Deep Video Inpainting: A Spatio-temporal Approach for Video Processing

David Leonardo Manrique Lesmes
Universidad de los Andes
Cra. 1 18a-12, Bogotá, Colombia
dl.manrique@uniandes.edu.co

Abstract

Nowadays, videos are of vital importance in society. Today for this reason the area of artificial intelligence has applied different techniques and tasks in these, such as the task of removal of objects present in videos. This is called video inpainting and can be studied as static mask removal and dynamic mask removal. The current models that handle this task may or may not include temporal information, this depends on whether they are designed for images or for video. The objective of this work is to demonstrate the difference in both quantitative and qualitative results of including temporal information when performing both types of tasks. With this in mind, experiments were carried out with four state-of-the-art moles in both tasks. The results show that temporal information is vital if qualitatively good results are to be obtained.

1. Introduction

Video inpainting is a computer vision technique that involves filling in missing or damaged parts of a video sequence based on the surrounding information. The result is a visually plausible and coherent output video that is consistent with the overall motion and appearance of the original video [12] [17]. The remotion of unwanted objects and video restoration are just two examples of the many real-world uses for video inpainting, also known as video completion [17]. Despite significant advances in deep learning-based inpainting of a single image, the extra time dimension in the video domain makes it difficult to apply the same techniques in video [8]. Additionally, when working with videos, the computational cost increases exponentially and due to the camera motion and the complex movement of objects, this task is highly challenging [17] [18]. Taking into account all of the above, video inpainting is a work of great interest, but it still presents challenges that are not easy to solve.

One approach that some authors used to tackle this problem is to directly perform image inpainting on each frame independently [10]. However, this kind of solution ignores motion regularities originating from the dynamics of the video, causing the creation of artifacts and videos without temporal consistency [8] [10]. Nevertheless, the majority of current video inpainting algorithms use a patch-based optimization task to formulate the problem, filling in missing regions by sampling spatial or spatio-temporal patches of the known regions, then solving a minimization problem [17]. Another typical methodology to study the problem is to consider this task as a pixel propagation problem. This involves the use of optical flow, and it is expected that this type of approach will help preserve temporal coherence [10]. Finally, the most recent methods are based on transformers and are commonly referred to as the use of pixel-wise attention mechanisms that allow us to improve performance in this task [18].

As previously mentioned, the video inpainting problem is composed of multiple tasks and has different applications. However, in this work we are going to focus on a very particular one: the removal of artifacts present in the videos. This can be studied in two ways: by removing objects present in the videos or using some algorithms, one can create artifacts in the videos and then find a way to remove them; this approach is also known as removing static masks from videos [18]. It should be noted that this second way reduces the complexity of the real problem, but it is considered as a valid initial approach to this kind of techniques. An example of how this type of approach would be carried out is shown in Figure 1; there it can be seen that a mask is created in the input image and the objective is to remove it. In our case, we are going to study the problem using both approaches.

Now, the objective of the project is to evaluate the influence of adding temporal information in order to improve the performance in the tasks of removing static and dynamic masks from videos.

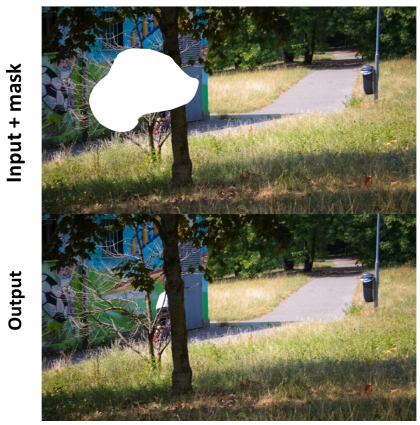

Figure 1: The top row show sample frame with white mask denoting static mask. The bottom row shows the desired completion results

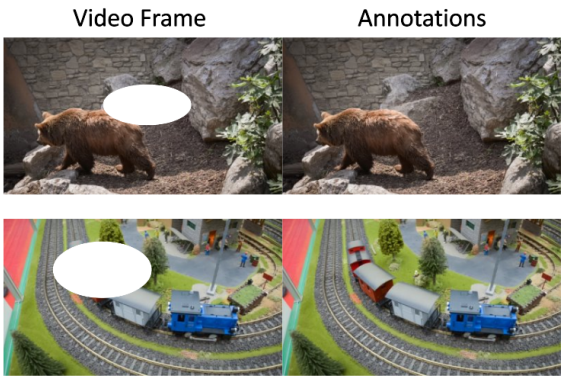

Figure 2: Example of the annotations to be used in the project.

2. Related Work

Various machine learning methods, from more conventional ones to Deep Learning ones, have been used in the past to study the video inpainting task. The most important datasets and methods that have been used in this task are described below.

2.1. Datasets

In general, the datasets created for the video inpainting task contain a variety of videos from different scenarios. In addition, the quality of the videos usually varies depending on the dataset.

Now, some of the benchmarks that have been most used in this task are DAVIS [1] and YouTube-VOS [9] (Video Object Segmentation), which contain 50 and 4000 videos respectively. Both datasets are designed for the task of video inpainting and have a preestablished division. Also, the datasets contain diverse scenarios, such as occlusions, motion blur, and dynamic backgrounds [13]. In the same way, there are several versions of the mentioned datasets, which are created periodically in order to update and improve the data. In the DAVIS dataset, the annotations are segmentation masks for one or multiple objects in each frame of the videos. In the Youtube-VOS dataset, there is a similar format. Nevertheless, in this dataset there is a classification of each pixel for each frame, taking into account all objects and the background of the frame. However, the big difference between the datasets lies on the quality of the videos and their origin. While the DAVIS dataset has videos at 480p and 1080p with 30 frames per second [1], Youtube-VOS videos have varying resolutions, ranging from 360p to 1080p, and varying frame rates, ranging from 24 to 60 frames per second [9]. Other datasets also used are Adobe-UW and Middlebury [13].

2.2. Families of methods

Over time, various approaches have been considered for the task of video inpainting. Traditional methods have been proposed, and they are patch-based and exemplar-based. This involves using information from the surrounding regions of the missing area and using them to fill in the gaps. However, these methods are not very effective and cannot be used for videos with complex scenes. To obtain better results, a lot of deep learning-based methods have been proposed. They can be summarized in the use of GANs, visual transformers, and, more recently, diffusion models [16].

Some of the state-of-the-art methods based on deep learning used for the video inpainting task are described below.

2.2.1 GANs

For some time, to adapt GANs for video inpainting, the researchers have created a variety of architectures, including spatio-temporal GANs and motion-compensated GANs. Spatio-temporal GANs base their operation on the use of both spatial and temporal information. An example of this type of architecture is the one proposed by Ya-Liang Chang et al., where a new loss function is proposed that takes into account both types of information [4]. As for Motion-compensated GANs, they use the optical flow to estimate the position of the objects between each frame and use this type of information to generate content that matches the motion patterns in the surrounding regions. An example of this approach is given by Chang et al. [5].

2.2.2 Visual Tranformers

As far as it is known, the method with the best results on DAVIS and Youtube-VOS datasets is End-to-End framework for Flow-Guided Video Inpainting (E2FGVI)

proposed by Zhen Li [10]. This method is based on the use of a visual transformer and is composed of 3 modules: flow completion, feature propagation, and content hallucination. Another visual transformer with similar results is the Spatial-Temporal Transformer Network (STTN) proposed by Yanhong Zeng et al [18]. In this method, spatio-temporal information is used to fill in the missing regions. Both methods have code available on GitHub [1] [2].

Other model based on transformers that shows good results in this task is FuseFormer proposed by Liu et al [11]. This work includes fine-grained information to improve the performance of the method. Likewise, this work has its code available on GitHub [3]. Finally, there are also many other models of visual transformers designed for this task, such as [8].

2.2.3 Diffusion Models

In recent years, diffusion models have been widely used for generation tasks. An example of that is the fact that these models have been used for image and video inpainting. For image inpainting we can find the LDM model proposed by Rombach et al, that achieves state-of-the-art result in this task performing the optimization problem in the latent space [15]. The code for this model can be found at GitHub [4]. Additionally, we have that one model for video procesing was proposed by Tobias Höppe et al. called RaMViD [7]. In that paper, they used 3D convolutions to extend image diffusion models to videos and introduced a novel conditioning technique for training. It is important to clarify that this model was mainly designed for video prediction and infilling tasks, however, it can also perform video inpainting task. The RaMViD code can be found at GitHub [5]

It is significant to note that there are only a few reported examples of these models because they have not been thoroughly investigated regarding video inpainting.

3. Approach

To study the video inpainting problem for stationary masks and moving masks, the performance of several state-of-the-art models designed for both image inpainting and video inpainting was evaluated. As such, it was decided to include models designed for image inpainting in order to evaluate the influence of including temporal information when dealing with the video inpainting problem. The models chosen for this work were the following: ZeroRA based Incremental Transformer Structure (ZITS) [6], Latent Diffusion Model (LDM) proposed by Rombach et al [15], Spatial-Temporal Transformer Network (STTN) [18] and FuseFormer (FF) [11]. The first two were designed for image inpainting and the other two for video inpainting. In the case that we had an image inpainting model, each of the frames was processed as a single image and the result was the video that was produced with these frames.

3.1. Baseline

The first approach to the problem consisted of evaluating the Latent Diffusion model (LDM) [15] and ZeroRA based Incremental Transformer Structure (ZITS) [6] using the test data of both type of problems (removing stationary and moving masks). It was decided to use these models since they are the state-of-the-art in the problem of inpainting in images. In view of the above, each frame was processed separately, so no temporal information was taken into account. The results of the models are shown in Table 1. In Figure 3 the outputs for different frames of static task are shown, in figure 4 the outputs of dynamic task are shown.

Table 1: Results of evaluating both models (LDM and ZITS) in both tasks.

Model	**Stationary Masks**			**Moving Masks**
	PSNR (dB) (↑)	**SSIM (↑)**	**VFID (↓)**	**VFID (↓)**
LDM	20.5	0.932	0.156	0.401
ZITS	19.5	0.952	0.136	0.255

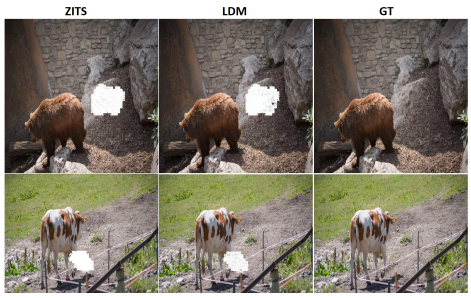

Figure 3: Baseline results for static task

3.2. Proposed Method

After obtaining the results of the evaluation of the image inpainting models in both tasks, a fine-tuning process was performed using the training and validation data of the static mask removal problem. In the case of the LDM model, variations were made in the learning rate and batch

[1] GitHub repository of E2FGVI: https://github.com/MCG-NKU/E2FGVI

[2] GitHub repository of STTN: https://github.com/researchmm/STTN.

[3] GitHub repository of FuseFormer: https://github.com/ruiliu-ai/FuseFormer

[4] GitHub repository of LDM: https://github.com/CompVis/latent-diffusion

[5] GitHub repository of RaMViD: https://github.com/Tobi-r9/RaMViD

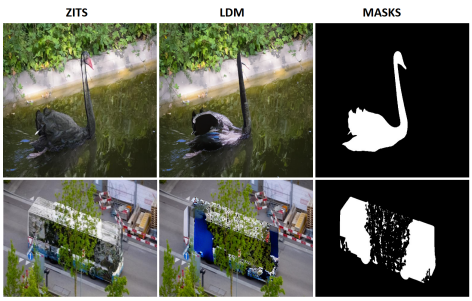

Figure 4: Baseline results for dynamic task

size, while for the ZITS model, the learning rate and the optimizer were modified. In addition, the FF and STTN models were evaluated using the evaluation sets specific to each of the proposed tasks. This allowed to obtain a comparison between the models that include temporal information and those that do not, in both tasks.

4. Experiments

4.1. Dataset

The DAVIS 2016 is a benchmark dataset for evaluating video inpainting models. It consists of 120 videos with varied content including people, animals, objects, and natural settings [14]. The dataset has the videos at two resolutions: 480p and 1080p. It is important to note that this dataset has an already established division in which 90 videos are used for training/validating and 30 videos for testing [14]. As the original test folder does not have dynamic masks for each video, this data will not be taken into account in the study. With this in mind, the remaining 90 videos are taken from the DAVIS dataset and divided as follows: 54 for train, 18 for valid and 18 for test. It is important to mention that for each of the problems (removing static and Dynamic masks) the videos are the same, only the type of annotation changes depending on the task.

DAVIS dataset contains annotations for each frame of the videos. As such, each frame is densely annotated with object masks, indicating the foreground objects in each frame, as well as per-pixel semantic labels indicating the object category [14]. However, for the part of removing static masks, the original frames will be used as ground truth. The stationary random shapes were generated following the algorithm proposed by [3] that is shown in 1.

Now, for the removal of moving masks the original annotations of the dataset would be used. It should be noted that these annotations are segmentation maps with the object to be removed. Figure 2 shows an example of the annotations that will be used during the project.

Algorithm 1 Algorithm for creating static masks. maxPointNum and maxLength are user defined.

```
mask = zeros(Height, Width)
pointNum = random.uniform(maxPointNum)
startX = origX = random.uniform(Width)
startY = origY = random.uniform(Height)
angles = linspace(0, 2*pi, pointNum)
for i=0 to pointNum do
    length = random.uniform(maxLength)
    x = sin(angles[i]) * length
    y = cos(angles[i]) * length
    Connect (startX, startY) to (x, y) by cubic Bezier curves.
    startX = x
    startY = y
end for
Connect (startX, startY) to (origX, origY) by cubic Bezier curves
```

For memory-saving purposes and taking into account the available resources, the decision was made to only work with 480p-quality videos. In addition, it was established that all frames will have a size of 512×512 pixels, in order to save computational time when performing the experiments and tests. Based on the fact that the problem to be worked on is a generative problem, I decided to use metrics such as the Peak Signal-to-Noise Ratio (PSNR) (equation 1), Structural Similarity Index (SSIM) (equation 2) and video-based Fréchet Inception Distance (VFID). VFID is a perceptual metric that quantifies the quality of the generated videos. It is important to note that, in order to calculate the VFID metric, a I3D [2] pre-trained video recognition model was used. Moreover, the above is given considering that these metrics allow me to determine the pixel-wise fidelity of the generated frames and the perceptual quality of the video. It is important to clarify that, for this project, only metrics that estimate the quality of the generated images/videos will be used, no optical flow metrics were worked with.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \tag{1}$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

4.2. Validation experiments

The results of the fine-tuning process for both the LDM (Table 3) and ZITS (Table 2) models are shown below. Based on the non-perceptual metrics, the best combination is shown in bold. It should be noted that these tests were

carried out taking into account the set of train and validation for the task of static masks.

Table 2: Experimentation of parameters for the finnetunning process of the ZITS model with stationary masks.

Learning rate	Optimizer	PSNR (dB) (↑)	SSIM (↑)	VFID (↓)
1.00E-04	SGD	19.6	0.931	0.23
6.00E-04	SGD	20.1	0.941	0.195
1.00E-04	Adam	19.55	0.933	0.25
6.00E-04	**Adam**	**20.63**	**0.965**	**0.135**

Table 3: Experimentation of parameters for the finnetunning process of the LDM model with stationary masks.

Learning rate	Batch size	PSNR (dB) (↑)	SSIM (↑)	VFID (↓)
9.00E-05	5	20.2	0.955	0.15
9.00E-05	**10**	**20.7**	**0.96**	**0.144**
1.00E-04	5	19.7	0.95	0.16
1.00E-04	10	19.7	0.95	0.159

4.3. Evaluation Experiments

This section presents the results on the test sets for the four models chosen in the two tasks (Table 4). It should be noted that, for the models that were designed for image inpainting, the results of the best model found in the respective experiments are shown. In figure 5 the results for static masks are shown. In figure 6 the results for dynamic masks are shown.

Table 4: Final results of all models in the test sets of the tasks.

Model	Stationary Masks			Moving Masks
	PSNR (dB) (↑)	SSIM (↑)	VFID (↓)	VFID (↓)
LDM	20.7	0.96	0.144	0.395
ZITS	20.63	0.965	0.135	0.246
FF	28.61	0.8585	0.124	0.529
STTN	25.39	0.72	0.3299	0.675

5. Discusion

However, after the work done, it can be concluded that the training for the stationary mask removal task is not sufficient to compensate for the lack of temporal information. Furthermore, in terms of qualitative evaluation, models that do not include temporal information tend to leave the artifact created practically intact. While the modules that include temporal information achieve almost total removal of the artifact. Another detail to take into account is that the VFID metric used sometimes does not agree with the qualitative evaluation because some results differ a little, because when it is high the results do not look too bad. As a result, it can be thought that this type of perceptual metric is not the most appropriate since the features are not entirely similar to human perception.

When looking at the qualitative evaluation of the removal of dynamic masks, it is now clear that the methods that do not use temporal information present rather poor and inconsistent results. In addition, for this type of task, the fact that the objects have shadows should also be included. This detail is of vital importance, since shadows are seen but the object is not seen, so perceptually they are quite strange.

As for the limitations of the method today we have to enter and then it was only performed with 54 videos, which greatly conditions the possible results that would be obtained. This is due to the fact that this is a problem of natural images and it is very difficult for a model trained with so few data to give good results for such a large problem.

In conclusion, the inpainting video problem needs temporal information to produce qualitatively better results. Furthermore, the models should take into account details such as shadows in future work in order to obtain qualitatively better results.

6. Ethical Considerations

In terms of ethical considerations, this type of method can be used for illegal activities. This is due to the fact that it is very common to have videos as evidence in legal processes, and by being able to eliminate certain objects from them, the veracity of the videos is almost null. In addition, fake news can also benefit from this type of techniques, since it is possible to create scenes that never existed in order to incriminate a person or create controversy. Having all the above and remembering the wide range of applications that can have this type of techniques, we must be very careful in the popularization of the same, because today we live in a society that consumes many videos and today blindly trusts the veracity of the same; therefore, today these tools could lead to a chaos of disinformation.

References

[1] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE/CVF*

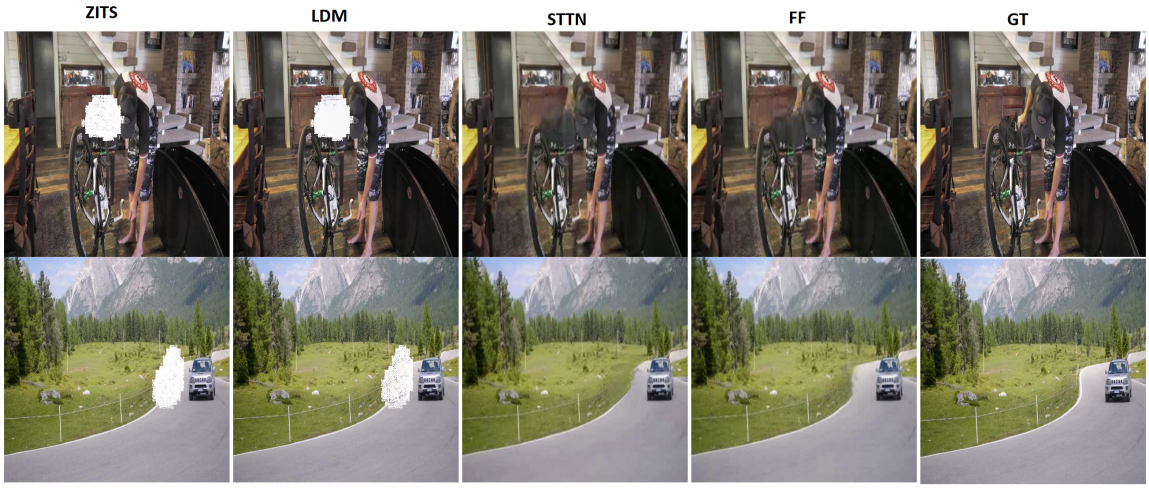

Figure 5: Results for all models in static masks task.

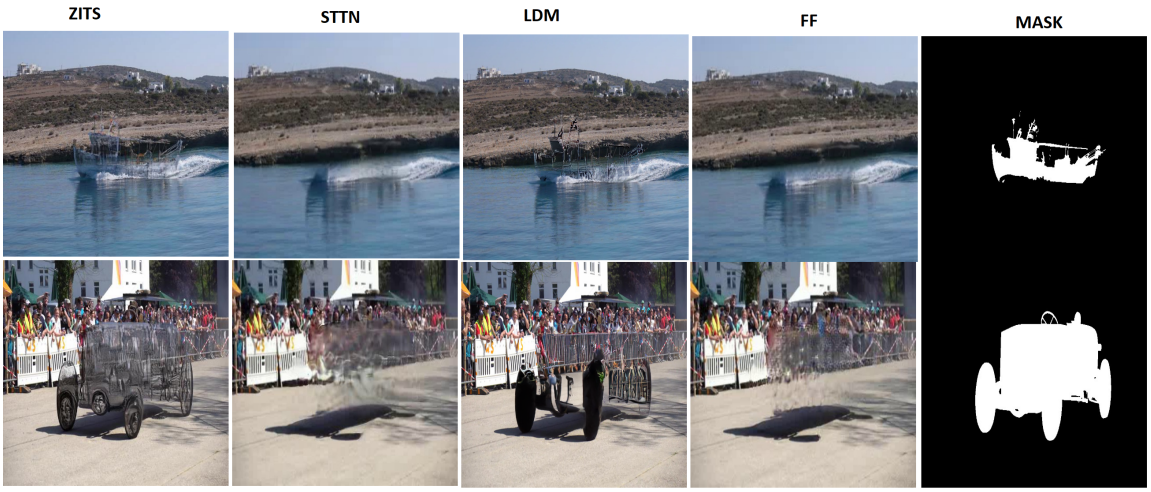

Figure 6: Results for all models in dynamic masks task.

International Conference on Computer Vision, pages 9066–9075, 2019.

[4] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Learnable gated temporal shift module for deep video inpainting. *arXiv preprint arXiv:1907.01131*, 2019.

[5] Ya-Liang Chang, Zhe Yu Liu, and Winston Hsu. Vornet: Spatio-temporally consistent video inpainting for object removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[6] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[7] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

[8] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019.

[9] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4413–4421, 2019.

[10] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17562–17571, 2022.

[11] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi,

Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *International Conference on Computer Vision (ICCV)*, 2021.

[12] Hao Ouyang, Tengfei Wang, and Qifeng Chen. Internal video inpainting by implicit long-range propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14579–14588, 2021.

[13] Pritika Patel, Ankit Prajapati, and Shailendra Mishra. Review of different inpainting algorithms. *International Journal of Computer Applications*, 59(18), 2012.

[14] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[16] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue 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.

[17] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2019.

[18] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 528–543. Springer, 2020.

7. Credits

All work was performed by the sole author.