the-art.** Our approach uses a generative model of formation of images from the scene to infer flows that are consistent with the scene. This reduces visual distortions in inpainting compared with state-of-the-art: DFG [29], STTN [37], OnionPeel [19], FGVC [5]. *Animation in the supplementary materials.*

datasets produce only small errors, there are two complications in applying optical flow to inpainting. First, the flow that is needed in the masked region for inpainting is the flow had the contents within the masked region been removed to reveal the part of the scene that the mask occludes. As this is not possible to determine directly, it is hallucinated, typically learned through data [29, 5]. However, there is no guarantee that this is consistent with the scene or hallucinations from other frames, producing visual distortions in inpainting. Secondly, even small errors in optical flow can produce noticeable visual distortions in the inpainted video, which is further amplified as flow is aggregated over multiple frames as data from far away frames may be needed. Attempts have been made to reduce these errors by applying temporal regularity to the flow [29, 38, 37, 5], but these naive regularizers (flow between frames is close) may not be consistent with the scene geometry and can still produce visual distortions (Figure 1).

In this paper, we aim to reduce visual distortions in video inpainting due to physically implausible and temporally inconsistent flows by deriving a generative model that closely models the physical image formation process in generating images from the scene, and using it to infer flow. This model represents the 3D scene (all of the scene correspond-

ing to the video outside the masked regions) as a 2D *scene template*. The inpainted image at frame t is a mapping (general piecewise smooth warping) of the part of the scene template in view in frame t to the image domain. The model constrains the warps to be consistent with the *scene* and the images. This induces temporal consistency by naturally and efficiently enforcing that all pairwise mappings between images generated from the model must correctly match images. This reduces implausible flow hallucinations common in current approaches. Our inference procedure computes the warps and the scene template jointly from this model, which reduces distortions in inpainting. Our contributions are specifically:

1. We solve flow-guided inpainting by applying a generative model of the scene to images, and using it to infer flow and a model of the scene (the *scene template*). This gives more temporally consistent and plausible pair-wise flows compared with existing methods, which results in inpainting results that are more temporally consistent and have less geometric distortions. **2.** We propose a novel L^2-L^1 combined optimization procedure that generates the inpainting from the scene template together with a interpolation strategy that significantly improves inpainting quality and further reduces geometric distortions and blurring artifacts. **3.** We introduce the first benchmark dataset (*Foreground Removal*) on removing occluding objects from video. We introduce a quantitative protocol while previous art relies on visual comparisons. **4.** We demonstrate the advantage of our algorithm on the DAVIS [21] and Foreground Removal datasets and demonstrate superior results (both through user studies and quantitatively) compared to state-of-the-art.

2. Related Work

Video Inpainting: Single image inpainting methods [4, 3, 36, 35, 8, 17] have had success in the past decades. However, when applied to video data, they generally produce artifacts due to a lack of temporal consistency. Early video inpainting methods [28, 20, 18] extend patch-based single image techniques to video data. More recent works [7, 22, 29, 14, 38, 5] use optical flow or variants to model spatio-temporal correspondence across frames. To hallucinate flow inside the masked region, non-learning approaches [22, 7] rely on energy minimization assuming smoothness of the flow; [29, 5] is a deep learning solution that first computes flow between image pairs, then uses a neural network to hallucinate flow inside the masked region. End-to-end learning methods [11, 26, 19, 38, 2, 37, 15] model cross-frame correspondence in their loss functions. For example, [38] jointly infers appearance and flow while penalizing temporal inconsistency. These methods only process a small fixed number of frames or run at low resolution as they are limited by hardware constraints. As our method does not have such a limitation and produces con-

sistency with the scene, we out-perform these methods.

Layered Approaches: Our method relates to layered approaches [23, 27, 13, 10], which represent a scene as moving 2D layers that can occlude each other. Layered approaches are powerful tools that can be applied to motion segmentation [1, 32, 24, 34, 12, 30, 31] as they provide a principled way of occlusion reasoning in videos. We adopt a layered formulation to create our scene template using modern advances in optical flow and deep learning, which we then use for inpainting.

3. Computing the Scene Template

We formulate a joint inference problem for the scene template and a set of transformations (warps) of the template to each of the images. The scene template, i.e., the *background*, is a 2D representation of radiance of the scene without the *foreground*, i.e., part of the scene corresponding to masked regions in images to be inpainted. The inference problem arises from a generative model that explains how images are formed through geometric transformations of the scene template; these transformations model transformations arising from both camera viewpoint change and dynamic scenes. This inference constrains motion behind masks to be plausible by being consistent with the generative model and scene template, and hence across video frames. This alleviates problems with frame-to-frame flow propagation approaches, which aim to hallucinate the motion in an image behind the masks without scene consistency. Such a hallucination can lead to errors, which is further amplified over multiple frames through temporal propagation. Given the scene template and transformations, the inpainted result is the mapping of the template to the mask of the image to be inpainted (see Figure 2).

3.1. Notation and Assumptions

The video, a set of frames, is denoted $\{I_i\}_{i=1}^T$ where $I_i : D \subset \mathbb{R}^2 \to \mathbb{R}^k$ ($k = 3$ for RGB values) is an image, D is the image domain, and T is the number of frames. We denote the radiance function of the background as $f : \Omega \to \mathbb{R}^k$, and $\Omega \subset \mathbb{R}^2$ is the domain of the scene template (often larger than D to accommodate data from all images). We denote the mappings (warps) from the domain of the scene template to each image domain as $\{w_i : \Omega \to D\}_{i=1}^T$. Note that w_i actually only maps the visible portion of the scene Ω at frame i to D, which is important to deal with moving cameras (details in next section). w_i's are non-rigid, so can handle dynamic scenes/backgrounds. Our model assumes images (outside the mask) are obtained from the scene as $I_i(x) = f(w_i^{-1}(x)) + \eta_i(x)$, where $\eta_i(x)$ is a noise process (to model un-modeled nuisances, e.g., small illumination change, shadows, etc) and w_i^{-1} is the inverse mapping from image i to the template. For the purpose of computation, η
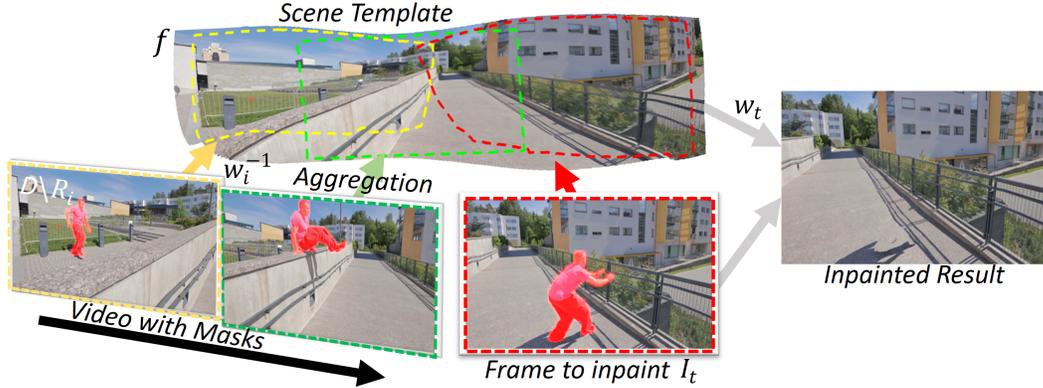
Figure 2: **Schematic of our approach.** Unmasked regions in images are mapped via piece-wise smooth warps w_i^{-1} to create the scene template. Note portions of the scene template that correspond to masked regions in images are naturally filled from other frames. The scene template and the warps are jointly inferred. This ensures warps are consistent with the scene and each other as the formulation implicitly imposes that pairwise mappings $w_j \circ w_i^{-1}$ through the scene template must correctly match unmasked portions of I_i and I_j. The part of the scene template in view at time t is mapped to a given video frame via the inferred warp to obtain the inpainted result in frame t.

will be assumed to be a zero-mean Gaussian noise process independent of x and i following Lambertian assumptions.

In video inpainting, masks M_i's are given for each frame. These can either be provided by user annotation or one can leverage object segmentation algorithms. M_i can contain multiple objects (that may move in different ways) of arbitrary size and shape. Inpainting is to retrieve the radiance in the scene behind M_i's.

3.2. Energy Minimization Formulation

We now formulate the inference of the scene template f and the warps w_i's as a joint energy minimization problem. Note that if f is known, w_i's can be determined by an optical flow problem. Vice-versa, if w_i's are known, then template radiance can be determined by back warping the region outside the masks to Ω. As neither of them are known, the problem is setup as a joint energy minimization on warps and the scene template as follows:

$$E_f(f, \{w_i\}_{i=1}^T) =$$
$$\sum_{i=1}^T \int_{D \setminus M_i} |I_i(x) - f(w_i^{-1}(x))|_2^2 \, \mathrm{d}x + \sum_{i=1}^T E_{Reg}(w_i). \tag{1}$$

The first term above favors warps (and templates) such that the mapping of the visible part of the scene radiance into the image domain matches the image intensity for all pixels outside the mask M_i. Each pixel in each image outside M_i maps to the scene template, and so each $x \in D \setminus M_i$ corresponds to some point in Ω, though not each point in Ω will correspond to some point in D. This is desired since the scene encompasses more than just what is visible from a single frame I_i. This is particularly important as we as-

sume that the camera may translate, and so only a portion of the scene template will be visible in frame I_i (see Figure 2), and thus the first term only penalizes the visible portion of the radiance $f(w_i^{-1}(x)), x \in D \setminus M_i$ in deviating from the image intensity. The second term is warp regularity that is required to make the problem well-posed in light of the aperture problem. We will discuss the particular form of regularization in Section 3.3.

One can recognize that this formulation is similar to optical flow, but for some key differences. Rather than mapping between frames, we map between images and the scene template (which is to be determined), providing natural consistency of the mappings with the scene and hence also each other, which is not present in previous inpainting works.

3.3. Optimization

To optimize, we iteratively update the scene template given the current estimate of the warps and vice-versa the warps given the estimate of the template.

Update for the scene template: Given estimates of w_i's, Ω is computed as the union of the back-warpings of each image domain, i.e., $\Omega = \cup_{i=1}^T w_i^{-1}(D)$. Note that Ω can be larger than D. We now update the scene template radiance f given Ω. Since f only appears in the first term of (1), we can ignore the second term to determine f. Performing a change of variables to compute the integrals on Ω, allows the summation to be moved inside the integrand. One can then show that the global optimizer for f is:

$$f^*(p) = \frac{\sum_{i=1}^T I_i(w_i(p)) \mathbb{1}_i(w_i(p)) J_i(p)}{\sum_{i=1}^T \mathbb{1}_i(w_i(p)) J_i(p)}, \quad p \in \Omega \tag{2}$$

where $\mathbb{1}_i(\cdot)$ is the indicator function of $D \setminus M_i$, i.e., 1 in

Algorithm 1 *Optimization for the scene template.*
1: Choose a key frame I_k
2: Initialize for w_i's by $w_i = w_{ki}$
3: **repeat** *// updating warp and template*
4:    $\Omega = \cup_{i=1}^{T} w_i^{-1}(D)$ and compute f by (2)
5:    For all i, update w_i and w_i^{-1} by computing optical flow between f to I_i *// computed in parallel*
6: **until** converges

Algorithm 2 *Scene template / Inpainting (faster).*
1: Initialization: $t = 1$, initialize w_1 by identity map
2: $t \leftarrow t + 1$, a new frame I_t and mask R_t available
3: Compute $w_{t,t-1}$ and $w_{t,t-1}^{-1}$ by optical flow
4: For all i's, update warps by $w_i \leftarrow w_{t(t-1)} \circ w_i$
5: **repeat** *// updating warp and template*
6:    Update template f by (2)
7:    For all i, update w_i's and w_i^{-1}'s by optical flow between I_i and f *// computed in parallel*
8: **until** converges
9: Compute inpainting result by (5)
10: Go to Step 2

$D \backslash M_i$, the background region in I_i, and 0 otherwise, and $J_i(p) = \det \nabla w_i(p)$, which results from the change of variables and measures the area distortion in the warp between Ω and D. To obtain the radiance at p, one computes a weighted average of all image intensity values from pixels over frames i that correspond to p.

Update for the warps: Given an estimate of the template f^*, we minimize (1) with respect to w_i's. This is equivalent to computing

$$w_i^* = \arg\min_{w_i} \int_{D \backslash M_i} |I_i(x) - f^*(w_i^{-1}(x))|_2^2 \, dx + E_{Reg}(w_i),$$
(3)

for each i, which is similar to an optical flow problem, but only the non-mask region ($D \backslash M_i$) is matched, and the shapes of the domains of the I_i (D) and f (Ω) are different. For convenience, w_i is extended to the whole domain Ω by a smooth extension outside the portion of the template visible in frame i through spatial regularity on all of Ω, which is determined by the second term. To naturally handle different shaped domains, we use SobolevFlow [33] to refine the warps initialized by current estimates of w_i's. Our initialization for w_i (described below) uses frame-to-frame flow composition, and the flow update (3) mitigates errors from the frame-to-frame flow and its aggregation, as it induces geometric consistency (the w_i's must be consistent with the scene template, which is a function of w_i's, and hence geometrically consistent with each other).

Initialization: We choose a key frame (e.g., the middle frame) I_k to be the scene template and initialize the warps w_i to be the optical flow between the key frame and frame i, i.e., $w_i = w_{ki}$. Note that k and i may represent distant frames, and thus may involve large displacements, challenging for optical flow methods. Therefore, we first compute warp between adjacent frames, i.e. $w_{i(i+1)}$ and $w_{(i+1)i}$, by ordinary optical flow then w_{ki} can be computed as a recursive composition of appropriate consecutive frame flows.

To compute the frame-to-frame flow, we use SobolevFlow [33], which naturally allows one to exclude the masked region M_i from computation leading to accurate flow. For faster computation, we initialize this with a deep learning based optical flow (FlowNet2 [9] to extract flow in the whole image, and then replace the flow inside the mask by spatial regularity). We compute flow in both forward and backward direction, and let $w_{ij}^{-1} = w_{ji}$ for the initialization, so all warps have an inverse.

Algorithm 1 summarizes the optimization pipeline. Empirically, it takes at most 2 iterations to converge.

3.4. Efficient Updates of the Scene Template

When processing longer videos with large range of background motion, Ω can grow arbitrarily large, reducing computational and memory efficiency. However, one usually does not need a full scene template to inpaint a frame I_i, since $w_i^{-1}(M_i)$ usually only maps to a small portion of the template. Therefore, we further propose an efficient implementation of our approach. To do this, we ensure that the scene template is aligned to the newest frame as follows. Suppose the template aligns with I_t, then given a new frame I_{t+1}, we update the warps to align to I_{t+1} through compositions $w_i \leftarrow w_i \circ w_{(t+1)t}$ and $w_i^{-1} \leftarrow w_{t(t+1)} \circ w_i^{-1}$ for all i's, and f is updated by (2). We can then crop Ω to be D (matching I_{t+1}), and $w_{t+1}^{-1}(M_{t+1})$ is likely to remain in D since w_{t+1}^{-1} is close to the identity map, as the initial template aligns with I_{t+1}. This also makes refinement of w_i's easier to be handled by existing optical flow methods, as the domains of the I_i and f are the same.

Further, w_i's are computed in parallel since the update of each are independent. Algorithm 2 shows this scheme, which is used for the experiments. Since each frame is only inpainted by previous frames, there may be holes remaining in the initial frames. To alleviate this, we do a forward sweep to inpaint every frame followed by a backward sweep, which fills the initial frames. In the experiments, We find using a sliding window of 7 frames to solve the optimization already provides good results. Nevertheless, our method can handle much larger number of frames (e.g. 60 frames in Figure 2) without hitting hardware constraints.

4. Inpainting

Although we can simply map the template f into the masked region M_i to be inpainted via the warp w_i computed in the previous section to produce the inpainting, this
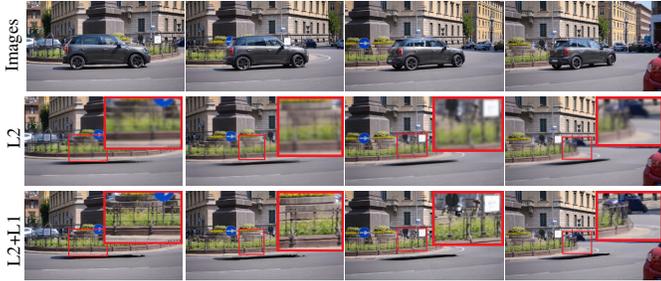
Figure 3: **Example:** L^2 **vs** L^2+L^1. L^2 creates smooth but blurry inpainting results. By L^1 regularization, the inpainting result preserves sharp and rigid appearance.

(a) *A toy template propagated by a rotational flow.*

(b) *Inpainting results: different interpolation methods.*

Figure 4: **Interpolation artifacts.** Bilinear interpolation induces blurring; nearest-neighbor interpolation creates distortion. Our combination achieves the most realistic result.

can result in blurry results, as the L^2 norm in (1) can result in a blurry template as a result of temporal averaging in (2). To mitigate these effects, we solve an energy minimization problem for the inpainted radiance $P_t : M_t \rightarrow \mathbb{R}^k$ that interpolates between the template f and raw image values from other frames that map to the mask, through the L^1 distance that reduces temporal blurring, as follows:

$$E_{img}(P_t) = \int_{M_t} \Big[|P_t(x) - f^*(w_t^{-1}(x))|_2^2$$
$$+ \beta \sum_{i=1}^{n} |P_t(x) - I_i(w_{ti}(x))|_1 \mathbb{1}_t(w_{ti}(x)) \Big] \, dx,$$
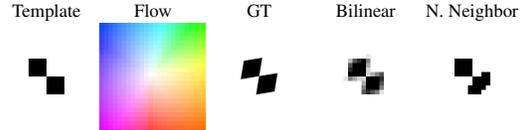(4)

where the first term measures fidelity of the inpainting to the template mapped into M_t, and the second measures fidelity of the inpainting to mappings of other frames into M_t. $\mathbb{1}_t(w_{ti}(x))$ indicates whether $w_{ti}(x)$ is visible in frame i.

Let $\{t_x^1, \ldots, t_x^{m_x}\}$ denote the m_x frames where $w_{ti}(x)$ maps into M_t. From [16], (4) has a closed form minimizer:

$$P_t^*(x) = \text{median}\{I_{t_x^1}(w_{tt_x^1}(x)), \cdots, I_{t_x^{m_x}}(w_{tt_x^{m_x}}(x)),$$
$$f^*(y) - \frac{m_x}{2}\beta, f^*(y) - (\frac{m_x}{2}+1)\beta, \cdots, f^*(y) + (\frac{m_x}{2})\beta\},$$
(5)

where $y = w_t^{-1}(x)$. The inpainting is a *temporal* median filtering of pixels from different frames that map into the mask as well as the template appearance. Figure 3 shows an example of the reduced blurriness of this L^1 formulation.

We now discuss the evaluation of the images at transformed pixels in (5), which requires interpolation. This choice is key to producing a visually plausible result. Common choices are nearest neighbors or bilinear interpolation. In Figure 4a, we propagate a toy template by a rotational optical flow and show the effects of each scheme. Bilinear preserves the shape but edges are blurred since the pixel value is a linear combination of white and black pixels around the endpoint. Nearest-neighbor preserves pixel values and thus is not blurred. However, rounding endpoints to nearest pixels lead to geometric distortion. Defects from both

interpolation schemes are further amplified and propagated throughout the whole video in flow guided inpainting.

We propose a simple yet effective solution to this long-ignored problem in flow-based inpainting that our framework naturally suggests. We apply bilinear interpolation in computing the template (2) to avoid geometric distortion. After obtaining a smooth template with well-aligned warps, nearest-neighbor interpolation is applied in computing the inpainting in (5) to preserve rigid edges. This achieves the result with the least artifacts. As suggested by Figure 4b, only using bilinear interpolation leads to a blurry result while only using nearest-neighbor interpolation leads to distortion. The best result is obtained by our combination.

After the inpainting described above, there may be some masked pixels that are not filled, as they correspond to points in the scene that never revealed in the entire video. To fill these pixels, we use DeepFill [35], following [29, 5].

5. Experiments

To the best of our knowledge, currently, there is no video inpainting benchmark dataset, but only for object segmentation (DAVIS [21]). While it is possible to use the segmented masks from DAVIS for video inpainting (e.g. [29, 5]), there is no ground truth and evaluation relies on user studies.

As a complement, many methods [29, 38, 11, 37] compose moving objects or masks over background videos, so inpainting accuracy can be numerically evaluated. Since data are created in different ways (some are not publicly released), a direct quantitative comparison is infeasible. Therefore, we introduce a new dataset under this setting, called *Foreground Removal* dataset, with quantitative evaluation protocols measuring inpainting accuracy and temporal consistency. The dataset will be made publicly available.

DAVIS [21] contains a total of 3455 frames in 50 videos with pixel-wise per-frame annotations. The task is to remove annotated moving objects. We perform a user study

	DAVIS				Foreground Removal				
	TPSNR↑	TSSIM↑	Avg↑	#1↑	TPSNR↑	TSSIM↑	PSNR↑	SSIM↑	FID↓
Ours+F+S	29.08	0.881	-	-	**34.58**	**0.935**	**29.15**	**0.857**	**1.042**
Ours+S	**30.86**	**0.922**	**7.24**	**581**	30.14	0.895	28.17	0.825	1.179
DFG[29]	27.58	0.852	6.89	457	30.74	0.886	28.28	0.803	1.376
ILA[38]	30.59	0.894	5.12	95	31.35	0.910	25.64	0.769	1.292
OnionPeel[19]	26.24	0.817	-	-	29.31	0.822	28.71	0.833	1.051
STTN[37]	29.04	0.874	-	-	34.07	0.926	27.62	0.817	1.136
DeepFill[35]	19.56	0.554	3.96	24	20.77	0.639	19.31	0.568	2.546
FGVC [5]	30.71	0.916	-	-	32.89	0.936	31.76	0.886	0.833
Ours+FGVC flow	**30.94**	**0.921**	-	-	**34.56**	**0.947**	**31.89**	**0.894**	**0.802**

Table 1: **Quantitative Results**. On both datasets, our method achieves the best performance in terms of temporal consistency and inpainting quality. Our results are more preferred in the user study. F: FlowNet2 [9]; S: SobolevFlow [33].
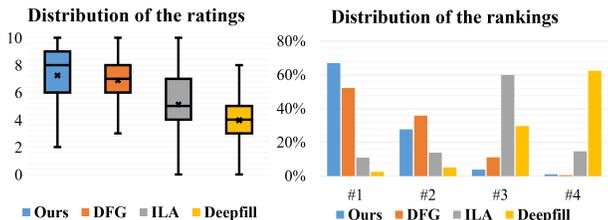

Figure 5: **User study on DAVIS.** Ours achieves highest average and median rating. Ours also receives the most # 1 ranking.

by inviting 24 volunteers from both inside and outside the field to rate the inpainting quality of each video from 1 to 10. Videos are displayed at 15 fps and users can stop, replay, and zoom-in freely. The ordering of the methods is randomly permuted (not known to the users). Each user is required to rate at least 15 sequences and we collected 871 results in total. We also evaluate temporal consistency following [38] by measuring visual similarity of the inpainted region in adjacent frames, labeled as TPSNR and TSSIM.

Foreground Removal dataset includes 25 composed videos ranging from 29 to 90 frames whose backgrounds are collected from Youtube. We paste moving foreground from DAVIS and SegtrackV2 [25] to background videos. The dataset contains representative challenging cases including viewpoint change, deforming background, illumination change, fast zooming-in, motion blur, image noise (e.g. rain), and constant (e.g. overexposed) regions. We evaluate inpainting accuracy by computing PSNR, SSIM, and Frchet Inception Distance [6] to the ground truth, and evaluate temporal consistency as in DAVIS.

Comparison: We compare our approach with state-of-the-art methods. They are: flow-guided *DFG* [29] and *FGVC* [5]; end-to-end *ILA* [38], *OnionPeel* [19], *STTN* [37]; and single image *DeepFill* [35]. For the user study, we choose DFG, ILA and DeepFill since they are representative of each category of method. Due to author-released code of [38, 29, 19] operating at different resolutions, we resize all results to the same resolution for a fair numerical and visual comparison. As described in Section 3.3, we ap-

ply SobolevFlow [33] for flow refinement (*Ours+S*). To be comparable to [29], we use FlowNet2 [9] for flow initialization (*Our+F+S*). We compare to FGVC [5], which uses a more advanced flow method. For this comparison, we initialize our method with the flow used by [5].

5.1. Results

DAVIS Dataset: Figure 6 shows representative visual results on DAVIS. Ours are more visually plausible than competing methods. DFG is vulnerable to distortion due to frame-to-frame propagation and nearest-neighbor interpolation. OnionPeel and ILA have blurry results (frequently observed in other learning-based methods [37, 11, 14], too). Our method preserves rigid object appearances, showing the effectiveness of the L^2-L^1 optimization and interpolation strategy. Ours also significantly outperforms the competition in videos with long-term occlusion (e.g. Figure 1) since the method is less vulnerable to flow error accumulation.

Figure 5 summarizes the user study. Our method has the best average user rating. It also receives the highest count of number one rankings (tie allowed). In 67% of the ratings, ours rank the best among the four methods. Table 1 shows quantitative results. *Ours+S* achieves the best temporal consistency. This is presumably because the region-based formulation of SobolevFlow provides a more consistent background motion estimation. Initialized by the same flow as [5], we achieve better temporal consistency, which shows that the scene consistency in our method improves over even more advanced optical flow (also see Figure 1).

Foreground Removal Dataset: Figure 9 and Table 1 show qualitative and quantitative results on the dataset. Ours obtains visually plausible results. Since the dataset contains more challenging background motion and FlowNet2 has a stronger capability to handle complex motion, Ours+F+S achieves dominant performance. Similar to DAVIS, our method improves over [5].

We want to highlight two cases in Figure 9: in the car scene (2nd column), our approach successfully handles two foreground objects with different motions; in the horse scene (5th column) our approach handles strong cam-

Figure 6: **Comparison on the DAVIS dataset.** With our scene template and corresponding optimization and interpolation scheme, our method obtains the most realistic results. *Better viewed zoomed in.*
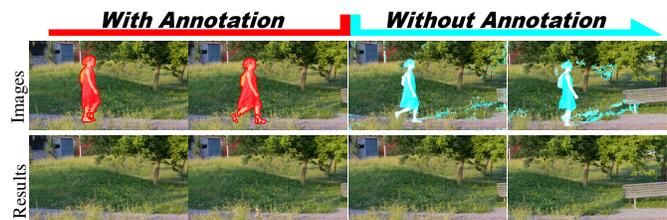

Figure 7: **Sample fixed region removal from a dynamic scene (from DAVIS).** Animation in the supplementary materials.
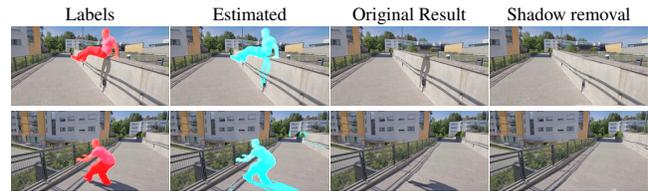
era zooming, a challenging case to video inpainting.

Fixed Regions (DAVIS): The literature considers removal of content within a fixed region of video in scenes with dynamic objects. Our formulation (1) can, in principle, be applied to this case (Figure 7). However, for high accuracy, one requires additional occlusion reasoning in optical flow, which will be subject of future work; the focus of the current paper is to illustrate the benefits of the scene template. Even with our current methodology, our results are comparable to [5], the state-of-the-art for fixed regions: PSNR 28.02 vs 28.20, SSIM 0.959 vs 0.957 on DAVIS following the experimental setups of [5]. We provide detailed discussions in the supplementary materials.

5.2. Further Application and Discussion

Incomplete annotations: One task that has not been considered by previous work is handling incomplete or in-

(a) *Inpainting with missing annotations.*

(b) *Automatic shadow estimation and removal.*

Figure 8: **Inpainting with incomplete annotations.** Our method can be applied to correct incomplete annotations. It estimates missing annotations by (6) and performs inpainting. *Red: given annotations; blue: estimated masks.*

perfect annotations. Existing methods (e.g. [7, 29, 38, 5]) assume perfect annotations. In practice, masks often come from user annotation or segmentation algorithms, and so annotations may not be available for the whole video or may contain errors. Figure 8 shows two examples. In the first,
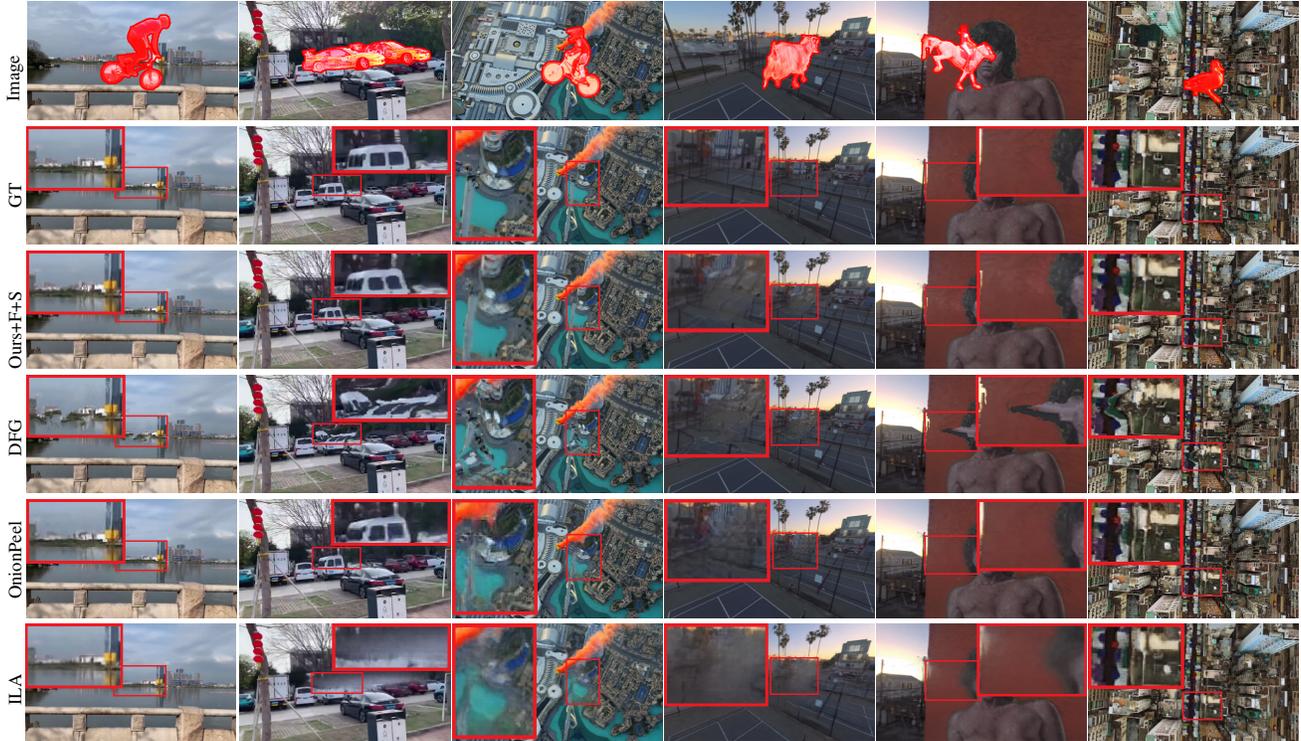
Figure 9: **Comparison on Foreground Removal dataset.** Our novel dataset contains challenging cases including irregular background motion, illumination change, fast zooming-in, motion blur, image noise, and constant regions. Our method obtains the most visually plausible and temporally consistent results. *Better viewed zoomed in.*

Figure 10: **Multi-frame aggregation improves regularity** Even if using the same optical flow, our method shows stronger regularity than frame-to-frame propagation.

masks from only the first 10 frames are provided; in the second, shadows are not included in masks. Our method can estimate foreground masks by thresholding the residual between the scene template (computed from available noisy annotations) and images:

$$R_t = \{|I_t(x) - f(w_t^{-1}(x))|_2^2 > \alpha\}, \quad \alpha = 0.1. \quad (6)$$

Our method infers missing annotations and corrects incorrect annotations, and so to the best of our knowledge, the first to inpaint despite inperfect annotations. This makes fully automated foreground removal algorithm possible, which can be a future direction of research.

Multi-frame aggregation improves regularity: In this experiment, we use the same optical flow as DFG (without

refinement), so the only difference is multi-frame aggregation v.s. frame-to-frame propagation. On DAVIS, TPSNR rises from 27.58 to 30.53 and TSSIM rises from 0.852 to 0.966, showing stronger temporal consistency. We can even observe results with stronger spatial regularity, shown in Figure 10. This shows the advantage of our scene template.

6. Conclusion

We proposed a novel method for flow-guided video inpainting by introducing the *scene template*, which is a 2D representation of the background. The method aggregates appearance information across frames into the scene template by non-rigid maps, which are solved jointly, then maps the template to the images for inpainting. This results in more plausible and temporally consistent flows than existing flow-based methods as the maps must be consistent with the *scene*. We proposed a simple interpolation scheme, which significantly reduced inpainting artifacts. Experiments showed that our method achieved state-of-the-art results on two datasets in terms of inpainting accuracy and temporal consistency. Our method can also handle missing and noisy user mask annotations.

References

[1] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010. 2

[2] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9066–9075, 2019. 2

[3] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 1, 2

[4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999. 2

[5] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf. Flow-edge guided video completion. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 5, 6, 7

[6] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 6

[7] 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, 7

[8] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):107:1–107:14, 2017. 2

[9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 4, 6

[10] J. D. Jackson, A. J. Yezzi, and S. Soatto. Dynamic shape and appearance modeling via moving and deforming layers. *International Journal of Computer Vision*, 79(1):71–84, 2008. 2

[11] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon. Deep video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019. 2, 5, 6

[12] D. Lao and G. Sundaramoorthi. Minimum delay moving object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4250–4259, 2017. 2

[13] D. Lao and G. Sundaramoorthi. Extending layered models to 3d motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451, 2018. 2

[14] S. Lee, S. W. Oh, D. Won, and S. J. Kim. Copy-and-paste networks for deep video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4413–4421, 2019. 2, 6

[15] A. Li, S. Zhao, X. Ma, M. Gong, J. Qi, R. Zhang, D. Tao, and R. Kotagiri. Short-term and long-term context aggregation network for video inpainting. *arXiv preprint arXiv:2009.05721*, 2020. 2

[16] Y. Li and S. Osher. A new median formula with applications to pde based denoising. *Communications in Mathematical Sciences*, 7(3):741–753, 2009. 5

[17] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 2

[18] 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

[19] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim. Onion-peel networks for deep video completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4403–4412, 2019. 1, 2, 6

[20] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–69. IEEE, 2005. 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 *Computer Vision and Pattern Recognition*, 2016. 2, 5

[22] M. Strobel, J. Diebold, and D. Cremers. Flow and color inpainting for video completion. In *German Conference on Pattern Recognition*, pages 293–304. Springer, 2014. 2

[23] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1768–1775. IEEE, 2012. 2

[24] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015. 2

[25] D. Tsai, M. Flagg, and J. M.Rehg. Motion coherent tracking with multi-label mrf optimization. *BMVC*, 2010. 6

[26] C. Wang, H. Huang, X. Han, and J. Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5232–5239, 2019. 2

[27] J. Y. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994. 2

[28] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004. 2

[29] R. Xu, X. Li, B. Zhou, and C. C. Loy. Deep flow-guided video inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 5, 6, 7

[30] Y. Yang, B. Lai, and S. Soatto. Dystab: Unsupervised object segmentation via dynamic-static bootstrapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2826–2836, 2021. 2

[31] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019. 2

[32] Y. Yang and G. Sundaramoorthi. Modeling self-occlusions in dynamic shape and appearance tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 201–208, 2013. 2

[33] Y. Yang and G. Sundaramoorthi. Shape tracking with occlusions via coarse-to-fine region-based sobolev descent. *IEEE transactions on pattern analysis and machine intelligence*, 37(5):1053–1066, 2014. 4, 6

[34] Y. Yang, G. Sundaramoorthi, and S. Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4408–4416, 2015. 2

[35] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 1, 2, 5, 6

[36] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019. 2

[37] Y. Zeng, J. Fu, and H. Chao. Learning joint spatial-temporal transformations for video inpainting. In *The Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 5, 6

[38] H. Zhang, L. Mai, N. Xu, Z. Wang, J. Collomosse, and H. Jin. An internal learning approach to video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2720–2729, 2019. 1, 2, 5, 6, 7