Occlusion-Aware Video Object Inpainting (Supplemental material)

Lei Ke[1] Yu-Wing Tai[2] Chi-Keung Tang[1]
[1]The Hong Kong University of Science and Technology [2]Kuaishou Technology

In this supplementary material, we provide more details on the object occlusion masks generation process for the Youtube-VOI benchmark, the implementation of our VOIN model, discussion on the influence of initial segmentation quality to the final object inpainting results and additional flow completion results comparison with [1, 2] in Figure 6. Please **refer to the attached video file** VOIN-results-comparison.mp4 for the extensive qualitative video results comparison with state-of-the-art video inpainting methods [2, 1, 3, 4].

1. Dataset Generation

Video Object Occlusion Masks Generation To produce both occlusion masks and visible masks for video objects in our Youtube-VOI benchmark (Figure 6 of the paper), and simulate the high-fidelity movements of overlapping video objects, our occlusion mask generation algorithm contains two steps: **1)**We adapt the free-form video mask generation algorithm in [5, 6] to produce object-like moving masks, which are smoothly moving random strokes with continuous velocity and acceleration. The control points inside the object-like masks are moved with a random probability to simulate object deformation. The object-like shape of the moving masks are controlled by adjusting the brush width and radius. **2)** We utilize the moving video object masks provided in Youtube-VOS and the generated masks in the previous step to simulate object overlapping situations, and only preserve the overlapping video cases with degrees of occlusion from 10% to 70% for each video frame.

2. Implementation Details

VOIN Network We use PyTorch1.1 to implement the VOIN model. For training the occlusion-aware shape completion module, we follow the structure design in [4, 7] to stack the it by eight layers with 1×1 convolutions for producing the query, key, value embeddings of the image patches and 3×3 convolution for the final fusion layers. We adopt LeakyReLU as activate function and bilinear interpolations for upsample operations on feature map. For the flow completion module, we adopt the pseudo ground truth flow computed from the original, un-occluded videos using RAFT [8]. The flow encoder-decoder with Unet structure has kernel size 3 for all convolutional layers, dilation values of 2, 4, 8, 16 for the intermediate dilated layers, and stride 2 for the first and third convolutional layers for down-sampling. For the encoder-decoder of our occlusion-aware inpainting generator, the proposed occlusion-aware gating layers have two continuous convolutions with kernel size 3. The Multi-Class Discriminator with STAM consists of 3D strided convolution layers with kernel sizes (3, 5, 5) and strides (1, 2, 2) followed by LeakyReLU, where we use the global average pooling to reduce the spatial size before the final classification output.

Details of the transformer layer in shape completion. The input image of the 8-layer transformer is resized to 320×180, where each transformer layer has a multi-head structure to deal with patch scales $[(80, 45), (64, 36), (32, 18), (16, 9)]$ respectively. For training the shape module, we sample 10 frames from a video consecutively (T = 10) with total video batch size is 12 for each iteration, where learning rate starts with 1e-4 and decays with factor 0.1 every 50k iterations. The transformer outputs T(10)$\times$h(180)$\times$ w(320) features with complete object mask for each frame. Different from DETR [9] with large input image size (800$\times$ 1333) for object detection and ImageGPT [10] training on ImageNet (over 14 million images) and 100 million unlabeled web images for universal image representation learning, we train the shape transformer on our constructed 65-category Youtube-VOI with 5K video clips for amodal shape completion.

Training Details We train using the Adam optimizer with a learning rate of 0.0001. During training, we first jointly train the object shape completion and flow completion modules, which takes about 1 day on the Youtube-VOI training set with 6 RTX 2080 Ti GPUs (total video batch size is 12). After the two modules converge, we combine them with the proposed occlusion-aware gated generator for continual training, which takes about another 4 days to finish on two NVIDIA RTX 2080

Ti GPUs. The PyTorch data and model parallel scheme is used to speed up the training process of our model. We set the loss weights denoted in Section 3.4 of the paper as $\lambda_{flow} = 5.0$, $\lambda_{app} = 5.0$, $\lambda_1 = 1.0$, $\lambda_2 = 10.0$, $\lambda_3 = 20.0$, $\lambda_4 = 2.5$, $\lambda_5 = 0.2$. Note that the algorithm of flow-guided pixel propagation has no trainable parameters.

3. Additional Video Results Comparison

In the **attached video file**, we provide extensive video results comparison between our VOIN and state-of-the-art video inpainting methods DFVI [1], LGTSM [3], FGVC [2] and STTN [4] for recovering the appearance of the occluded video objects under complex occlusion scenarios. The video file contains four parts: 1) Video **Object Inpainting** Results Comparison with free-form occlusion masks on our proposed Youtube-VOI test set. 2) Video Scene **De-occlusion** Results Comparison for removing the occluding video objects on the videos collected from both the Youtube-VOS and DAVIS2017. 3) Influence of Initial Segmentation Quality. 4) Video Scene **Manipulation** Results Comparison to reverse the positional order of the original occluder and occludee in the presence of large occlusion holes. Due to video size limitation, we suggest to adjust the progress bar for playing videos manually for better view and comparison.

4. Experiments on Influence of Initial Mask Segmentation

The following shows the graceful degradation of VOIN inpainting results against visible mask segmentation errors. Figure 1, 2, 3 and 4 show sample frames. Please watch the attached video for the whole sequence comparison. Our mask degradation attacks include:

1) Inaccurate visible masks are produced by dilating and eroding along the detected mask boundary by 6, 14 and 20 pixels, which respectively result in deviation at about 11%, 25%, and 36% from the original mask. In Figure 1, our final inpainting results are still reasonable up to deviation rate 30%-40% against masks dilation, which is the typical error range produced by existing SOTA video instance segmentation methods [11, 12] (usually less than 3 pixels). In Figure 2 the erosion squeezes the occluded inpainting regions, causing "leakage" of the occluding object (person) into the completed occluded object (car). Thus, we suggest to dilate along the occluding object's mask boundary by a small amount (about 3 pixels) in implementation for getting rid of small erosion errors in the initial segmentation prediction.

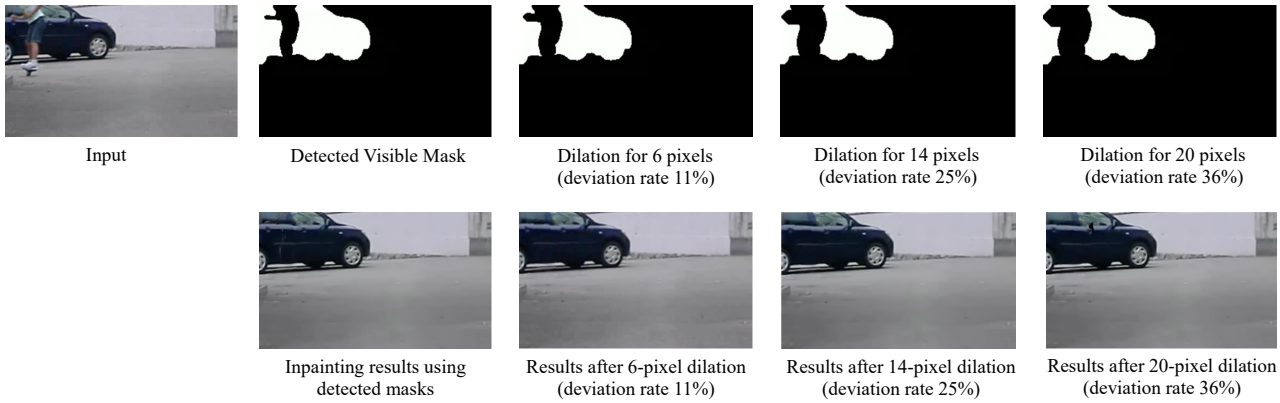

Figure 1. Visual results of VOIN with mask dilation up to 11%, 25%, and 36% deviations from the original mask, where our final inpainting results are still acceptable up to about 30%-40% against mask dilation errors.

2) Extreme attacks by adding gross segmentation errors in the intermediate five frames: we randomly sample three of them and show the results in Figure 3. This experiment reveals that even in the presence of intermediate temporal shaking artifacts during the visible mask detection, our final object inpainting results are not adversely affected thanks to our long-range object shape associations.

3) Extreme attacks by cutting off a large region of the detected visible masks (left part of each frame) at the beginning 5 continuous frames. In Figure 4, our shape completion fails in completing the whole car shape in this situation (with over 50% corruption on the detected masks). In this extreme case, our visual result is comparable to and thus can be regarded as lower-bounded by SOTA conventional video inpainting method STTN [4] given an arbitrary input hole, as compared in Figure 5.

The above demonstrates not only the main advantage of occlusion and object-awareness in video inpainting as compared to conventional video inpainting, but also the robustness of our VOIN even in presence of gross errors in visible mask segmentation.

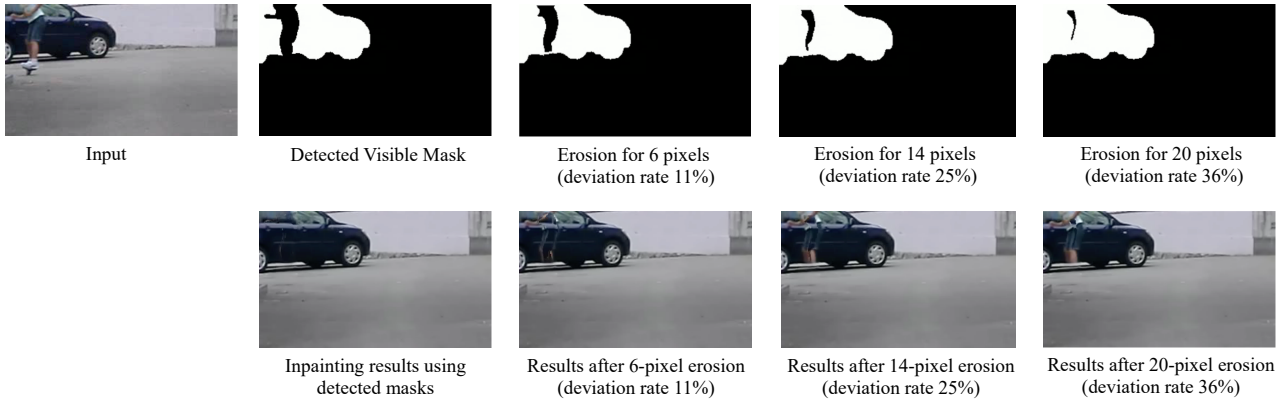

Figure 2. Visual results of VOIN with mask erosion up to deviations 11%, 25%, and 36% from the original mask, where the erosion squeezes the occluder's mask causing leakage into to the completed car.

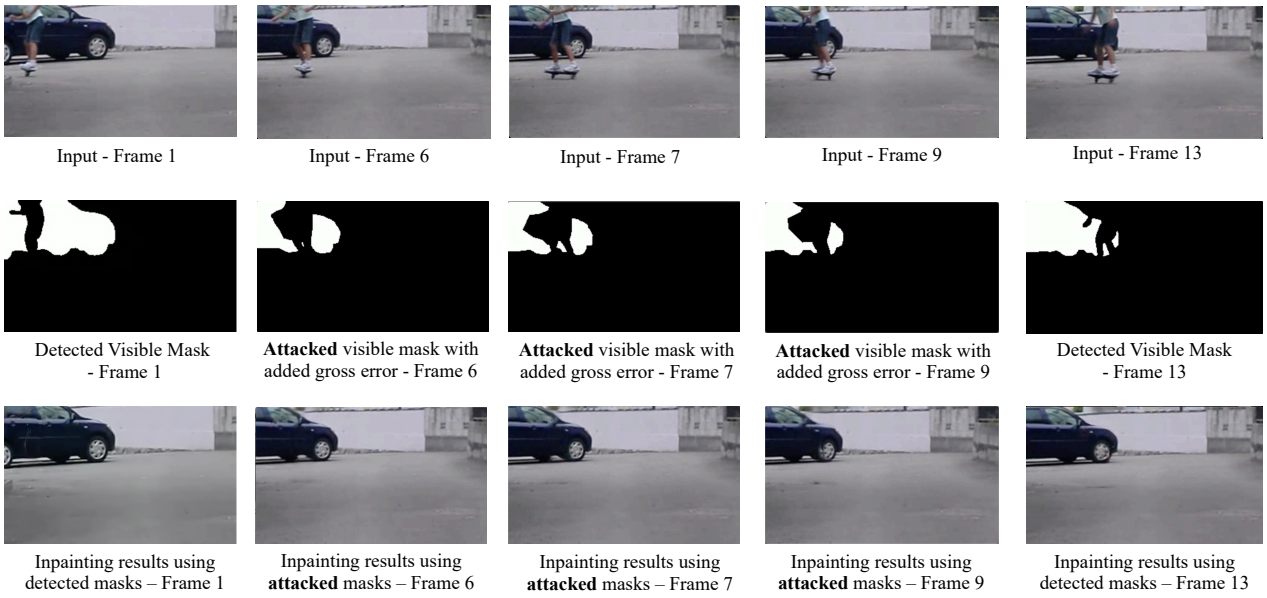

Figure 3. Visual results of VOIN with extreme attacks by interfering detected masks with **gross segmentation errors in the intermediate 5 frames**. We provide the whole sequence in attached video file.

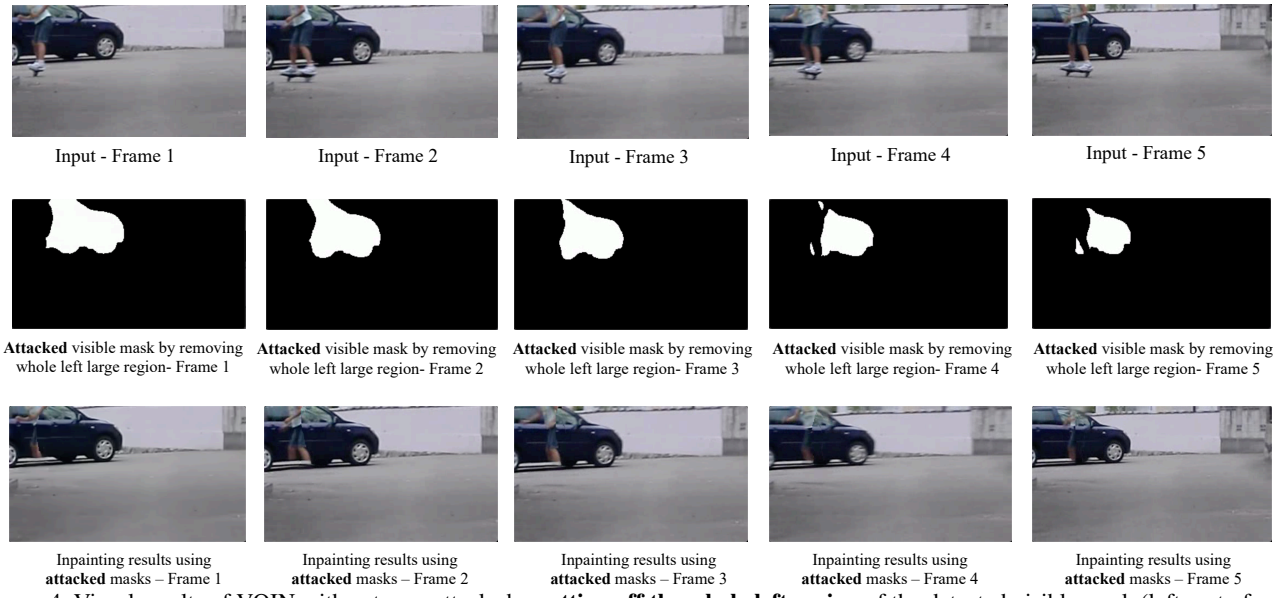

Figure 4. Visual results of VOIN with extreme attacks by **cutting off the whole left region** of the detected visible mask (left part of each frame) at the beginning 5 continuous frames.

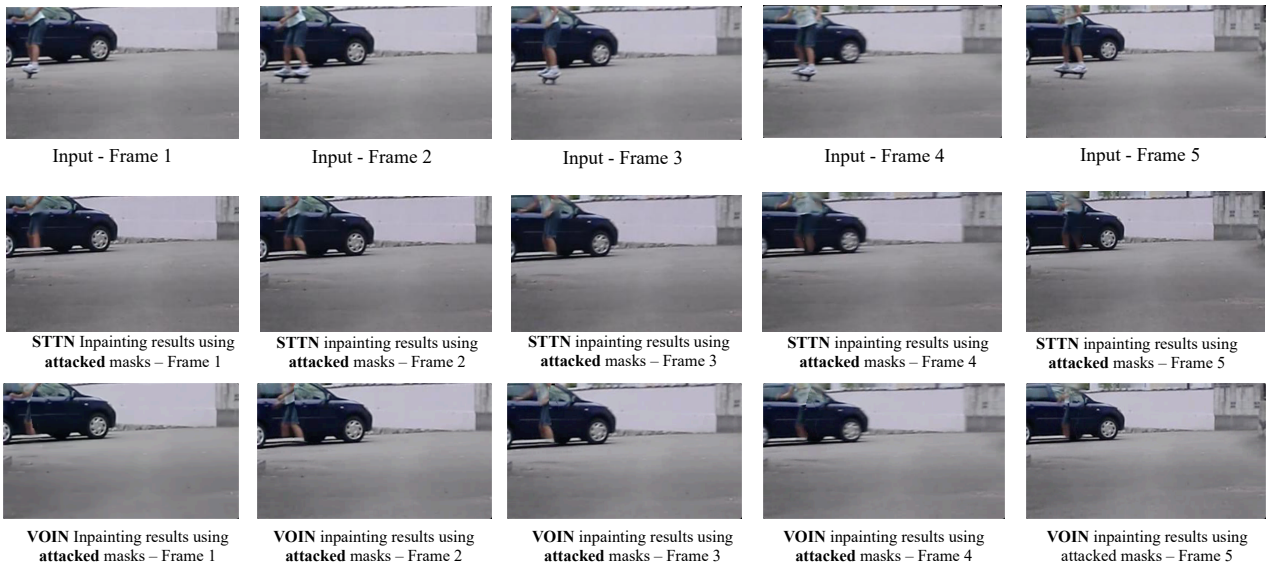

Figure 5. We also compare the results of our VOIN under the extreme attacked masks with STTN [4] in the situation of Figure 4 where **the whole left large region is cut off** from the detected visible masks (whole left part of the car). The recovered object shape is inaccurate and the performance of VOIN can be regarded as lower bounded by conventional state-of-the-art video inpainting method STTN [4].

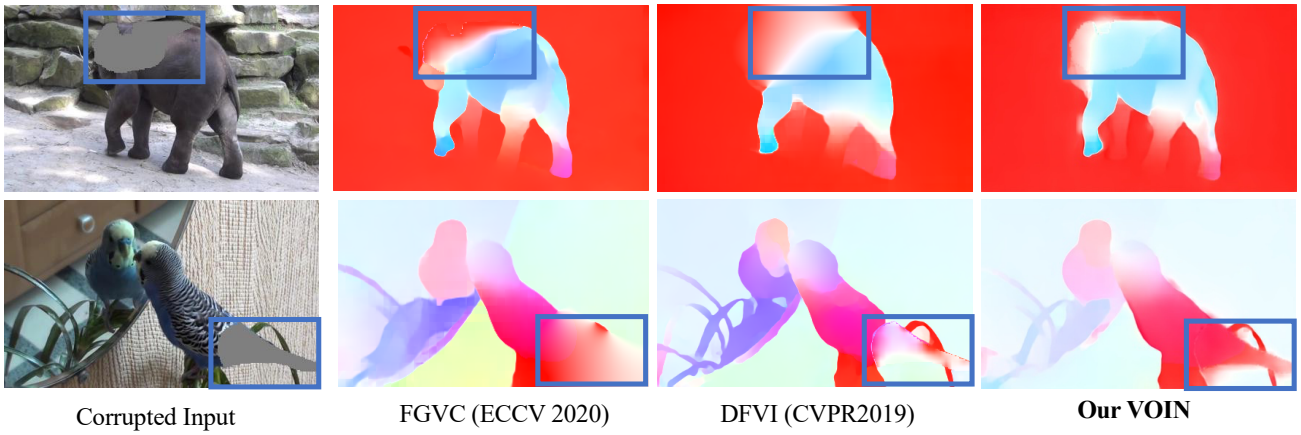

Figure 6. Additional object flow completion results comparison with DFVI [1] and FGVC [2].

References

[1] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, 2019. 1, 2, 4

[2] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, 2020. 1, 2, 4

[3] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Learnable gated temporal shift module for deep video inpainting". In *BMVC*, 2019. 1, 2

[4] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, 2020. 1, 2, 4

[5] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *ICCV*, 2019. 1

[6] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *CVPR*, 2019. 1

[7] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *ECCV*, 2020. 1

[8] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 1

[11] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020. 2

[12] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 2