Artificial Intelligence-Enabled Video Inpainting Using Spatio-Temporal Correlation

Shibo Li

Submitted in fulfilment of the requirements of the degree of
Doctor of Philosophy

James Watt School of Engineering
College of Science and Engineering
University of Glasgow

September 2024

I would like to dedicate this thesis to the ones I love.

Abstract

This Ph.D. thesis explores Artificial Intelligence (AI)-enabled video inpainting using spatio-temporal correlations. Video inpainting, an important technique in digital image processing and computer vision, aims to reconstruct the corrupted regions within a video and generate a visually pleasant result for viewers. This technology has broad applications across various fields, such as post-processing for television and films, historical protection, virtual reality, smart traffic systems, and medical healthcare. Due to the difficulties in manual editing and the rapid developments of AI techniques, there is a growing need to develop effective and robust AI-driven video inpainting approaches.

Although significant progress has been made with various traditional and deep learning-based methods in recent years, there are still a few challenges to develop effective and efficient video inpainting techniques. Key issues, such as visual and contextual inconsistencies, complex scenes with multiple layers, and high computational demand, often lead to noticeable artifacts and flickering in inpainted results. In order to address these problems, this research focuses on three perspectives: enhancing visual and contextual consistency, handling complex scenes with multiple layers, and improving computational efficiency. The main contributions of this Ph.D. thesis are as follows:

- Enhancing Visual and Contextual Consistency: A Short-Long-Term Propagation-based Video Inpainting (SLTPVI) approach was proposed for object removal tasks. This approach integrates two key modules: a Short-Term Propagation-based Inpainting (STPI) module and a Long-Term Propagation-based Inpainting (LTPI) module. The STPI module focuses on filling missing regions in a target frame by propagating local reference information from neighboring frames, employing three sub-components: source-region acquisition, illumination adaptation, and progressive fusion. The LTPI module builds on STPI by performing intra-group-of-pictures (GOP) and inter-GOP inpainting, ensuring consistency across frames by progressively propagating information throughout the video. The model was evaluated on the DAVIS and Landscapes datasets for the object removal task. Comparative analyses against the State-Of-The-Art (SOTA) methods demonstrated that the proposed approach achieves superior results in both temporal consistency and visual quality with better evaluation scores and reduced flickering in inpainting results. Though SLTPVI presented strong performance for object removal, it had difficulties dealing with video restorations task where both foreground and backgrounds are broken.

- Handling Complex Scenes with Multiple Layers: A Depth-Guided Deep Video Inpainting (DGDVI) was developed to handle challenging scenarios involving complex scenes where missing regions spans both foregrounds and backgrounds. The network consists of three sequential modules: a depth completion module, a content reconstruction module, and a content enhancement module. The depth completion module uses a spatio-temporal transformer to predict depth information for each video frame, creating a foundation for layer-aware inpainting. The content reconstruction module leverages depth-guided feature propagation to generate initial inpainting results, effectively addressing the spatial relationships between layers. Finally, the content enhancement module, based on a flow-guided deformable convolutional network, refines the texture details and temporal coherence of the inpainted results. The proposed method was evaluated on the DAVIS and YouTube-VOS datasets for both video restoration and object removal tasks. Experimental results showed that the model produces visually plausible and structurally accurate results for complex scenes with multiple depth layers. Comparisons with SOTA methods demonstrated that DGDVI achieved higher performance both qualitatively and quantitatively while handling more complex scenarios with multiple layers effectively. Though the proposed DGDVI showed robust performance on both video restoration task and object removal task, it still faced challenges in high-resolution video inpainting task.

- Improving Computational Efficiency: To address the computational challenges of high-resolution inpainting task, a Hierarchical Sparse Transformer for Video Inpainting (HSTVI) was proposed. The model processes videos in a coarse-to-fine manner through two stages: Low-resolution Global Context Reconstruction (LGCR) and High-resolution Sparse Content Enhancement (HSCE). The LGCR module, built on a sparse spatio-temporal transformer, sparsely samples frames in the temporal domain and utilizes global reference information to reconstruct the structure of missing regions at low resolutions. The HSCE module, based on a sparse flow-guided transformer, refines textures and enhances visual consistency by focusing on corrupted regions within high-resolution frames. This hierarchical approach significantly reduces computational and memory requirements, enabling efficient processing of videos at resolutions up to 2K. The proposed HSTVI is evaluated on the DAVIS and YouTube-VOS datasets across resolutions from 240p to 2K. Experimental results demonstrated that HSTVI outperforms SOTA methods in handling high-resolution videos, achieving superior results while demonstrating scalability and efficiency.

This thesis makes significant contributions to the field of AI-enabled video inpainting by addressing fundamental challenges in visual and contextual consistency, handling complex scenes with multiple layers, and computational efficiency. The proposed approaches—short-long-term propagation, depth-guided modeling, and hierarchical sparse Transformer—have advanced the state of the art in video inpainting, achieving superior results across diverse datasets and tasks, including object removal and video restoration. Beyond addressing the technical challenges, these contributions establish a foundation for scalable and robust video inpainting methods that can be adapted for real-world applications in media production, healthcare, scientific research, and beyond.

Looking ahead, video editing tasks, including inpainting, are poised to witness transformative developments over the next 5–10 years. As AI techniques continue to evolve, future research is expected to focus on enhancing user interactivity and customization in video editing. By integrating Natural Language Processing (NLP) and generative models, video inpainting systems could allow users to specify their desired edits through textual prompts, enabling intuitive and precise control over the content generation process. Additionally, advances in multimodal learning may lead to systems capable of simultaneously processing video, audio, and textual data, providing comprehensive tools for complex multimedia editing tasks. However, some challenges must be addressed to enable widespread adoption of video inpainting technologies in real-world applications, such as real-time processing and the reliability and ethical concerns of AI-generated content in sensitive domains. By addressing the challenges of real-time performance, ethical considerations, and user-centric design, future research can further enhance the impact of video inpainting and related video editing tasks, paving the way for transformative advancements in media, healthcare, education, and beyond in the coming decade.

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Shibo Li
September 2024

Acknowledgements

The journey of my Ph.D., spanning from 2020 to 2024, has been both an exciting and transformative adventure. Over these four years, I dedicated my research to two main areas: video inpainting for multimedia applications and wireless sensing for healthcare applications, utilizing AI technologies. These projects have not only improved my academic and engineering skills, but also shaped my perspective on the impact of technology on society. Beyond my technical research, I also spent significant time exploring philosophy and literature, reflecting on the future of humanity and AI, while contemplating deeper existential questions about why we exist and where we are headed. During this intellectual journey, the works of Hermann Hesse, Albert Camus, Aldous Huxley, and Nicolas Berdyaev provided me with inner guidance through moments of nihilistic reflections. However, it was the invaluable support and encouragement from many individuals that truly propelled me forward, both in my academic life and beyond. I express my deepest gratitude to all those who contributed to the success of my Ph.D.

First and foremost, I owe an immense debt of gratitude to my primary supervisor at the University of Glasgow, Prof. Jonathan Cooper. His unwavering support and guidance, both academically and personally, have been crucial throughout my Ph.D. journey. His mentorship provided not only expert academic direction but also life lessons that have deeply influenced both my personal and professional development.

I am also grateful to my co-supervisor at UESTC, Prof. Shuyuan Zhu, whose academic guidance over the years played an important role in shaping my research. His dedication to improving the quality of my work and helping me refine my papers has been invaluable.

I extend my sincere appreciation to the rest of my supervisory team at University of Glasgow, Prof. Muhammad Ali Imran and Prof. Qammer Abbasi. Their consistent support and mentorship were vital to my progress, both academically and personally. I am deeply grateful for their encouragement throughout my Ph.D. journey.

I am also profoundly grateful to my esteemed collaborators: Dr. Yao Ge, Mr. Minjian Shentu, Mr. Yuzhou Huang, Mr. Boyuan Zhang, and many others who contributed to my research through their insightful discussions, valuable advice, and active participation.

I express my heartfelt thanks to the University of Glasgow and UESTC for their financial support and their commitment to my academic journey. Their contributions were crucial in enabling my research and fostering my academic growth.

Last but certainly not least, I want to thank my parents, Mr. Qiuzhong Li and Mrs. Jinping Hu, for their unwavering support, both financially and emotionally, throughout this journey. I am also deeply grateful to my best friend, Mr. He Sun, whose constant encouragement and belief in me have been a source of motivation during challenging times. Their faith in me will continue to inspire and guide me in the long journey ahead.

Publications

Journal Papers:

1. **Shibo Li**, Yuzhou Huang, Shuyuan Zhu, Shuaicheng Liu, Bing Zeng, Muhammad Ali Imran, Qammer H. Abbasi, and Jonathan Cooper, "Short-long-term propagation-based video inpainting," *IEEE MultiMedia*, vol. 30, no. 4, pp. 91-104, 2023.

2. **Shibo Li**, Shuyuan Zhu, Yao Ge, Bing Zeng, Muhammad Ali Imran, Qammer H. Abbasi, and Jonathan Cooper, "Depth-guided deep video inpainting," *IEEE Transaction on Multimedia*, vol.26, pp. 860-5871, 2024.

Conference Papers:

1. **Shibo Li**, Yao Ge, Minjian Shentu, Shuyuan Zhu, Muhammad Ali Imran, Qammer H. Abbasi, and Jonathan Cooper, "Human activity recognition based on collaboration of vision and wifi signals," in *2021 International Conference on UK-China Emerging Technologies (UCET)*, pp. 204-208, 2021.

2. Yao Ge, **Shibo Li**, Minjian Shentu, Ahmad Taha, Shuyuan Zhu, Jonathan Cooper, Muhammad Ali Imran, and Qammer H. Abbasi, "A doppler-based human activity recognition system using WiFi signals," in *2021 IEEE Sensors*, pp. 1-4, 2021.

3. Yao Ge, **Shibo Li**, Shuyuan Zhu, Ahmad Taha, Jonathan Cooper, Muhammad Ali Imran, and Qammer H. Abbasi, "Respiration detection of sedentary person using ubiquitous WiFi signals," in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pp. 872-873, 2022.

4. Yu Liu, **Shibo Li**, Shuyuan Zhu, Siu-Kei Au Yeung, Xing Wen, and Bing Zeng, "Hierarchical coding for talking-head video," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3043-3047, 2022.

5. Yao Ge, Jingyan Wang, **Shibo Li**, Liyuan Qi, Shuyuan Zhu, Jonathan Cooper, Muhammad Ali Imran, and Qammer H. Abbasi, "WiFi sensing of Human Activity Recognition using Continuous AoA-ToF Maps," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp.1-6, 2023.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3D 3 Dimensional

AAST Axial Attention-based Style Transformer

AI Artificial Intelligence

AR Augmented Reality

AVC Advanced Video Coding

BPI Backward Propagation Inpainting

cGAN conditional Generative Adversarial Network

CLIP Contrastive Language-Image Pre-training

CNN Convolutional Neural Network

CPNet Copy-and-Paste Network

CPU Central Processing Unit

CvT Convolutional vision Transformer

CycleGAN Cycle-consistent Generative Adversarial Network

dB deciBels

DCN Deformable Convolutional Network

DETR DEtection TRansformer

DFVI Deep Flow-guided Video Inpainting

DGDVI Depth-Guided Deep Video Inpainting

DGSTTB Depth-Guided Spatio-Temporal Transformer Block

DLT Direct Linear Transformation

DNN Deep Neural Network

DSTT Decoupled Spatio-Temporal Transformer

$\mathbf{E}_{warp}$ flow warping error

E2FGVI End-to-End framework for Flow-Guided Video Inpainting

F3N Fusion Feed-Forward Network

FC Fully Connected

FFN Feed Forward Network

FGDW Flow-Guided Deformable Warping

FGF3N Flow-Guided Fusion Feed-Forward Network

FGHFTB Flow-Guided Hybrid Focal Transformer Blocks

FGSW-MSA Flow-Guided Soft Window Multi-head Self-Attention

FGT Flow-Guided Transformer

FGVC Flow-edge Guided Video Completion

FID Fréchet Inception Distance

FLANN Fast Library for Approximate Nearest Neighbors

FLOPs Floating Point Operations Per second

FPI Forward Propagation Inpainting

FuseFormer Fusing fine-grained information in transFormers

GAN Generative Adversarial Network

GDPR Generation Data Protection Regulation

GDSTB Global Dynamic Sparse Transformer Blocks

GOP Group Of Pictures

GPU Graphics Processing Unit

GST-MSA Global Spatio-Temporal Multi-head Self-Attention

GT Ground Truth

HEVC High Efficiency Video Coding

HF4N Hybrid Focal Fusion Feed Forward Network

HFKT Hybrid Focal Kernel-based Tokenization

HFSS Hybrid Focal Soft Split

HSCE High-resolution Sparse Content Enhancement

HSTVI Hierarchical Sparse Transformer for Video Inpainting

I3D Inflated 3D convnet

IIVI Implicit Internal Video Inpainting

ILSVRC ImageNet Large Scale Visual Recognition Challenge

LGCR Low-resolution Global Context Reconstruction

LGTSM Learnable Gated Temporal Shift Module

LIDAR LIght Detection And Ranging

LPIPS Learned Perceptual Image Patch Similarity

LTPI Long-Term Propagation-based Inpainting

MLP Multi-Layer Perception

MMSA Mutual Multi-head Self Attention

MRI Magnetic Resonance Imaging

MSA Multi-head Self Attention

MSE Mean Square Error

NLP Natural Language Processing

OPN Onion-Peel Network

PCEB Parallel Content Enhancement Block

PDE Partial Differential Equations

PSNR Peak Signal-to-Noise Ratio

RANSAC RANdom SAmple Consensus

SAVIT Semantic-Aware Video Inpainting Transformer

SC Soft Composition

SIFT Scale-Invariant Feature Transform

SLAM Simultaneous Localization And Mapping

SLIC Simple Linear Iterative Clustering

SLTPVI Short-Long-Term Propagation-based Video Inpainting

SOTA State-Of-The-Art

SS Soft Split

SSIM Structural Similarity Index Measure

STPI Short-Term Propagation-based Inpainting

STTB Spatio-Temporal Transformer Block

STTN Spatial-Temporal Transformation Network

StyleGAN Style-based Generative Adversarial Network

SURF Speeded Up Robust Features

SVD Singular Value Decomposition

TCCDV Temporally Coherent Completion of Dynamic Video

TSAM Temporal Shift-and-Aligned Module

VAE Variational Auto-Encoder

VFID Video Fréchet Inception Distance

VGG Visual Geometry Group

VINet deep Video Inpainting Network

ViT Vision Transformer

VMAF Video Multi-method Assessment Fusion

VQA Video Quality Assessment

VR Virtual Reality

VVC Versatile Video Coding

1

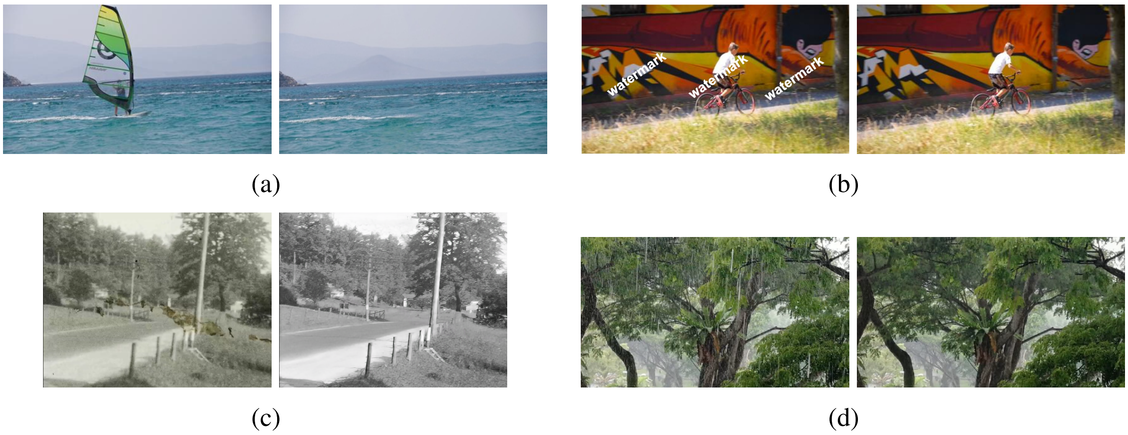
Introduction

1.1 Background and Significance

Video inpainting is an important technique in digital image processing and computer vision, which is used to reconstruct missing regions within a video. With the target to obtain a visually coherent and reasonable inpainting result, it is crucial to utilize the spatio-temporal correlations across the frames in the process of video inpainting. Traditionally, the edition and restoration of images and videos was a labor-intensive task performed by skilled artists and technicians, particularly in the art and film industries. Manual video editing always takes a large amount of time and energy, but often obtains visually inconsistent results. This fact creates a growing need to achieve automatic video inpainting for high-quality inpainting results. Meanwhile, with rapid developments in AI and Deep Neural Network (DNN), great progress has been made in digital multimedia and computer vision. Therefore, this research delves into AI-enabled video inpainting algorithms using spatio-temporal correlations, aiming to generate visually satisfying video inpainting results both efficiently and effectively.

In addition to the art and film industries, video inpainting is also widely applied across various fields, significantly enhancing developments within these domains. Based on real needs and application scenarios, video inpainting can be categorized into two primary tasks, i.e., object removal [3] and video restoration [8, 9].

Object removal, also known as video editing, aims to produce visually coherent results for the missing regions where unwanted objects are removed within the video. This technique is widely used in film and television post-production to remove elements, such as wires [12], fences [13], or moving targets [3], and creates special visual effects, such as making performers gradually invisible from scenes. In Augmented Reality (AR) and Virtual Reality (VR), this technique can be used to create clean 3 Dimensional (3D) reality without obstructions. Additionally, in traffic communication [14], object removal is employed to remove vehicles or pedestrians in image pre-processing to produce clean street maps, which prevents the interference of these objects. Furthermore, in privacy protection, it can also be used to remove targeted sensitive information from the videos while preserving the global contexts.



(a) (b) (c) (d)

Fig. 1.1 Various applications of video inpainting. From left to right, up to bottom: (a) unwanted object removal, (b) watermark restoration, (c) old film restoration [1], (d) video deraining [2].

Video restoration focuses on repairing damaged or degraded videos to recover original or enhanced content. In multimedia, video restoration techniques are widely utilized to improve the viewing experience for audiences, such as bringing old films back to life [1], fixing scratches on damaged videos [15], removing visual obstructions like snow or raindrops [2]. In history and culture preservation [16], it plays an important role in recovering the historical records, allowing future researchers and generations access to valuable visual documentations. Additionally, in medical imaging and healthcare applications [17, 18], video restoration can be used to enhance the quality of recorded medical videos by improving details and removing reflecting lights or obstructions, thereby increasing the accuracy of disease diagnosis. Moreover, in scientific research [19, 20], when data is unavailable for a specific period due to loss or other factors, video restoration can be used to predict missing data, as data from various sources can be transformed into images or videos for analysis.

Overall, the diverse applications of both object removal and video restoration proves the great potential of video inpainting techniques across various domains. To meet the increasing need for high-quality video inpainting results, it motivates us to explore both effective and efficient AI-driven video inpainting schemes.

1.2 Key Challenges

Similar to other low-level tasks in image and video processing, such as video denoising and super-resolution, video inpainting also involves in recovering important visual information, including edges, textures, and shapes in videos. An important fact is that these low-level vision tasks are essentially ill-posed problems [21, 22], as their solutions are not unique. The existence of multiple plausible outputs for an input introduces inherent uncertainty, making it difficult to achieve accurate and reliable results. However, video inpainting is more challenging than other tasks, despite sharing the same mathematical nature. The primary difficulty lies in the need to reconstruct highly degraded regions where the pixels are entirely missing. This highly differs from video denoising, where pixels are present but contaminated by noise, and video super-resolution, where intact structures exist but at a lower resolution. Consequently, despite numerous approaches proposed in recent years, several key technical challenges still remain in developing an effective and efficient video inpainting scheme.

Challenge for Visual and Contextual Inconsistencies

One of the most significant challenges in video inpainting is to ensure both visual consistency and contextual coherence for the results. Visual consistency refers to the spatio-temporal smoothness of the inpainted video, which is important for providing a satisfying viewing experience. For an inpainting result with good visual consistency, the reconstructed content in the missing regions must be reasonable and can be seamlessly and naturally blended with the surrounding valid areas. Additionally, the inpainted content within each frame should remain consistent across consecutive frames to avoid noticeable flickering during playback. Contextual coherence, on the other hand, involves maintaining the overall narrative and structure of the entire video. This requires the inpainting technique to incorporate reference information not just from short-range, neighbouring frames but also from long-range sources, particularly in cases involving large missing regions. Achieving these two aspects is complex, as it demands that the inpainting algorithm effectively integrate spatial and temporal cues, while leveraging both short-range and long-range information to effectively reconstruct the missing content.

Challenge for Complex Scenes with Multiple Layers

Complex scenes often involve various objects across distinct foreground and background layers, with different motion patterns for each layer. As the information in the target regions is totally lost, it becomes extremely difficult to determine which layer different missing pixels belong to, especially when dealing with large missing regions spanning multiple layers. If inpainting algorithms fail to correctly build the relationships between layers, this can result in blurred object structures and unnatural interactions between the layers. When dealing with such cases, we can notice obvious artifacts in the results from most previous video inpainting approaches. Therefore, handling complex scenes with multiple layers poses a significant challenge in designing an effective AI-enabled video inpainting algorithm.

Challenge for High Computational Demand

Compared to image inpainting, video inpainting presents greater computational challenges due to the need to process a sequence of frames rather than a single image. To generate consistent inpainting results, the algorithm must gather abundant reference information from both spatial and temporal dimensions. This requires the analysis of a long sequence of frames to capture the relevant cues, leading to high computational and memory demands for both Central Processing Unit (CPU) and Graphics Processing Unit (GPU). If the target video is too long, many inpainting approaches fail in memory management and collapse due to the computation overload. When dealing with high-resolution videos, this issue is further exaggerated as the amount of data increases exponentially. Therefore, the computational issue severely hinders real-time video inpainting applications, which is an important requirement for computational management and algorithm optimization in the development of efficient video inpainting approaches.

1.3 Motivations and Objectives

With the aim of developing AI-enabled video inpainting approaches that can produce high-quality, reliable, and visually consistent results, addressing the key challenges in video inpainting, including enhancing visual and contextual consistency, handling complex scenes with multiple layers, and managing high computational demand, serves as the primary motivations in this research. The following paragraphs introduce the detailed motivations and objectives of this thesis.

Enhancing Visual and Contextual Coherency

In order to enhance the visual and contextual consistency in video inpainting, the key lies in the effective utilization of spatio-temporal correlations within the video to generate reliable content. Given that there is abundant repetitive information across frames, we must prioritize the use of available information within the video, especially cues from temporal dimensions. If we ignore the importance of temporal correlations, solely using spatial contexts within single frames in video inpainting instead, it will lead to obvious flickering due to the inconsistent content across consecutive frames. To make full use of spatio-temporal correlations, it requires to facilitate both short-term and long-term propagation of reference information to achieve results with best visual consistency. In short-term propagation, the cues provided by neighboring frames are more reliable and consistent references but may not be sufficient for the entire missing area, especially for videos with large corrupted regions. As for long-term propagation, the reference information provided by distant frames contains sufficient contexts but may be inconsistent with the target frame. Our approach aims to effectively gather the cues by using both short-term and long-term propagation, ensuring a coherent and seamless inpainting result.

Handling Complex Scenes with Multiple Layers

To effectively handle complex scenes with multiple layers, it is important to build correct relationships between different layers within the video. First of all, depth or semantic information can be utilized to model different layers within the video in the design of video inpainting framework. When the layers for valid areas has been determined, one solution to produce plausible inpainting results is that we can separate different layers and complete the inpainting for each layer, respectively. In this way, the reference information can be propagated to the corresponding layers independently and then combine together to acquire a visually satisfying result. Another approach is to analyze the layer information in missing regions, i.e., predicting the depth or semantic information for missing pixels. During the subsequent inpainting process, corrupted regions can be reconstructed under the guidance of the predicted layer information, so that clear structures can be reconstructed for target regions spanning multiple layers.

Improving Computational Efficiency

Addressing the high computational demand of video inpainting algorithms is essential to real applications and improve scalability, especially for long-sequence or high-resolution videos. It can be achieved from two perspectives, including controlling algorithm complexity and data flow. Firstly, to reduce the algorithm complexity, we aim to build the video inpainting into a hierarchical framework, including low-resolution reconstruction and high-resolution enhancement. With this architecture, the algorithm can deal with long-range cues and capture global contextual information at low resolutions, and afterwards improve the texture and details at high resolutions, thereby effectively cutting down the computation cost. Secondly, how to control the data flow in computation is also important for efficient video inpainting. The large amount of redundant information within the video brings a large data flow in signal processing. Meanwhile, we observe that mask regions often occupy only a small proportion of pixels in the videos. This indicates that most valid regions could waste computation resources. To improve computational efficiency, we can compress and prune the data flow, which means extracting the most useful information and abandoning redundant information. For long-sequence videos, key frames can be sampled from the temporal dimension to control the length of frames for inpainting. For high-resolution videos, important key windows that contain target regions can be extracted from the spatial dimension to perform sophisticated computation and refinement, preventing the waste of resources in valid regions. Our proposed approach aims to enhance the computational efficiency of video inpainting algorithms by combining these strategies.

1.4 Overall Contributions

The overall research goal of my research is to design effective and efficient AI-enabled video inpainting approaches using spatio-temporal correlations. In this Ph.D. thesis, the main contributions can be summarized as follows:

- To address the challenge of visual and contextual consistency, a short-long-term propagation-based video inpainting approach is proposed for object removal. Our method combines STPI and LTPI modules. The STPI module infills an frame using local reference information from adjacent frames, while the LTPI module uses multiple STPI modules to inpaint the entire video, ensuring high temporal consistency and low complexity. With these modules, the correlated spatio-temporal information can be propagated throughout the video, providing reliable source information from both local and global ranges for inpainting. The experimental results demonstrate that our proposed method provides results with better visual and contextual consistency compared with the state-of-the-art.

- To deal with the challenge of complex scenes with multiple layers, a depth-guided deep video inpainting network is designed. Our approach divides the inpainting process into three stages: depth completion, content reconstruction, and content enhancement. We develop a depth completion module based on a spatio-temporal transformer to obtain completed depth information for each video frame. A content reconstruction module generates the initial inpainted video, using depth-guided feature propagation to fill in missing regions. Finally, a content enhancement module improves temporal coherence and texture quality. All proposed modules are jointly optimized to guarantee high inpainting efficiency. Experimental results show that our method provides superior inpainting results, both qualitatively and quantitatively, compared to previous state-of-the-art techniques.

- To tackle the challenge of high computational demand, a hierarchical video inpainting approach for high-resolution videos is developed. This approach consists of two modules: a low-resolution global context reconstruction module and a high-resolution content enhancement module. The global context reconstruction module, built upon a sparse spatio-temporal Transformer, generates an initial low-resolution inpainting result by leveraging global context information. The subsequent high-resolution content enhancement module, based on a sparse flow-guided Transformer, refines the initial output to produce high-resolution results with finer details and improved temporal consistency across consecutive frames. By dividing the inpainting process into these two stages, our framework effectively manages computational and memory resources, significantly improving efficiency. Experimental results demonstrate that our method outperforms previous state-of-the-art approaches on high-resolution videos, both qualitatively and quantitatively.

1.5 Organization of Thesis

In this Ph.D. thesis, it consists of six chapters and is organized as follows. Chapter 1 provides an introduction to the overall thesis. Chapter 2 conducts a literature review for image and video inpainting progress in recent years, containing traditional approaches and deep learning-based ones, and introduces the basic knowledge of related techniques. Chapter 3 introduces the short-long-term propagation-based video inpainting for object removal. Chapter 4 designs the depth-guided deep video inpainting approach for complex scenes with multiple layers. Chapter 5 presents the hierarchical sparse Transformer for efficient high-resolution video inpainting. Finally, Chapter 6 gives the conclusion of the whole thesis and discusses the technical challenges and limitations based on recent researches in the thesis. Meanwhile, the chapter provides perspectives and plans for future work.

2

Literature Review

2.1 Introduction

In this chapter, we provide a comprehensive literature review on AI-enabled video inpainting using spatio-temporal correlations. We mainly focus on two parts, i.e. the related work of image and video inpainting, as well as the related conceptions and techniques in these areas.

Firstly, we conduct a review of image and video inpainting methods that have shown great progress in recent years. Image inpainting focuses on reconstructing missing regions within a single image, while video inpainting techniques aim to reconstruct missing regions within a sequence of frames. The development of image inpainting facilitates video inpainting as well, providing great insights and ideas for video inpainting. Therefore, we will first present the key developments in image inpainting and then explore the evolution of video inpainting approaches. By discussing the previous methods and the existing challenges in image and video inpainting, we can build a clear comprehension of this field to further design more effective and efficient video inpainting approaches.

Secondly, we introduce some key conceptions and technologies related to video inpainting in the chapter. These include two typical frame alignment techniques, i.e. homography-based warping and optical flow estimation, which are important for information propagation in video inpainting. We also give a brief introduction to depth estimation, which is essential for handling complex scenes with multiple layers. Moreover, we introduce several important deep learning architectures in image and video inpainting, such as Convolutional Neural Network (CNN), Vision Transformer, and Generative Adversarial Network (GAN). Finally, we present evaluation metrics used to quantitatively assess the performance of video inpainting algorithms. By exploring these aspects, we can establish a system of basic knowledge for further exploration of AI-driven video inpainting approaches.

2.2 Image and Video Inpainting

2.2.1 Image Inpainting

Image inpainting aims to generate reliable and visually pleasant content for corrupted regions in an image. This field can be broadly categorized into traditional methods and deep learning-based methods. Traditional methods, such as **diffusion-based approaches** [23, 24] and **exemplar-based approaches** [25, 26], rely on mathematical models or statistical principles to propagate surrounding textures into the missing regions or find and copy similar patches from the undamaged areas. However, these methods often struggle with complex structures and large missing areas. Deep learning-based methods have significantly advanced the field by leveraging the representational power of neural networks. Among them, **CNN-based methods** [27–40] have been widely used for their ability to capture local patterns and hierarchical features, making them highly effective for generating detailed textures and structures. More recently, **Transformer-based methods** [41–43] have emerged, utilizing self-attention mechanisms to model long-range dependencies, which is particularly beneficial for reconstructing content in large or contextually complex missing regions. Together, these advancements have propelled image inpainting toward producing more realistic and contextually consistent results.

Traditional Image Inpainting

Before deep learning methods were widely applied in this filed, image inpainting was traditionally implemented in a non-learning manner, which can be generally divided into diffusion-based methods [23, 24] and exemplar-based methods [25, 26]. The idea of diffusion-based methods comes from analogy with physical phenomena like heat propagation in thermodynamic systems. Bertalmio *et al.* [23] proposed the first diffusion-based method by formulating this process with Partial Differential Equations (PDE), which smoothly propagates local information from the exterior to the interior of the hole to fill in missing regions. Shen *et al.* [24] subsequently introduced the total variation to recover broken edges, ensuring the robustness of the method with respect to noise. Diffusion-based methods are suitable for inpainting with scratches and curves, but they are easy to generate over-smooth results without fine textures and details when dealing with large missing regions due to PDE-based smoothness constraints. A different complementary scheme involves exemplar-based methods, which is also known as patch-based methods, which synthesizes contents for the target regions by sampling and copying patches from the known regions within the image. For example, Criminisi *et al.* [25] proposed an isophote-oriented patch ordering approach for image inpainting and the sampling ordering of patches is based on structure continuity.

Jin *et al.* [26] subsequently introduced facet deduced directional derivatives in searching candidate patches so that the continuity of boundaries of the inpainted region could be well guaranteed. Exemplar-based methods can produce good results for broken videos when source patches are easy to acquire from valid areas. But if there is no existence of similar patches in known regions, it usually fails to generate reasonable results. Moreover, it suffers from a high computation cost if the algorithm searches similar patches in a greedy manner.

CNN-based Image Inpainting

Deep learning-based image inpainting has demonstrated impressive performance in recent years, especially CNN architectures. For instance, Pathak *et al.* [44] introduced GAN [45] to image inpainting, achieving much more impressive results than traditional methods. Other advanced modules or learning strategies were also proposed to produce high-quality inpainted images, including contextual attention [27–30], partial convolution [31], gated convolution [32] and Fourier convolution [33]. In these methods, Yu *et al.* [27] introduced a contextual attention module to implement the coarse-to-fine generative image inpainting. Liu *et al.* [31] designed an image inpainting network using the partial convolutions to effectively extract features from degraded regions, producing better inpainting results. Yu *et al.* [32] constructed a gated convolution-based inpainting network to selectively integrate valid information collected from the surrounding regions, thereby composing seamless content for the missing region. Recently, Suvorov *et al.* [33] built an image inpainting network using the Fourier convolution to obtain wide receptive field so that the model can implement the large mask inpainting.

Due to the absence of structures and textures in missing regions, some methods [34–40] introduced intermediate clues as guidance to generate more reliable results. For example, Nazeri *et al.* [34] constructed an Edgeconnect network to predict the edges for missing contents. The predicted edges are used to guide the inpainting process with a completion network. Xiong *et al.* [35] developed a foreground contour completion network to predict foreground contour for the missing regions and built a CNN-based image completion network to generate contents with the guidance of the predicted foreground contour. Ren *et al.* [36] proposed a structure reconstruction CNN model to complete the missing structure information of image and also designed a texture generator to yield image details according to the reconstructed structures. Wu *et al.* [37] constructed a two-staged generative model for image inpainting, which firstly accurately predicts the structural information of the missing region based on local binary pattern learning and subsequently builds a structure-guided image inpainting network using spatial attention. Song *et al.* [38] developed a segmentation prediction model to obtain segmentation information for missing regions and then built a segmentation-guided image inpainting network to generate semantically consistent content. Moreover, instead of one-way semantic guidance, Zhang *et al.* [39] established a semantic-guided image inpainting framework in which the semantic segmentation guides image inpainting and also receives feedback from image inpainting to generate more reliable inpainting results. Additionally, Sun *et al.* [40] proposed a deep network which learns to decompose a complex mask area into several basic mask types and inpaints the damaged image in a patch-wise manner to enhance the robustness of the method. These CNN-based methods have stronger abilities than traditional methods in the extraction of features and the generation of non-existing contents in the image. But there are still some limitations in CNN-based methods. Due to the limited perception field of CNN to capture the global context, many methods cannot guarantee structural cohesion or a harmonious texture between the inpainted region and the image context, especially for high-resolution cases.

Transformer-based Image Inpainting

Besides CNN-based methods, Transformer-based image inpainting approaches [41–43] also achieved remarkable performance due to their ability to capture long-range dependencies within an image. For instance, Yu *et al.* [41] proposed a bidirectional autoregressive Transformer for image inpainting. Li *et al.* [42] constructed a mask-aware Transformer to repair the image with large missing areas. Dong *et al.* [43] designed a Transformer model to restore the low-resolution structure for the broken image and combined it with a Fourier CNN model to generate fine textures in the missing regions, guided by the up-sampled structure. Transformer-based methods can produce more coherent and reasonable results than CNN-based methods, but some challenges still remain when dealing with large missing regions and ensuring controlled generation of content to meet specific user requirements.

2.2.2 Video Inpainting

Video inpainting, an extension of image inpainting, focuses on reconstructing missing or corrupted regions in video frames while maintaining temporal consistency across consecutive frames. This field can be broadly categorized into patch-based methods, flow-guided methods, and deep learning-based approaches. Traditional **patch-based methods** [46, 47, 3, 48–50] rely on searching and copying spatio-temporal patches from valid regions to fill the missing areas. While effective for small and simple missing regions, these methods often fail to handle large missing areas or complex motion dynamics, leading to visible artifacts. **Flow-guided methods** [6, 51–54] utilize optical flow to align and propagate information from neighboring frames to inpaint missing regions. These methods effectively maintain temporal consistency but can struggle with inaccuracies in flow estimation, especially in cases of large missing regions or complex motion patterns. Deep learning-based methods have further revolutionized video inpainting by leveraging neural networks to learn spatio-temporal correlations. **CNN-based methods** [4, 5, 7, 55–57] use convolutional operations to extract spatial and temporal features, providing efficient solutions for local content restoration but often struggling with long-range dependencies. **Transformer-based methods** [58–61, 8, 9, 62, 63], on the other hand, leverage self-attention mechanisms to capture global context and long-range dependencies, making them particularly effective for large or complex missing regions. These advancements, combined with flow guidance, have significantly improved the quality and robustness of video inpainting, enabling its application in diverse scenarios.

Patch-based Video Inpainting

Similar with developments of image inpainting, traditional patch-based video inpainting approaches were developed before the wide utilization of DNN. The primary idea of patch-based video inpainting approaches is to fill the missing regions with the available patches collected spatially or temporally from the known regions, which can be implemented in either a greedy or a global fashion. The greedy-based solutions [46] were used to fill the regions pixel by pixel, but often produced inconsistent results. To solve this problem, a global objective function [47] was introduced to optimize patch search and accordingly constructed the global solution. Huang *et al.* [3] proposed a novel framwork called Temporally Coherent Completion of Dynamic Video (TCCDV), which effectively optimizes the the performance and efficiency of [47] by adding an optical flow term to enforce temporal consistency. To speed up patch collection and strengthen the coherence, a 3D Patch-Match was adopted [48] and the resulting method was optimized [49] by introducing an additional optical flow term to ensure temporal coherence. Additionally, Aldahdooh *et al.* [50] used motion maps and adaptive searching windows to construct an error-concealment scheme for video inpainting.

Flow-guided Video Inpainting

Due to the limited temporal correlations across frames, patch-based methods often result in inconsistent visual artifacts in the inpainted videos. To solve this problem, some approaches [6, 51–54] adopted optical flow as the guidance to propagate cues from the neighboring frames to target frame for content construction. For example, Xu *et al.* [51] proposed a Deep Flow-guided Video Inpainting (DFVI) that first builds a flow completion module to estimate the flow of the missing region, facilitating the propagation and composition of content in the pixel domain. Based on DFVI, Zou *et al.* [52] predict the flow and proposed a novel Temporal Shift-and-Aligned Module (TSAM) to propagate the reference cues for content construction in feature domain. To obtain the reliable flow for the generation of temporally coherent results, Gao *et al.* [6] designed Flow-edge Guided Video Completion (FGVC), which employs Edgeconnect [34] to predict the edges of missing contents and then uses the edge information to guide the completion of flow. Then, based on the motion information of object, Zhang *et al.* [53] introduced inertia prior for optical flow estimation, thereby generating better flows to produce more reliable inpainting results. Recently, Kang *et al.* [54] proposed an error compensation method to improve the accuracy of flow completion for the implementation of flow-guided inpainting with high-efficiency.

CNN-based Video Inpainting

In recent years, the application of deep learning, especially CNN-based methods, has also demonstrated good performance in video inpainting. For example, Lee *et al.* [4] constructed a Copy-and-Paste Network (CPNet) to aggregate the cues collected from the reference frames to the inpaint target frame. Oh *et al.* [5] proposed a Onion-Peel Network (OPN) that progressively fills the hole by collecting the contents in the reference images. Furthermore, Ouyang *et al.* [7] applied an Implicit Internal Video Inpainting (IIVI), processing challenging or complex scenarios which contain ambiguous backgrounds or long-term occlusion. To obtain both spatial and temporal cues for inpainting, some methods [64, 55–57] employed 3D convolution to construct the deep video inpainting network. Specifically, Kim *et al.* [55] designed a deep Video Inpainting Network (VINet) for inpainting by using both 3D and 2D convolutions to collect spatial and temporal information. Chang *et al.* [56] built a temporal PatchGAN model based on the proposed 3D gated convolution for free-form video inpainting. Additional, Chang *et al.* [57] also proposed a Learnable Gated Temporal Shift Module (LGTSM) for video inpainting models that could effectively tackle arbitrary video masks without additional parameters from 3D convolutions.

Transformer-based Image Inpainting

In addition to the CNN-based method, Transformer models were also widely applied for video inpainting in recent years and presented impressive performances. Based on the Vision Transformer (ViT) [65], Zeng *et al.* [58] first developed a Spatial-Temporal Transformation Network (STTN) for video inpainting. To improve STTN, Liu *et al.* [59] built FuseFormer model to generate fine-grained contents by using overlapped patch embeddings. Furthermore, Liu *et al.* [60] constructed a Decoupled Spatio-Temporal Transformer (DSTT) scheme, implementing independent spatial propagation and temporal propagation with two different attention blocks to compose the contents. Masum *et al.* [61] constructed an end-to-end network based on Axial Attention-based Style Transformer (AAST) to achieve consistent video inpainting. To introduce flow-based guidance into the Transformer-based model, Li *et al.* [8] designed an End-to-End framework for Flow-Guided Video Inpainting (E2FGVI),

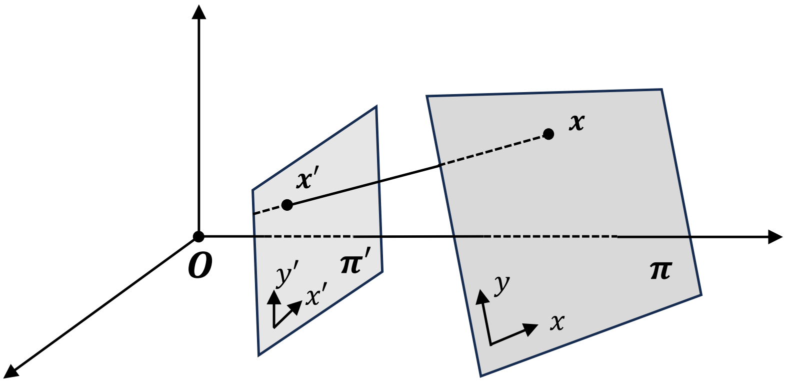


Fig. 2.1 Projection of points from one plane to points on another plane.

which adopts flow-guided convolution for short-term propagation across neighboring frames and uses the Transformer model to facilitate long-term spatio-temporal propagation. Additionally, Zhang *et al.* [9] designed a Flow-Guided Transformer (FGT) model to fuse cues to produce high-quality results. Lee *et al.* [62] introduced a Semantic-Aware Video Inpainting Transformer (SAVIT) model that can exploit semantic information within the input to effectively improve reconstruction quality and restore clear object boundaries. Furthermore, to address the constraints in flow-guided propagation and long-range exploration of reference information from distant frames, Zhou *et al.* [63] designed an improved Transformer called ProPainter by combining a dual-domain propagation and a mask-aware sparse Transformer.

2.3 Related Conceptions and Technologies

2.3.1 Homography-based Warping

Homography-based warping is a crucial technique in computer vision that involves transforming the perspective of an image or video frame to align with another view. This technique is based on estimating a homography matrix, which builds the geometric relationship between two planes for linear transformation. Homography-based warping is widely used in various applications, such as image stitching, panorama creation, AR, and 3D reconstruction.

A homography matrix [66] is represented by a 3×3 matrix that relates the coordinates of corresponding points between two planes, as shown in Fig. 2.1. Let (x, y) be the coordinates in the source image and (x', y') be the coordinates in the target image. The homography relationship can be expressed as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.1}$$

where H is the homography matrix defined as

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \tag{2.2}$$

The matrix H has 8 degrees of freedom, which means it is determined with 8 independent parameters. By controlling these parameters, the homography matrix H can complete the transformations in image warping, such as shifting, rotation, scaling, deformation, and perspective transformation.

The process to estimate the homography matrix between two images involves several key steps. First, feature extraction is performed using techniques such as Scale-Invariant Feature Transform (SIFT) [67] or Speeded Up Robust Features (SURF) [68] to detect distinctive points in both images. Next, feature matching is performed to find correspondences between the features in the two images, typically using methods like the Fast Library for Approximate Nearest Neighbors (FLANN) [69] matcher. After obtaining the initial set of matched features, outliers are filtered out using robust techniques such as RANdom SAmple Consensus (RANSAC) [70], which iteratively estimates the homography and retains only the inliers that support the best model. Once a set of reliable correspondences is established, the homography matrix is computed by solving a system of linear equations derived from these correspondences, often using methods like Direct Linear Transformation (DLT) [66] combined with Singular Value Decomposition (SVD) [71] to find the optimal solution. Finally, the initial homography estimate can be further refined using non-linear optimization techniques such as the Levenberg-Marquardt algorithm [72] to minimize the projection error, ensuring more accurate alignment between the images. Once the homography matrix H is obtained, it can be used to warp the source image to align with the target image, enabling the alignment and mapping of two images.

Though the process of single homography-based warping can be effective for many scenes, it often appears misalignment or significant distortions in scenarios that contain multiple planes with obvious parallax. Mesh homography-based warping [73] can effectively mitigate this problem by dividing the image into many smaller grids or meshes, with a separate homography estimated for each mesh. The algorithm constructs an optimization function to maintain the global shape of the mesh, so that it can prevent the distortion of

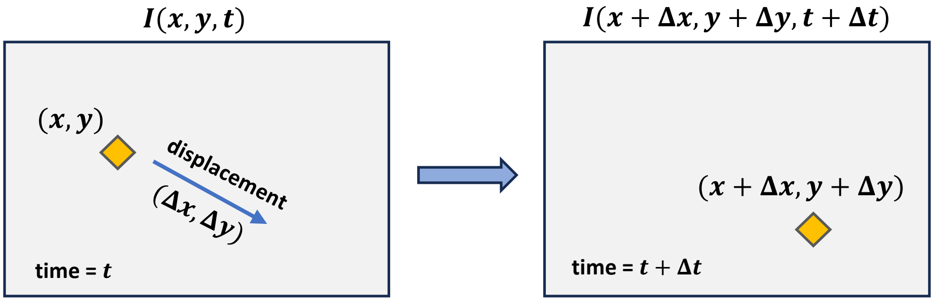


Fig. 2.2 Displacements of points being tracked across frames.

the content while warping the image. Meanwhile, the model estimates local homography matrices for each grid under the global constraint of the mesh, allowing it to independently warp its local region. This model provides a more flexible and accurate approach for image alignment that is tolerant to parallax in many complex scenarios.

Discussion: Homography-based warping significantly contributes to video inpainting by enabling the accurate alignment and transformation of frames, which is essential for maintaining temporal coherence and spatial consistency. By estimating the homography matrix, we can map the corresponding points between consecutive frames, allowing the seamless propagation of information across the video sequence. This capability is particularly crucial when dealing with dynamic scenes and camera movements, as it helps to accurately reconstruct missing or corrupted regions in each frame based on their aligned counterparts. Moreover, homography-based warping facilitates the handling of multiple planes within a scene by dividing the image into smaller meshes and applying separate homographies, thereby addressing complex transformations and perspective distortions. This ensures that the inpainted content appears natural and integrated, preserving the overall visual continuity and enhancing the quality of the inpainted video.

2.3.2 Optical Flow Estimation

Optical flow estimation is a fundamental technique in computer vision used to compute the apparent movement of objects, surfaces, and edges from one frame to the next in a video sequence. The optical flow field describes the displacement of each pixel between consecutive frames, providing critical information about the relative movement between the scene and the camera.

The fundamental mathematical equation for optical flow estimation is based on the brightness constancy assumption, which states that the intensity of a pixel remains constant as it moves between frames. Let us denote the brightness intensity at position (x, y) in the image I at time t as $I(x,y,t)$. Following the brightness constancy assumption, the problem of optical flow can be expressed as

$$I(x,y,z) = I(x+\Delta x, y+\Delta y, t+\Delta t). \tag{2.3}$$

where Δx and Δy represent the movements in the horizontal and vertical directions and Δt represents a very short period of time, as shown in Fig. 2.2. Using a first-order Taylor series expansion, we approximate $I(x+\Delta x, y+\Delta y, t+\Delta t)$ as

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t. \tag{2.4}$$

The equation can be simplified to be

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0. \tag{2.5}$$

In compact, it can be expressed as

$$I_x V_x + I_y V_y + I_t = 0, \tag{2.6}$$

where I_x, I_y, and I_t is used to express the derivatives of the image at (x,y,t) in the corresponding directions, and V_x and V_y represent the velocity in the horizontal and vertical directions. This equation is known as the optical flow constraint equation. To find a unique solution, optical flow algorithms introduce additional constraints and conditions to estimate the flow.

Optical flow estimation techniques can be broadly categorized into traditional approaches and deep-based ones. Traditional optical flow estimation methods, such as the Horn-Schunck [74] and Lucas-Kanade [75] algorithms, solve partial differential equations to minimize the difference between pixel intensities in consecutive frames. These methods typically rely on assumptions of smoothness and brightness constancy to compute the flow field. Although computationally efficient, classical methods often struggle with the robustness such as large displacements, occlusions, and complex motion patterns, limiting their accuracy in challenging scenarios. Recent advances in deep learning have led to the development of more robust and accurate optical flow estimation techniques. Models such as FlowNet [76, 77] and RAFT [78] utilize CNN to learn motion patterns from large datasets, significantly improving the quality of the estimated flow fields. These methods can handle complex motions, large displacements, and occlusions more effectively than classical approaches. Additionally, deep learning models can generalize better to different types of scenes and motion dynamics, making them more versatile for various applications.

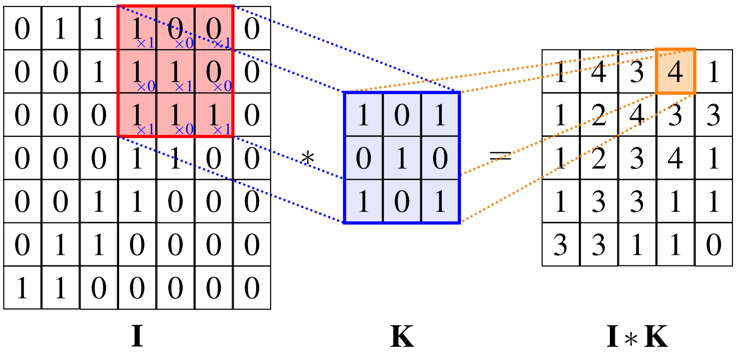
Discussion: Optical flow guidance [6, 51–54] is crucial for maintaining temporal coherence in video inpainting, where ensuring temporal consistency between frames is essential for reconstructing visually plausible content. By synthesizing realistic motions for corrupted regions and accurately predicting the locations of missing or occluded pixels, optical flow facilitates the propagation of reference information through the temporal dimension. Moreover, it enhances the temporal correlations and texture quality of the inpainted result. However, high-quality optical flow is required, as errors in optical flow estimation can propagate through the inpainting process, leading to artifacts and inconsistencies.

2.3.3 Depth Estimation

Depth estimation is a crucial task in computer vision that involves predicting the distance of objects from the camera in a scene. This task has significant applications in various fields, including Simultaneous Localization And Mapping (SLAM), AR, autonomous driving, robotics, and computer vision.

Traditional depth estimation techniques primarily rely on stereo vision [79–81], which utilizes two or more cameras to capture different viewpoints of the same scene, calculating depth by triangulating the disparity between corresponding points in the images. Recent advances in deep learning have significantly improved the accuracy and robustness of monocular depth estimation techniques, which focus on estimating depth information from monocular images with its inherent relationships. Eigen *et al.* [82] pioneered the use of supervised learning for monocular depth estimation, demonstrating that a CNN trained on large datasets could outperform traditional methods. In addition to these supervised approaches [82, 83], which require large amounts of labeled data, semi-supervised and unsupervised depth estimation techniques have gained significant attention in recent years, as they offer the potential to overcome these limitations. Semi-supervised learning methods [84–86] incorporate additional information, such as synthetic data, surface normals, and LIght Detection And Ranging (LIDAR), to reduce the dependence of training on Ground Truth (GT) depth maps, enhancing scale consistency and improving the accuracy of depth maps. Unsupervised depth estimation techniques [87–89], on the other hand, do not rely on any labeled GT data. Instead, they exploit the inherent structure and relationships within the data to learn depth estimation.

Discussions: Depth estimation is important in video inpainting, particularly in handling complex scenes with multiple layers. The depth information helps inpainting algorithms understand the spatial relationships between objects and the scene structure, ensuring that the reconstructed content maintains the correct depth ordering. The estimated depth information also helps to propagate contextual cues across frames to correct depth layers, enhancing contextual coherence for the inpainted result. Though current depth estimation models provide solutions to predict the depth information, there are difficulties to obtain depth for



$$\begin{pmatrix} 0&1&1&1&0&0&0\\ 0&0&1&1&1&0&0\\ 0&0&0&1&1&1&0\\ 0&0&0&1&1&0&0\\ 0&0&1&1&0&0&0\\ 0&1&1&0&0&0&0\\ 1&1&0&0&0&0&0 \end{pmatrix} * \begin{pmatrix} 1&0&1\\ 0&1&0\\ 1&0&1 \end{pmatrix} = \begin{pmatrix} 1&4&3&4&1\\ 1&2&4&3&3\\ 1&2&3&4&1\\ 1&3&3&1&1\\ 3&3&1&1&0 \end{pmatrix}$$

Fig. 2.3 The mechanism of the convolution layer.

missing regions in the video inpainting task if we want to apply depth guidance into video inpainting framework. To introduce depth information into video inpainting, we designed a depth completion model to predict the depth in missing regions and a depth-guided content reconstruction model to generate reliable contents.

2.3.4 Convolutional Neural Network

CNNs are a class of deep learning models that have had a significant impact on computer vision in recent years. Originally inspired by the structure and function of the visual cortex, as proposed by Hubel and Wiesel [90] in the 1950s, LeCun *et al.* [91] developed LeNet, which applied convolutional layers to recognize handwritten zip code, in 1989. However, CNNs gained widespread popularity following the success of AlexNet [92] in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, where it significantly outperformed traditional approaches. Since then, CNNs have revolutionized the field of computer vision and many other domains. Designed to automatically and adaptively learn spatial hierarchies of features from input images, CNNs are highly effective for many tasks such as image classification, object detection, and segmentation.

The core components of CNNs include convolution layers, activation functions, and fully connected layers. Each of these components plays a critical role in the architecture and functionality of CNNs, enabling them to efficiently process and learn from image data.

Convolution layers are the fundamental element of CNNs, which use a set of learnable filters or kernels to the input data to produce feature maps. The working mechanism of convolution layer is illustrated in Fig. 2.3. Each filter detects specific patterns or features such as edges, textures, or shapes. The convolution operation can be expressed as

$$(I * K)(x,y) = \sum_i \sum_j I(x+i, y+j) K(i,j), \tag{2.7}$$

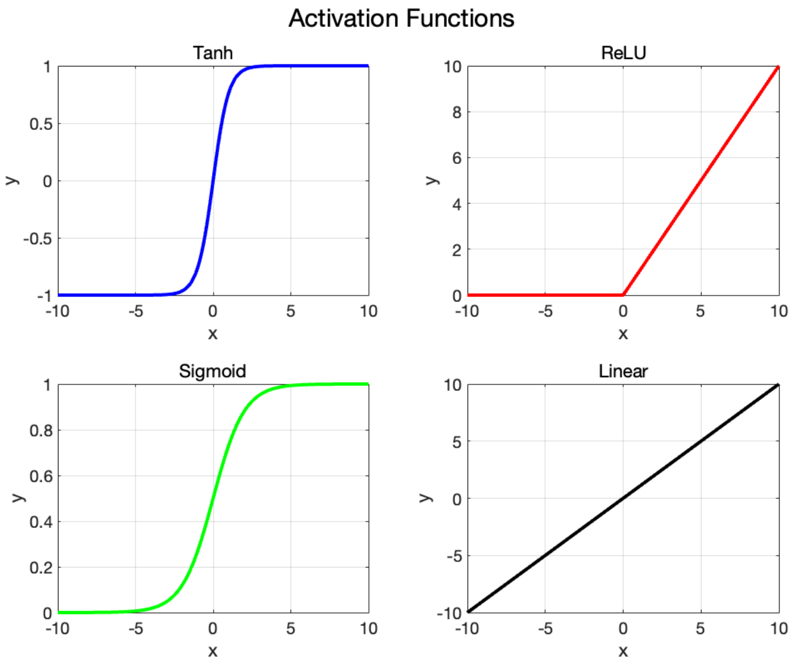


Fig. 2.4 Comparisons of common activation functions and linear function.

where I is the input image, K is the filter, and (x,y) are the coordinates of the output feature map. Convolution layers exploit local connectivity and parameter sharing, making them highly efficient in extracting local features from data with spatial hierarchies, such as images and videos.

Activation functions introduce non-linearity into neural networks, enabling them to learn and capture complex patterns during training, as illustrated in Fig. 2.4. Without activation functions, which are inherently non-linear, neural networks would be limited to learning only linear relationships, as convolutional layers and fully connected layers are linear operations. Common activation functions include Sigmoid, Tanh, and ReLU (Rectified Linear Unit). Specifically, the Sigmoid function maps input values to the range (0, 1), making it useful for binary classification tasks. It is defined as

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{2.8}$$

The Tanh function maps input values to the range (-1, 1), providing zero-centered outputs. It is defined as

$$\mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{2.9}$$

ReLU is widely used due to its simplicity and effectiveness in mitigating the vanishing gradient problem, enabling the training of deep networks. It is defined as

$$\mathrm{ReLU}(x) = \max(0,x). \tag{2.10}$$

Variants of ReLU, such as Leaky ReLU and Parametric ReLU, address the issue of zero gradients for negative input values by allowing a small, non-zero gradient when the unit is not active.

Fully Connected (FC) layers, also known as dense layers, are a crucial component of neural networks where each neuron in one layer is connected to every neuron in the previous layer. This allows for the maximum amount of interaction between neurons, enabling the network to learn complex, non-linear relationships. The operation of a fully connected layer can be mathematically described as

$$y = f(Wx+b), \tag{2.11}$$

where x and y are the input and output vectors, respectively, W is the weight matrix, b is the bias vector, and f is the activation function. Fully connected layers are used to construct Multi-Layer Perception (MLP). MLPs are a type of feedforward neural network that is stacked sequentially with multiple fully connected layers, including an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to every neuron in the subsequent layer, allowing the network to learn complex, non-linear relationships in the data.

The CNN and its variants have demonstrated impressive performance across various tasks in computer vision. One of the most impactful variants is Deformable Convolutional Network (DCN) [93, 94]. The core innovation of DCN lies in the deformable convolution operation. Deformable convolution extends the capabilities of vanilla convolution by introducing learnable offsets to the sampling locations of the convolutional kernels. This allows the kernels to adapt their shape dynamically according to the input data, enhancing the feature extraction ability of the model compared to the fixed rectangular kernels used in traditional convolutions. The working mechanism of the deformable convolution can be mathematically expressed as

$$(I*K)(x,y) = \sum_i \sum_j I(x+i+\Delta p_i, y+j+\Delta p_j) K(i,j), \tag{2.12}$$

where Δp_i and Δp_j are the learnable offsets that allow the kernel to adapt its shape to the underlying structure of the input feature map. These offsets are predicted from the input feature map using an additional convolutional layer, making the sampling process both flexible and adaptive. Compared with the standard convolution, the deformable convolution provides a larger receptive field for the model and enhances its ability to capture intricate patterns and variations in feature extraction. Consequently, DCN significantly improves performance in challenging scenarios.

Discussions: in recent years, CNNs [95, 5, 55–57] have significantly contributed to video inpainting by providing powerful tools for feature extraction and spatial representation. Leveraging their ability to learn hierarchical features, CNNs effectively capture textures, edges, and structural information from video frames, making them suitable for reconstructing missing regions with high fidelity. Their inherent local connectivity and weight-sharing mechanisms enable efficient processing of frame-level spatial information, which is crucial for generating visually plausible inpainting results. Additionally, CNN-based architectures [55–57] can be extended to process temporal information by stacking frames or using 3D convolutions, allowing them to model short-term temporal consistency. However, CNNs face limitations when dealing with complex video inpainting tasks. Their reliance on fixed receptive fields can hinder their ability to capture long-range spatio-temporal dependencies, which are essential for ensuring global coherence across frames. Moreover, CNNs may struggle with large missing regions or complex scenes involving multiple layers and motions due to their focus on local features. These limitations highlight the need for integrating CNNs with complementary mechanisms, such as attention-based models or recurrent networks, to enhance their ability to handle the challenges of video inpainting.

2.3.5 Vision Transformer

Transformer model was originally introduced by Vaswani *et al.* [96] in 2017 in NLP. Compared to the efficiency of CNNs in aggregating local features, Transformers offer a more robust framework for integrating contextual information from across the entire input while enabling parallelized training, making them a powerful architecture in machine learning. The core idea behind the Transformers is Multi-head Self Attention (MSA), which allows the Transformer model to dynamically weigh the importance of different input tokens, mimicking how humans perceive information from the real world. This model has recently been adapted for various tasks in computer vision, leading to significant advances in the field.

ViT [65] is one of the earliest and most influential adaptations in computer vision, where images are split into a sequence of patches and processed similarly to sequences in NLP tasks. The framework of ViT is illustrated in Fig. 2.5. More specifically, an input image is first divided into non-overlapping patches of fixed size. Each patch is then flattened into a vector and linearly embedded into a lower-dimensional space. These embeddings are combined with position embeddings to retain spatial information. The resulting sequence of patch

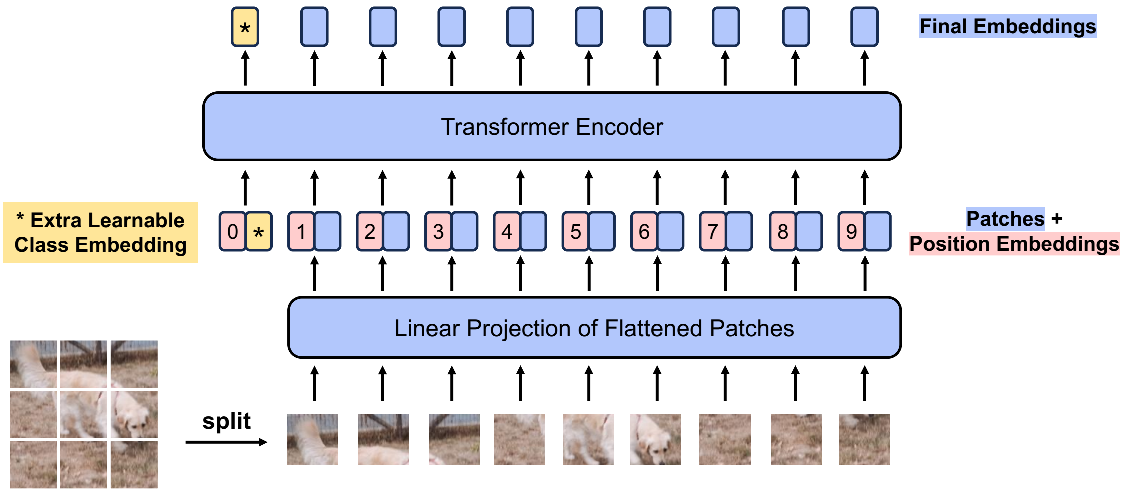

*
Final Embeddings
Transformer Encoder
* Extra Learnable Class Embedding
0 *
1
2
3
4
5
6
7
8
9
Patches + Position Embeddings
Linear Projection of Flattened Patches
split

Fig. 2.5 The visualized framework of ViT that applies vanilla Transformer for image recognition task.

embeddings is fed into a Transformer encoder, which consists of multiple layers of MSA and Feed Forward Network (FFN). In MSA, each input patch embedding is linearly projected into query, key, and value for the calculation of attention scores, allowing the model to capture global relationships for each pair of patches. Mathematically, the self-attention mechanism is defined as

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.13}$$

where Q, K, and V represent queries, keys, and values, and d_k is the dimension of the key vectors. The MSA mechanism enables Transformer to effectively capture long-range dependencies and model complex spatial relationships within images.

Following ViT, numerous Transformer variants have been developed for computer vision. For example, Liu *et al.* [97] introduced Swin Transformer, which uses hierarchical feature maps and shifted windows to enhance computational efficiency and scalability for high-resolution images, making it suitable for a range of dense prediction tasks like object detection and segmentation. Zhu *et al.* [93] designed DEtection TRansformer (DETR), integrating Transformers with convolutional backbones to create end-to-end object detection models that leverage self-attention for improved spatial context understanding. Additionally, Wu *et al.* [98] developed Convolutional vision Transformer (CvT), combining the strengths of CNNs and Transformers to benefit from both local feature extraction and global context modeling. Overall, these Transformer variants have achieved impressive performance across various high-level tasks, such as image classification [99, 100], segmentation [101, 102], and object detection [103, 104]. Moreover, they have also been effectively applied to many low-level tasks, such as image super-resolution [105], restoration [106], and enhancement [107].

Discussions: Transformers [58, 59, 8, 9] have brought transformative advancements to video inpainting by leveraging their ability to model long-range dependencies and capture global contextual information. Unlike CNNs, which are limited by fixed receptive fields, Transformers use a self-attention mechanism to dynamically weigh the importance of each token in relation to others, enabling the propagation of reference information across spatial and temporal dimensions. This makes them particularly effective in reconstructing missing regions in videos with large gaps or complex motions. For example, Transformers can link distant frames to maintain global coherence, ensuring that the reconstructed content aligns with the broader context of the video. Additionally, their flexibility in handling sequences allows them to process spatio-temporal relationships in a unified framework, making them versatile for both object removal and video restoration tasks. However, Transformers also have limitations. Their quadratic computational complexity with respect to the input size makes them resource-intensive, especially for high-resolution video inpainting. Furthermore, they require large-scale datasets and extensive training to generalize effectively, which may limit their applicability in domains with limited data availability. These challenges highlight the need for optimized Transformer architectures, such as sparse attention mechanisms or hierarchical models, to balance efficiency and effectiveness in video inpainting tasks.

2.3.6 Generative Adversarial Network

GAN is a class of machine learning framework designed to generate realistic and diverse data across various domains, especially in computer vision. Introduced by Ian Goodfellow *et al.* [45] in 2014, GANs have rapidly become an important generative model, impacting areas such as image and video synthesis, generation, and enhancement. As shown in Fig. 2.6, the architecture of GANs consists of two neural networks: a generator, which creates data resembling real-world samples, and a discriminator, which evaluates the authenticity of the generated data. Specifically, the generator network takes a random noise vector as input and transforms it into a data sample that resembles the training data. The generator typically uses CNNs or Transformers to create the generated data, such as transforming the input noise vector into a full-sized image. The discriminator network takes a data sample (either from the real dataset or generated by the generator) as input and outputs a probability score indicating whether the sample is real or fake. This network is typically implemented using convolutional layers or Transformer blocks that extract features from the input data and make a binary classification.

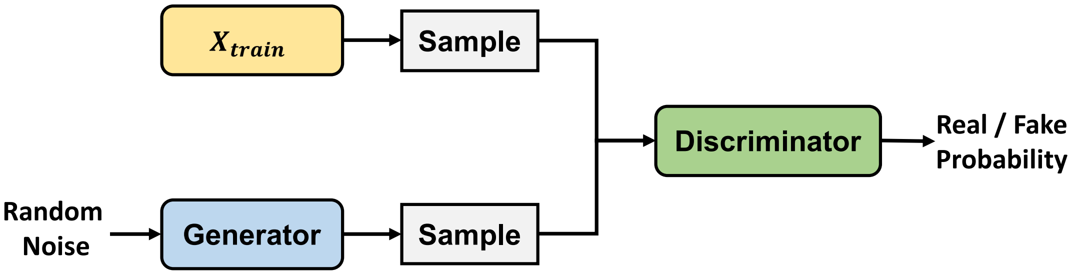


Fig. 2.6 The architecture of GAN.

The training process for GAN is essentially a Minimax game, simultaneously improving through adversarial training. The generator aims to produce increasingly realistic data, while the discriminator aims to become better at distinguishing between real and generated data. This adversarial process is formulated as a minimax optimization problem as

$$\min_{G}\max_{D}\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]+\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))], \tag{2.14}$$

where $x \sim p_{\text{data}}(x)$ denotes samples from the real data distribution, $z \sim p_z(z)$ denotes samples from the noise distribution, G is the generator network that generates data samples from the noise distribution $p_z(z)$, and D is the discriminator network that outputs the probability that a given data sample is real.

There are many variations of GANs which enhance the flexibility and stability of standard GANs, making them more effective for a variety of tasks. For example, Mirza *et al.* [108] designed conditional Generative Adversarial Network (cGAN) to produce results under the control of additional input by incorporating additional information, such as class labels or image data. Zhu *et al.* [109] designed Cycle-consistent Generative Adversarial Network (CycleGAN) to facilitate image-to-image translation without requiring paired training data by using cycle consistency loss. Karras *et al.* [110] introduced Style-based Generative Adversarial Network (StyleGAN) to produce high-resolution and highly detailed images for applications, such as face generation and artistic content creation.

Discussions: In recent years, GANs [56–59] have contributed significantly to image and video inpainting by enabling the generation of realistic and contextually coherent results. In video inpainting, the combination of a generator and discriminator ensures that the inpainted video frames blend seamlessly with the surrounding content while maintaining temporal coherence across frames. This is crucial for avoiding visual artifacts and preserving spatio-temporal consistency in the inpainted results. Additionally, GANs can incorporate additional information, such as optical flow and depth information, to produce content with clearer structure and better textures. Consequently, GANs provide a powerful framework for addressing the challenges of video inpainting. However, GANs also face several limitations. They are challenging to train due to instability in the adversarial process, often leading to mode collapse or difficulty in converging. Moreover, GANs require extensive computational resources and large datasets to produce high-quality results.

2.3.7 Evaluation Metrics

Evaluation metric is an important aspect to assess the performance of video inpainting algorithms in reconstructing missing regions. In order to provide a comprehensive and objective evaluation for inpainting approaches, various quantitative metrics are adopted to measure the quality of inpainted results, such as Peak Signal-to-Noise Ratio (PSNR) [111], Structural Similarity Index Measure (SSIM) [111], Learned Perceptual Image Patch Similarity (LPIPS) [112], Video Fréchet Inception Distance (VFID) [113], and flow warping error (E_{warp}) [114]. Each of these metrics provides a different perspective on the visual quality of inpainted results.

PSNR [111] is a widely used metric for evaluating the quality of reconstructed images and videos, which measures the ratio between the maximum possible power of a signal and the power of noise that affects the quality of its representation. The PSNR is calculated using the Mean Square Error (MSE) between the original and reconstructed frames and is expressed in deciBels (dB). The formula for PSNR is given by

$$\mathrm{PSNR}(I,\hat{I}) = 10\log_{10}\left(\frac{\mathrm{MAX}_\mathrm{I}^2}{\mathrm{MSE}}\right) = 20\log_{10}\left(\frac{\mathrm{MAX}_\mathrm{I}}{\mathrm{MSE}}\right), \tag{2.15}$$

where MAX_I is the maximum possible pixel value of the image (this value is 255 for 8-bit images), and MSE is defined as

$$\mathrm{MSE}\left(I,\hat{I}\right) = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-\hat{I}(i,j)]^2, \tag{2.16}$$

where I is the original frame, $\hat{I}$ is the inpainted frame, and m and n are the spatial dimensions of the frames. A higher PSNR value indicates that the inpainted image is of higher quality and has a lower reconstruction error. However, it is important to note PSNR is not a perfect metric and may not always correlate with human perception of image quality.

SSIM [111] is a perceptual metric that measures image degradation as perceived change in structural information. The SSIM index ranges from -1 to 1, where 1 indicates the most perfect structural similarity. The SSIM between two $N \times N$ patches x and y of I and $\hat{I}$ is computed as

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{2.17}$$

where μ_x and μ_y are the average pixel values of x and y, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance between x and y, and c_1 and c_2 are constants to stabilize the division with weak denominator values. Typically, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, where L_I is the dynamic range of the pixel values (for 8-bit images, $L = 255$), and k_1 and k_2 are small constants (e.g., $k_1 = 0.01$ and $k_2 = 0.03$). SSIM can be used to evaluate the performance of video inpainting algorithms by measuring the structural similarity between an original and inpainted results.

LPIPS [112] is a modern metric to assess perceptual similarity between images. Unlike traditional metrics such as PSNR and SSIM, which concentrates on the differences between pixel or patches, LPIPS computes the distance between images in the high-level feature space of a pre-trained DNN, such as Visual Geometry Group (VGG) [115] or AlexNet [92], which aligns more closely with human visual perception. LPIPS between original frame I and inpainted frame $\hat{I}$ is computed as

$$\mathrm{LPIPS}(I,\hat{I}) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} ||w_l \cdot (\phi_l(I)_{hw} - \phi_l(\hat{I})_{hw})||_2^2, \tag{2.18}$$

where ϕ_l represents the features extracted from the l-th layer of a pre-trained network, w_l are learned weights for each layer, and H_l and W_l are the height and width of the feature maps at layer l. By introducing learned perceptual features, LPIPS provides a more accurate and meaningful evaluation of visual quality compared to traditional metrics, especially in applications involving complex textures and structures.

VFID [113] is also an modern DNN-based metric which is designed to evaluate the quality of generated or reconstructed video sequences by measuring the similarity between the feature distributions of real and synthetic videos. Developed from the Fréchet Inception Distance (FID) [116], which is commonly used to assess the quality of images, VFID extends FID to videos. To compute VFID, we firstly extract features of each frame using a pre-trained DNN, such as Inflated 3D convnet (I3D) [117], which can capture spatio-temporal information from videos. Then we compare the mean and covariance of these feature distributions for real videos P and generated videos Q. The formula of VFID is given as

$$\mathrm{VFID}(P,Q) = ||\mu_P - \mu_Q||^2 + \mathrm{Tr}(\Sigma_P + \Sigma_Q - 2(\Sigma_P \Sigma_Q)^{1/2}), \tag{2.19}$$

where (μ_P, Σ_P) and (μ_Q, Σ_Q) are the mean and covariance of the feature vectors from the real and inpainted videos, respectively, and Tr represents the trace in matrix computation. A lower VFID score indicates a more similar distribution between real videos and reconstructed results, representing generated videos with better quality and diversity. VFID provides a comprehensive perceptual assessment of video quality from both spatial and temporal dimensions, making it an effective metric to evaluate the performance of video inpainting methods.

E$_{warp}$ is an important metric for the evaluation of the temporal consistency in inpainted videos. This metric measures temporal consistency by comparing the similarity of corresponding pixels from consecutive frames. To calculate E$_{warp}$, frame warping is firstly implemented to compensate the displacements between consecutive frames and we typically adopt optical flow estimation network, such as FlowNet 2.0 [77] or RAFT [78]. The warping error is then calculated with the MSE between the warped inpainted frame I_w and the neighboring frame I_n as

$$E_{warp} = \frac{1}{N} \sum_{i=1}^{N} ||I_w(i) - I_n(i)||^2, \tag{2.20}$$

where N is the number of pixels. A lower warping error represents better temporal consistency for reconstructed regions in inpainted results.

3

Short-Long-Term Propagation-Based Video Inpainting for Object Removal

3.1 Introduction

In this chapter, we will introduce a SLTPVI approach for object removal. Object removal [118, 3, 119] is a critical area of research in video inpainting. When unwanted objects are removed artificially or automatically from a given video, missing regions appear in the frames where these objects once existed. The primary goal of object removal in video inpainting is to restore these missing parts of a video sequence in a visually plausible manner, maintaining temporal consistency and spatial coherence across frames. This technique was initially implemented based on image inpainting [120, 22] and has been further developed by incorporating the temporal correlation of video frames [46].

The patch-based approaches [46, 47, 121, 48, 3, 49] were designed to fill the missing regions by using the available patches collected spatially or temporally from known regions of the video. In general, the filling of the missing regions could be implemented in either a greedy fashion [46, 47, 121] or a global fashion [48, 3, 49]. The methods effectively processed the non-stationary inpainting scenes, although searching for such patches always resulted in rather high computational complexity, which limited their speed and the range of applications.

In contrast, propagation-based methods [3, 51, 6] have been developed based on the spatio-temporal correlation of video frames. With the guidance of optical flow or homography, the source information collected for inpainting was propagated throughout the video so that the empty areas can be filled by the composed content with high temporal coherence. The performance of these approaches depended upon the propagation efficiency. Consequently, how to implement effective information propagation remains a key issue for the design of the propagation-based inpainting.

Recently, deep learning-based methods, including the application of CNN [64, 95, 55, 56] and Transformer models [58, 59, 8], have demonstrated impressive inpainting results for videos. Many of CNN-based methods designed the end-to-end schemes and applied the 3D convolution to fuse spatial-temporal features for the synthesis of content. However, these approaches often produced coarse textural detail due to the lack of effective alignment of features. Attention-based methods have also been developed based upon the spatio-temporal context aggregation module to compose content; although, these approaches could not produce fine-grained textures and thus could not process complicated video scenes. Note that the deep learning-based approaches suffered from high computational cost and complexity, especially at the training stage. Moreover, the robustness of these methods was not as good as expected. Their performance highly depends upon the training datasets.

Despite the significant progress made in video inpainting, several challenges remain in designing an effective scheme for object removal due to the dynamic nature of video content. These challenges include handling diverse and complex motion patterns, varying lighting conditions across frames, ensuring long-term temporal coherence, and achieving visually satisfying effects. Diverse and complex motion patterns often make it difficult to accurately predict the movement of objects being removed, leading to artifacts or unnatural transitions in the inpainted regions. Varying lighting conditions complicate the maintenance of consistent color and texture, resulting in noticeable discrepancies. Ensuring long-term temporal coherence requires collecting and integrating reference information from multiple frames, which increases computational complexity and memory usage. Additionally, real-world videos often contain occlusions and complex interactions between multiple objects, further complicating the task of generating visually satisfying results. These challenges necessitate the development of advanced algorithms capable of maintaining consistency and realism across dynamic video content to achieve impressive performance for object removal.

In order to tackle these problems, we proposed a combined module comprising SLTPVI to fill the missing areas of the video with removed objects. The proposed SLTPVI was composed by both STPI and LTPI. Specifically, the STPI module was designed to fill a single frame with the reference information obtained from its adjacent neighbors. In STPI, a depth-guided mesh-warping model with an illumination adaptation algorithm and a progressive fusion algorithm were developed to fill the missing region with high quality information. The LTPI module was constructed based on STPI but uses more reference information from more distant frames. Moreover, it was designed to inpaint the whole video by propagating the spatio-temporal information of frames through the video. In LTPI, the intra Group Of Pictures (GOP) inpainting strategy and the inter GOP inpainting strategy were developed to reduce the computational complexity. The main contributions of this paper are summarized as follow:

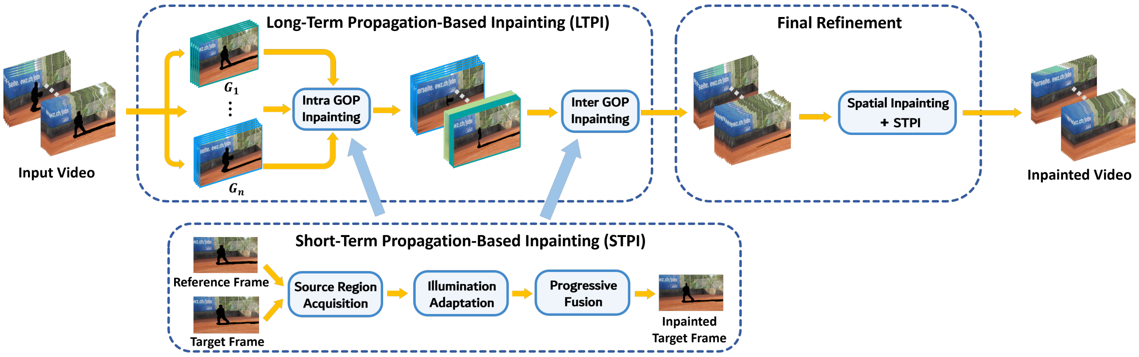


Fig. 3.1 Schematic showing the pipeline of our proposed method. Firstly, the input video is divided into several GOP. Then, the intra and inter GOP inpainting are sequentially performed to compose the LTPI module and both are constructed based on the STPI module. The STPI module consists of source region acquisition, illumination adaptation and progressive fusion, which are employed to obtain the reliable source region and seamlessly transfer it to target region. The final refinement, which is composed by spatial inpainting and STPI, is carried out if the frames cannot be completely filled by LTPI.

- Firstly, we proposed combining two temporal propagation-based modules, i.e., STPI and LTPI, to implement a high-efficiency video inpainting by using the reliable source information obtained based on the spatio-temporal correlation of frames.

- We then designed a depth-guided mesh-warping model to predict the moving motion for missing regions of video data, including both single-layer and multi-layer alignment-based source region acquisition methods for different scenarios within the STPI.

- We next proposed an illumination adaptation algorithm to eliminate the brightness variation of frames together with progressive fusion algorithm for STPI, seamlessly transferring the source region to the target area.

- Finally, we developed both intra and inter GOP inpainting strategies for LTPI, where all the frames of a video are separated into several groups so that the reference information can be propagated forward and backward to implement a progressive inpainting with a high temporal consistency and a low complexity.

3.2 Methodology

3.2.1 Overview

The pipeline of our proposed SLTPVI is illustrated in Fig. 3.1, where both the STPI and LTPI modules collect the correlated spatio-temporal information to fill the missing areas of video frames. Specifically, the STPI module is designed to inpaint a single frame with the reference information provided by its adjacent neighbors. The LTPI module is designed to inpaint the whole video, with the aim of reducing complexity and maintaining high temporal consistency. The pipeline of activity is constructed based on STPI but can obtain more reference information from the long-distance frames. In addition, the final refinement adopted in SLTPVI is used to inpaint the frames which cannot be completely filled by LTPI.

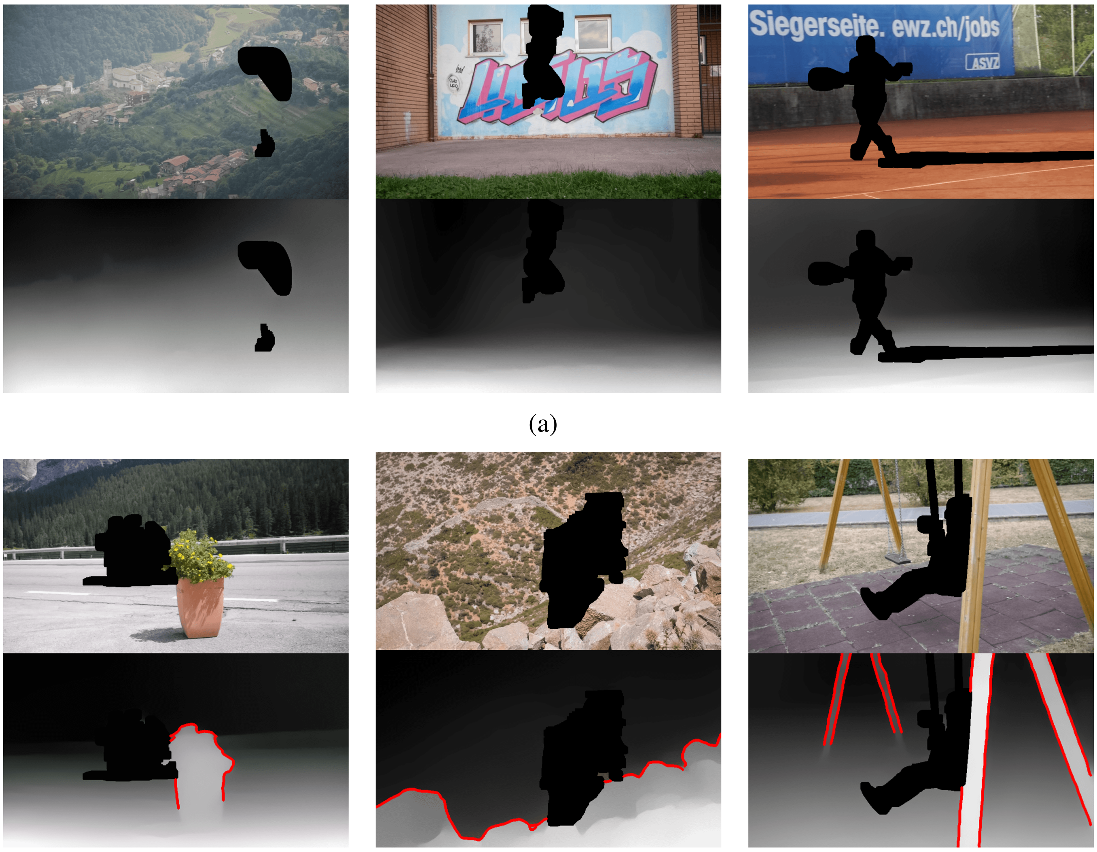
The STPI module consists of source region acquisition, illumination adaptation and progressive fusion. To acquire reliable source regions, the depth-guided mesh-warping model is designed to predict the motion for missing regions, which can be divided into both single-layer and multi-layer alignments, according to the continuity of the estimated depth map [122].

The LTPI module is implemented by progressively applying STPI to video frames. In LTPI, all the frames of a video are firstly divided into GOPs. Then, STPI is performed on the frames of each GOP along both the forward and backward directions, achieving the intra GOP inpainting. Subsequently, STPI is applied to the frames of the two adjacent GOPs, implementing the inter GOP inpainting. If some frames of a video cannot be completely inpainted by LTPI, the spatial inpainting coupled with STPI is performed to complete the inpainting, which accordingly achieves the final refinement.

3.2.2 Short-term Propagation-based Inpainting

The STPI module consists of source region acquisition, illumination adaptation and progressive fusion. For source region acquisition, we use the single-layer and multi-layer alignment-based techniques to obtain source region according to whether the frame includes multiple layers.

In our proposed STPI module, the inpainting of a target frame relies upon the source information collected from its neighboring reference frames. However, varied motion for different layers often induces misalignment between frames, often resulting in difficult acquisition of reliable information for inpainting. To tackle this problem, we firstly align the reference frame F_r to the target frame F_t. Then, with the user-specified mask, we obtain the source region from the aligned F_r to fill the missing region of F_t. In addition, we design two source region acquisition methods based on the single-layer alignment and multi-layer alignment for two scenarios. Different alignment is adopted according to the continuity of the depth map [122], where the existence of discontinuous object edges in the depth



(a)

(b)

Fig. 3.2 Examples of single-layer and multi-layer scenes. The discontinuous depth boundary detected by Canny is highlighted in red color. (a) Single-layer scenes. First row: video frames. Second row: depth maps. (b) Multi-layer scenes. First row: video frames. Second row: depth maps.

map indicates the existence of different layers in the frame. In our work, a Canny edge detector [123] is used to detect the discontinues edge in the depth, as illustrated in Fig. 3.2.

Single-layer Alignment-based Source Region Acquisition

For the video whose frames do not contain obvious foreground and background layers, i.e., composed by only one layer, we design a single-layer alignment-based method to acquire the source region. It is based on the mesh-warping model [124–126] to align the reference and the target frames. This model warps a frame with local homography for alignment and introduces mesh grids to optimize the homography. Moreover, it does not separate the frame into different planes, which makes it applicable for the alignment of frames with single layer and continuous depth.

To construct the mesh-warping model, the matched features of F_t and F_r should be obtained first. Before generating features, we initially fill the missing regions of F_t and F_r by using the inward interpolation and the available boundary pixels. This processing constructs a smooth region to avoid the matched features of F_t and F_r falling around the boundaries of missing regions. After the initialization, we generate the SURF [68] of both F_t and F_r. Then, we gather the matched features of these frames and apply the RANSAC [127] to remove undesired features. Finally, we use the remaining features to construct the mesh-warping model, which produces optimal local homography and accordingly guarantees a low warping loss for frame alignment.

In the mesh-warping model, assuming that $\hat{F}_r$ is a warped reference frame, let $\hat{V}_i = [\hat{v}_i^1, \hat{v}_i^2, \hat{v}_i^3, \hat{v}_i^4]^T$ represent the vertex vector of a grid cell for $\hat{F}_r$. The optimal warping is determined by minimizing

$$E(\hat{V}) = E_d(\hat{V}) + \eta E_s(\hat{V}) \tag{3.1}$$

where $\hat{V}$ is composed by all the warped grid vertices, E_d and E_s are the data term and similarity transformation term, respectively, and η is the weighting factor. The data term is employed to minimize the distance of corresponding points between target frames and warped reference frame. The similarity transformation term is used to preserve the shape of the global content, which minimizes the deviation of each output grid cell from a similarity transformation of its corresponding input grid cell.

Specifically, by performing the bilinear combination on a mesh grid cell of $\hat{F}_r$, we obtain the feature $\hat{f}_i$ of $\hat{F}_r$ as $\hat{f}_i = c_i \hat{V}_i$, where $c_i = [c_i^1, c_i^2, c_i^3, c_i^4]$ is the bilinear weighting vector. The above minimization problem is quadratic and can be readily solved using a standard sparse linear solver as suggested in [124]. Then, the data term is defined

$$\begin{aligned} E_d(\hat{V}) &= \textstyle\sum_i \|\hat{f}_i - f_i\|_2^2 \\ &= \textstyle\sum_i \|c_i \hat{V}_i - f_i\|_2^2. \end{aligned} \tag{3.2}$$

The similarity term is defined to constrain the frame warping with small deformations. Note that each grid cell can be split into two triangles [124, 125]. Assuming that $\triangle v v_1 v_2$ is a triangle with three vertices v, v_1 and v_2, v can be represented by v_1 and v_2 in a local orthogonal coordinates system

$$v = v_1 + (u_1 + u_2 R_{90})(v_2 - v_1) \tag{3.3}$$

where u_1 and u_2 are the coordinates of v in the local coordinate system and R_{90} is the 90° rotation matrix for v [124]. Based on Eq. (3.3), the similarity term is defined as

$$E_s(\hat{V}) = \textstyle\sum_{\hat{v}} \|\hat{v} - [\hat{v}_1 + (u_1 + u_2 R_{90})(\hat{v}_2 - \hat{v}_1))]\|^2 \tag{3.4}$$

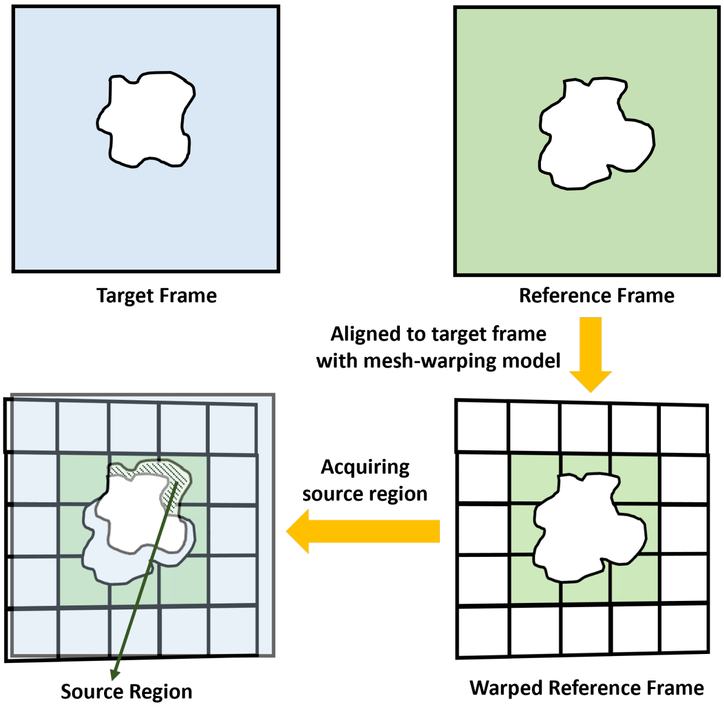


Fig. 3.3 Source region acquisition with single-layer alignment.

where $\hat{v}$, $\hat{v}_1$ and $\hat{v}_2$ represent three vertices of the triangle in the optimized mesh grid. After substituting Eqs. (3.2) and (3.4) into Eq. (3.1), we can minimize the resulting $E(\hat{V})$ with a sparse linear system solver, which accordingly generates the optimal mesh grid.

Once the optimized mesh grid is obtained, we can generate the local homography for each grid cell. To achieve a low complexity, we warp the local area rather than the whole reference frame to form the source region for the target region. The corresponding procedure is illustrated in Fig. 3.3. Firstly, based on the user-specified mask, we produce the warped reference sub-mask for each grid cell with its homography. Secondly, we form the warped reference mask $\hat{\Omega}_r$ with all the resulting sub-masks. Thirdly, we generate a binary mask Ω_o to determine whether warp local area or not. Also, Ω_o is used to obtain the reference information for inpainting from the warped local area. In this work, Ω_o is obtained as

$$\Omega_o = \Omega_t \odot (I - \hat{\Omega}_r), \tag{3.5}$$

where Ω_t is the target mask, $\odot$ is the element-wise product, and I is a mask whose elements are all 1. Finally, if Ω_o is not an empty mask, i.e., the mask whose elements are all 0, we will use it to obtain the inpainting information from the local reference area. To obtain this area, we firstly form a regular mask, denoted Ω_R, with the grid cells of the given mask of the reference frame. Then, we get the reference region R_r as

$$R_r = \Omega_R \odot F_r. \tag{3.6}$$

Next, we warp R_r with the local homography matrixs and obtain the source region for inpainting as

$$R_s = \Omega_o \odot warp(R_r). \tag{3.7}$$

If Ω_o is an empty mask, it indicates that we cannot get source region from neighboring frame For this scenario, the LTPI module is adopted and it collects source region from long-distance frame for inpainting.

Multi-layer Alignment-based Source Region Acquisition

Single-layer alignment is used to obtain source region for inpainting the video whose frames do not contain obvious foreground and background, i.e., with continuous depth. It works well in the single-layer scenario because all the objects within a frame have similar motions. However, in the multi-layer scenario, the foreground and background produce a discontinuity in the depth, so resulting in different motions between them. If the single-layer alignment is applied to this scenario, the features of foreground are always treated as outliers and accordingly eliminated. Therefore, applying the single-layer alignment to multi-layer scenarios cannot achieve desired alignment. To tackle this problem, we separately warp different layers of the reference frame to implement the multi-layer alignment. With this alignment, we can obtain the reliable source region. The multi-layer alignment-based source region acquisition consists of depth estimation, depth-guided separation and source region acquisition, as illustrated in Fig. 3.4.

With the depth maps of target videos estimated by [122], we designed a depth-guided method to separate the foregrounds and backgrounds of frames. Depth map is obtained in depth-aided alignment decision. After separation, we aligned the corresponding layers of the reference and target frames to acquire the appropriate source information. To implement the layer separation, the frame is firstly split into a number of super-pixels. Then, according to the depth and color information, these super-pixels are clustered into two classes. With this classification, the frame is finally separated into foreground and background.

Super-pixels are perceptually meaningful groups of pixels that represent irregularly shaped regions within an image, where the pixels within each super-pixel share similar attributes such as texture, color, and depth. Unlike individual pixels, which may represent fine-grained and noisy information, super-pixels aggregate local information into larger, coherent regions, making them a useful tool for high-level image and video processing tasks.

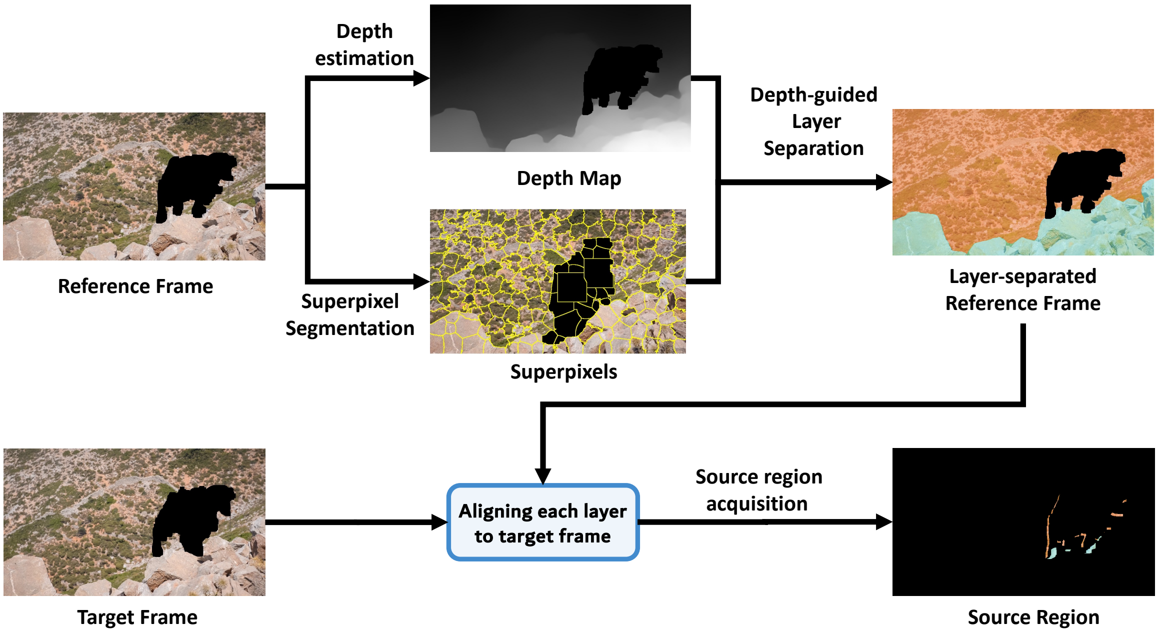

Depth estimation
Depth Map
Depth-guided Layer Separation
Reference Frame
Superpixel Segmentation
Superpixels
Layer-separated Reference Frame
Aligning each layer to target frame
Source region acquisition
Target Frame
Source Region

Fig. 3.4 Source region acquisition with multi-layer alignment. The depth information of valid regions is firstly estimated for references frames. Then the reference frames are separated into multiple layers based on the depth information. Each layer is aligned to the target frame with mesh warping model, acquiring the source regions from different layers of reference frames.

By reducing the number of units to process, super-pixels enable efficient segmentation and provide a robust representation of structures within a frame.

In our work, we leverage super-pixels to facilitate the segmentation of multiple layers in a video frame, addressing challenges associated with complex scenes containing both foreground and background layers. Super-pixels simplify the layer-separation process by grouping adjacent pixels with similar features, thereby reducing noise and redundancy in the input data. Specifically, we utilize the Simple Linear Iterative Clustering (SLIC) [128] algorithm to convert an $H \times W$ video frame into an $H/n \times W/n$ super-pixel map, where n denotes the size of each super-pixel cluster. The usage of super-pixels for layer-separation can bring two benefits. First, they provide a compact and meaningful representation of visual information, reducing the complexity of multi-layer separation in scenes with intricate structures and overlapping objects. Second, super-pixels offer flexibility and adaptability for analyzing various spatial features, such as texture, color, and depth, which are crucial for distinguishing between foreground and background layers. It significantly reduces the computational complexity of processing high-resolution frames, as operations are performed at the super-pixel level rather than on individual pixels.

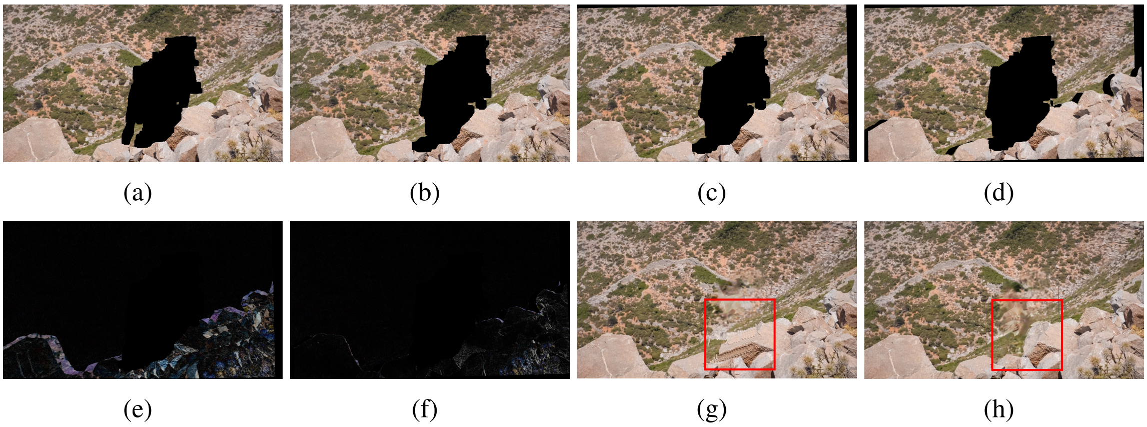


Fig. 3.5 Results for the verification of multi-layer alignment. (a) Target frame; (b) reference frame; (c) single-layer alignment result; (d) multi-layer alignment result; (e) single-layer alignment error; (f) multi-layer alignment error; (g) inpainting with single-layer alignment; and (h) inpainting with multi-layer alignment.

To classify the super-pixels into distinct layers, we compute the mean values of color and depth for each super-pixel cluster. These mean values serve as low-dimensional representations, enabling efficient layer classification. Using these representations, we apply the Meanshift algorithm [129], which groups super-pixels into coherent layers based on their feature similarities. This method not only enhances segmentation accuracy but also reduces the computational burden compared to processing raw pixel-level data.

In our work, we combine the color and depth of a super-pixel to form the classification parameter

$$t_i = |color_{mean} - color_i| + \lambda |depth_{mean} - depth_i| \tag{3.8}$$

where $color_{mean}$ and $depth_{mean}$ denote the mean values of color and depth of a cluster, respectively, $color_i$ and $depth_i$ are the color and depth values of a specific super-pixel, respectively, and λ is the depth weight. By comparing t_i with a given threshold T, we can accordingly divide all the super-pixels into two clusters. With these clusters, we separate the frame into foreground ($t_i > T$) and background ($t_i \leq T$). Furthermore, according to the resulting layers of a reference frame, we can divide the user-specified mask into the corresponding foreground and background masks.

After the depth-guided layer separation, the reference frame is divided into foreground and background. Then, we separately apply the single-layer-based acquisition to these layers to obtain the corresponding source regions for them. Finally, we combine these regions together to form source region for target frame.

To verify the effectiveness of multi-layer alignment-based acquisition, we apply the method to the frames with multiple layers and compared the results with those obtained using the single-layer alignment. The corresponding results, including the aligned reference frames, the alignment errors and the final inpainting results are all given in Fig. 3.5. The results presented in Fig. 3.5 indicate that using the multi-layer alignment-based methods provides reliable source information, which accordingly generates better inpainting results.

Illumination Adaptation

Any illumination changes during video capture often induces brightness variation between frames. If we directly migrate the acquired source region to the target frame, it will result in apparent boundary effects in the inpainted frame. To tackle this problem, we design an illumination adaptation algorithm and apply it to the obtained source region before transferring it to target frame.The illumination adaptation occurs in the LAB color space and the source region is converted from the RGB color space to the LAB space. After the color conversion, the L channel of the source region, denoted L_s, is adjusted as

$$\hat{L}_s = \alpha L_s + \beta \mu \tag{3.9}$$

where α is the scaling factor , β is the compensation term and $\mu = [1,...,1]^T$. In our method, the optimal (α,β) is determined by minimizing the difference between the source region L_s and the target region L_t in the LAB space. However, the target information is not available in the inpainting task. To solve this problem, we determine (α,β) based upon the common available information around L_t and L_s. Then, we use the resulting (α,β) to adjust L_s.

Let $\bar{L}_s$ and $\bar{L}_t$ represent the available neighboring areas for L_s and L_t, respectively. In our work, $\bar{L}_s$ and $\bar{L}_t$ are obtained by performing a specific mask Ω_L on the source region and the target region, respectively, where $\Omega_L = (I - \hat{\Omega}_r) \odot (I - \Omega_t)$. Meanwhile, due to the existence of local difference and warping error, some corresponding pixels of $\bar{L}_s$ and $\bar{L}_t$ are quite different to each others, which always degrades the accuracy to find optimal (α,β). To solve this problem, we use the absolute deviation-based method [130] to exclude these pixels. After that, the optimal (α,β) is determined by

$$(\hat{\alpha}, \hat{\beta}) = \underset{(\alpha,\beta)}{\arg\min} \|\bar{L}_t - (\alpha \bar{L}_s + \beta \mu)\|_2^2. \tag{3.10}$$

The closed-form solution to Eq. (3.10) is obtained by using the least squares estimation, i.e.,

$$\begin{cases} \hat{\alpha} = \dfrac{\bar{L}_s^T \bar{L}_t - \bar{L}_s^T \mu}{\bar{L}_s^T \bar{L}_s - (\bar{L}_s^T \mu)^2} \\ \hat{\beta} = \dfrac{(\bar{L}_s^T \bar{L}_s)(\bar{L}_t^T \mu) - (\bar{L}_s^T \bar{L}_t)(\bar{L}_s^T \mu)}{\bar{L}_s^T \bar{L}_s - (\bar{L}_s^T \mu)^2}. \end{cases} \tag{3.11}$$

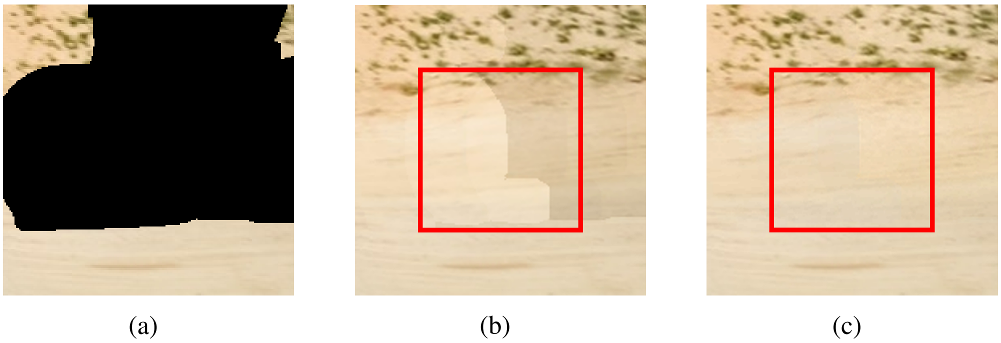


Fig. 3.6 Results for the verification of illumination adaptation. (a) Local region of target frame; (b) with illumination adaptation; and (c) without illumination adaptation.

We adjust L_s according to Eq. (3.9) and then convert it with the chrominance components to the RGB space. The converted result is finally used to fill the target region.

We present exemplar results in Fig. 3.6 to demonstrate the effectiveness of our proposed illumination adaptation, where the inpainting results with and without the adaptation are both given in Fig. 3.6. According to these results, it is found that the illumination adaptation effectively reduces boundary effects and produces more pleasant inpainting result.

Progressive Fusion

To guarantee the continuity of the filled region and its neighboring area, we design a progressive fusion algorithm to seamlessly transfer the neighboring area of the source region to the neighboring area of the target region. Specially, we combine those pixels around the boundary of the source region with the pixels at the same locations in the target region to fill the neighboring area of the target area. Let d denote the city-block distance between a selected neighboring pixel $p_s(i,j)$ and the boundary of the source region. In this work, we select a number of neighboring pixels with different distances, i.e., $d = 1,2,...,d_{max}$, to progressively fuse them with the neighboring pixels of the target region, where d_{max} is the maximum range to collect pixels. With this distance, we define a weighting factor for each selected pixel as $\omega = 1 - d/d_{max}$. Then, we combine $p_s(i,j)$ with the corresponding pixel $p_t(i,j)$ in the target region to generate a fused pixel $\hat{p}_t(i,j)$

$$\hat{p}_t(i,j) = \omega \cdot p_s(i,j) + (1-\omega) \cdot p_t(i,j). \tag{3.12}$$

We demonstrate the effectiveness of the progressive fusion by comparing the inpainting results obtained with and without it. The corresponding results are given in Fig. 3.7, which shows that adopting progressive fusion in our method produces a smoother result.

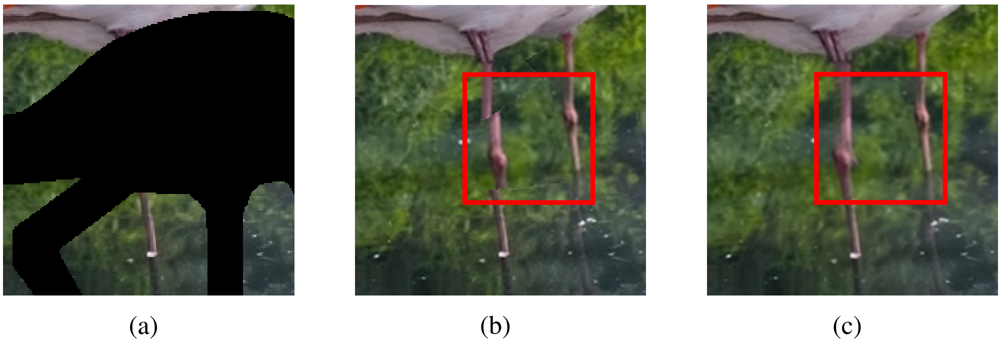


Fig. 3.7 Results for the verification of progressive fusion. (a) Local region of target frame; (b) with progressive fusion; and (c) without progressive fusion.

3.2.3 Long-term Propagation-based Inpainting

In this section, we design the LTPI module based upon our proposed STPI to inpaint the whole video. This module can collect reference information from the long-distance frames. As a result, it may potentially obtain more available information for inpainting. Moreover, it is designed to rapidly deliver source information over frames and guarantee the temporal consistency of the inpainted video.

To implement LTPI, we firstly divide all the frames of a video into several GOPs, i.e., $\{G_1, G_2, ..., G_n\}$, where each GOP contains m frames. Then, we independently inpaint each G_i by using its inside reference information, which constructs the intra GOP inpainting. After all the GOPs are processed by the intra inpainting, we further fill the remained empty regions of the frames of G_i with the reference information offered by the other groups, which compose the inter GOP inpainting. Moreover, both the intra and inter GOP inpainting methods consist of Forward Propagation Inpainting (FPI) and Backward Propagation Inpainting (BPI), where FPI and BPI are implemented based upon our proposed STPI. To implement FPI, the given frame $F^{(j-1)}$ is selected as the reference frame for its next frame $F^{(j)}$. In contrast, to implement BPI, $F^{(j+1)}$ is used as the reference frame for its previous frame $F^{(j)}$. We design the GOP-based inpainting for LTPI so that we can avoid the propagation of inpainting errors over the whole video.

Intra GOP Inpainting

The intra GOP inpainting occurs inside each G_i, where FPI is firstly performed on the frames of G_i and then BPI is applied to them. To implement the intra FPI, our proposed STPI is used to fill the frames of G_i from the first to the last. In contrast, to implement the intra BPI, STPI is carried out from the last frame to the first frame of G_i. The implementation detail of the intra GOP inpainting is summarized in Algorithm 1.

Algorithm 1: Intra GOP inpainting

```
Input: Input video
Output: Inpainted GOPs $\{\hat{G}_1, \hat{G}_2, ..., \hat{G}_n\}$
Initialization: Dividing $S$ into GOPs $\{G_1, G_2, ..., G_n\}$
Processing frame $F_i^{(j)} \in G_i$:
for $i = 1 : n$ do
    Forward propagation-based inpainting (FPI):
    for $j = m : 2$ do
        Set $F_t = F_i^{(j)}$ and $F_r = F_i^{(j-1)}$;
        Update $F_i^{(j)}$: $F_i^{(j)} = STPI(F_t, F_r)$.
    end
    Backward propagation-based inpainting (BPI):
    for $j = 1 : m-1$ do
        Set $F_t = F_i^{(j)}$ and $F_r = F_i^{(j+1)}$;
        Update $F_i^{(j)}$: $F_i^{(j)} = STPI(F_i^{(j)}, F_i^{(j+1)})$.
    end
    Composing $\hat{G}_i$ with the updated $F_i^{(j)}$.
end
```

Inter GOP Inpainting

After the intra GOP inpainting is applied to all the groups, each frame of a GOP has been fully or partially filled by using the source information collected from its neighboring frame(s). After this inpainting, if there still exist empty areas in the frames of an inpainted GOP $\hat{G}_i$, the inter GOP inpainting will be applied to it to fill these areas. In the inter GOP inpainting, the source information is collected from the other GOPs and propagated gradually from the close groups to the distant ones. This strategy guarantees that the available information of the adjacent frames but different groups can be used with high priority. It also avoids the accumulation and propagation of inpainting error over the whole video.

The inter GOP inpainting is implemented in an iterative manner. In each iteration, FPI is firstly performed on the specific frames of the video, and then BPI is carried out. Both FPI and BPI are applied to the two adjacent GOPs, where one GOP offers reference frame for the inpainting of frames of another. More specifically, FPI starts from the last two GOPs, i.e., $(\hat{G}_{n-1}, \hat{G}_n)$, while BPI starts from the first two GOPs, i.e., $(\hat{G}_1, \hat{G}_2)$. To fill an incomplete GOP, when FPI is carried out, the last frame of $\hat{G}_{i-1}$ and all the frames of $\hat{G}_i$ are firstly gathered according to the temporal order. Then, STPI is sequentially performed on these frames, i.e., from the first one to the last one. To implement the inter BPI, all the frames of $\hat{G}_i$ and the first frame of $\hat{G}_{i+1}$ are collected firstly. Subsequently, STPI is carried out from

Algorithm 2: Inter GOP inpainting

```
Input: Intra GOP inpainted GOPs $\{\hat{G}_1, \hat{G}_2, ..., \hat{G}_n\}$
Output: Inpainted video
Processing $\{\hat{G}_1, \hat{G}_2, ..., \hat{G}_n\}$:
for $k = 1 : n-1$ do
    for $i = n : -1 : 1+k$ do
        if $\hat{G}_i$ is not completely inpainted then
            Update $\hat{G}_i$: $\hat{G}_i = FPI(\hat{G}_i, \hat{G}_{i-1})$
        else
            Skip
        end
    end
    for $i = 1 : n-k$ do
        if $\hat{G}_i$ is not completely inpainted then
            Update $\hat{G}_i$: $\hat{G}_i = BPI(\hat{G}_i, \hat{G}_{i+1})$
        else
            Skip
        end
    end
end
Composing video with the updated $\hat{G}_i$.
```

the last frame to the first one over the gathered frames. After one iteration is accomplished, the first two adjacent GOPs will be removed from the FPI procedure in the next iteration and the last two adjacent GOPs will also be removed from the BPI processing in the next iteration. As a result, with the increment of iteration, the participated GOPs in both FPI and BPI are progressively reduced, which effectively avoids some repeated FPI and BPI operations over the same adjacent GOPs. By applying FPI and BPI to all the adjacent GOPs, the spatio-temporal correlated information of all the frames can be propagated over the whole video, which offering reliable reference information for inpainting. Note that if $\hat{G}_i$ does not contain the incomplete frames, the inter GOP inpainting will be skipped. The detail to implement our proposed inter GOP inpainting is summarized in Algorithm 2.

We present results in Fig. 3.8 to demonstrate the effectiveness of our proposed inter GOP inpainting by making a comparison between the inpainting results with and without this operation. One can see from Fig. 3.8 that the use of the inter GOP inpainting is able to produce the pleasant results, with the propagation of the long-term source information throughout the video to reduce error accumulation.

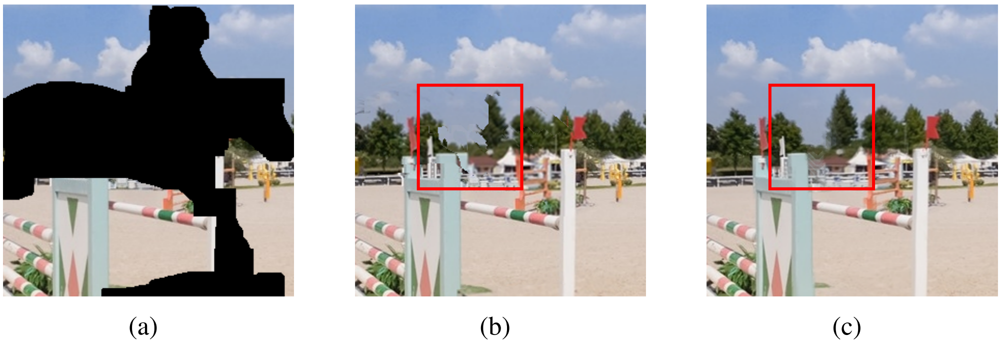


Fig. 3.8 Results for the verification of inter GOP inpainting. (a) Target frame; (b) without inter GOP inpainting; and (c) with inter GOP inpainting.

3.2.4 Final Refinement

Our proposed propagation-based inpainting methods, i.e., the STPI and LTPI modules, collect the source information from the other frames to fill the missing regions of a given frame. If all the reference frames cannot provide enough information to completely fill the empty areas of some specific frames, we will employ the GAN-based spatial inpainting [32] coupled with the propagation-based inpainting to complete them, which accordingly implements the final refinement. The former inpainting is used to fill all the missing areas of a single frame with a high visual quality and the latter one is employed to complete frames guaranteeing a high temporal consistency and a low complexity.

More specifically, for the frames which cannot be completely inpainted by the propagation-based methods, we firstly apply spatial inpainting to the frame with the largest remaining-empty region. After this frame is completely inpainted, the filled information is propagated to offer the source information for the other incomplete frames, where STPI is applied to these frames to guarantee a high temporal consistency. After that, if a frame still cannot be completely inpainted, the spatial inpainting will be performed to restore all the missing information. Note that the adoption of STPI avoids applying the spatial inpainting to all the incompletely inpainted frames, which accordingly achieves a low complexity.

We demonstrate the effectiveness of the final refinement by comparing the inpainting results obtained before and after its implementation. These results are presented in Fig. 3.9. According to the results shown in Fig. 3.9, the final refinement completely fills all the missing areas of the target frame.

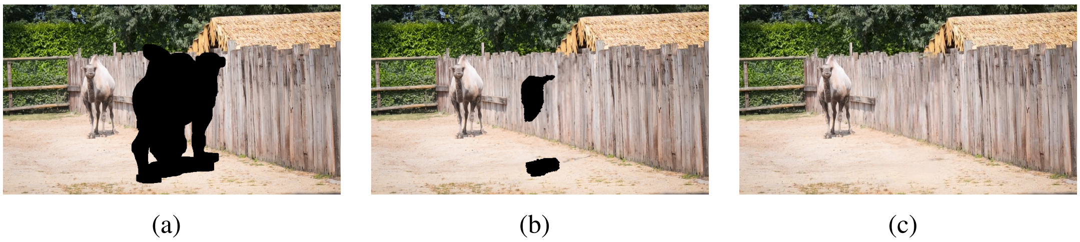


Fig. 3.9 Results for the verification of final refinement. (a) Target frame; (b) before final refinement; and (c) after final refinement.

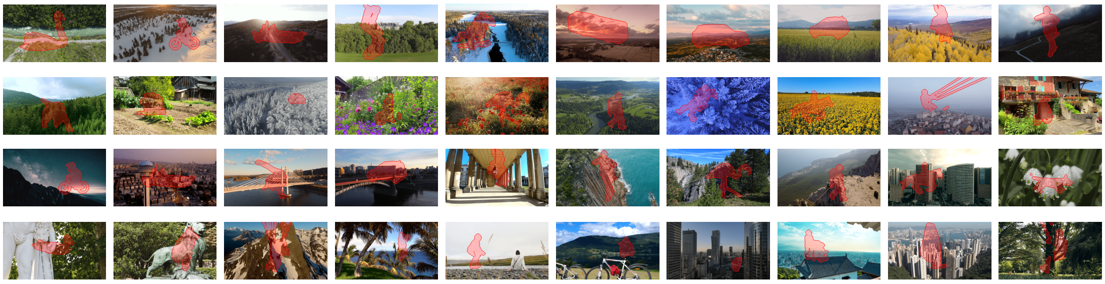


Fig. 3.10 The Landscape dataset with object-like masks for quantitative experiments. The top two rows are single-layer scenes and the bottom two rows are multi-layer scenes.

3.3 Experiments

3.3.1 Settings

Implementation Details

Our proposed method was developed based upon the Matlab platform with CPU except the depth estimation and the final refinement which were implemented on the Pytorch platform with GPU. In this work, we adopted Matlab 2020b and Pytorch 1.2.0 with Inter(R)-Core(TM) i7-9700k 3.60GHz CPU and NVIDIA GTX 2080Ti 11GB GPU in the experiments.

Datasets

We carried out the experiments on 40 landscape videos, as shown in Fig. 3.10, which were collected from the Internet, and 27 popular videos selected from the DAVIS dataset [10]. The resolutions of all the landscape videos are initially 720p (720×1280) and we resized them into the ones with three other resolutions, i.e., 240p (240×432), 480p (480×848) and 1080p (1080×1920). We applied the compared approaches and our method to these videos with different resolutions to make a compressive comparison. Meanwhile, the resolutions of all the adopted DAVIS videos are 480p in the experiment.

Table 3.1 Parameter determination for the shape-preserved weighting parameter η in the Single-Layer Alignment module. Various values of η are tested on the Landscapes dataset using object-like masks for the object-removal task. Performance is evaluated using PSNR and SSIM metrics, with higher values indicating better reconstruction quality. The best results for each metric are highlighted in bold.

η	0.5	1.0	1.5	2.0
PSNR (dB) ↑	35.81	**36.03**	35.72	35.68
SSIM ↑	0.9901	**0.9922**	0.9892	0.9880

3.3.2 Determination of Hyper-parameters

In our experiment, the hyperparameters for the single-layer alignment, multi-layer alignment, and progressive fusion modules were determined through a series of preliminary experiments. In practical applications, it is often impossible to quantitatively evaluate the inpainting performance for object removal due to the lack of GT data. To address this, we conducted experiments for hyperparameter determination by selecting random masks from the DAVIS dataset [10] and applying them to videos from the Landscapes dataset to generate artificially corrupted videos, paired with their corresponding GT. These experiments were performed on videos from the Landscapes dataset at a resolution of 480p.

For Single-Layer Alignment Module

To determine the shape-preserved weighting parameter η used in the single-layer alignment, we conducted experiments with different η on 20 videos from Lanscape dataset, to select the most applicable parameter η. The average quantitative results over these videos are given in Table 3.1. Based upon the results presented in Table 3.1, we choose $\eta = 1.0$ for our work due to the best performance offered by it.

For Multi-Layer Alignment Module

To determine the best combination of depth weight λ and super-pixel size n, we applied multi-layer alignment with various combinations of them to broken landscape videos. The other 20 videos from Lanscape dataset were adopted in this experiment and the classification threshold T was empirically specified as $T = 30$. The average quantitative results over these videos are given in Table 3.2. According to the results presented in Table 3.2, we selected $n = 80$ and $\lambda = 1.0$ for our method due to the better results achieved by using such a combination.

Table 3.2 Parameter determination for the super-pixel number n and the depth weight λ in the Multi-layer Alignment module. Various combinations of n and λ are tested on the Landscapes dataset using object-like masks for the object-removal task. Performance is evaluated using PSNR and SSIM metrics, with higher values indicating better reconstruction quality. The best results for each metric are highlighted in bold.

n	λ	PSNR (dB) ↑	SSIM ↑
40	0.5	35.54	0.9879
	1.0	35.75	0.9903
	1.5	35.68	0.9887
80	0.5	35.73	0.9896
	1.0	**35.81**	**0.9904**
	1.5	35.77	0.9902
120	0.5	35.41	0.9868
	1.0	35.64	0.9974
	1.5	35.28	0.9885

Table 3.3 Parameter determination for the maximum distance d_{max} in the Progressive Fusion module. Various values of d_{max} are tested on the Landscapes dataset using object-like masks for the object-removal task. Performance is evaluated using PSNR and SSIM metrics, with higher values indicating better reconstruction quality. The best results for each metric are highlighted in bold.

d_{max}	10	20	30	40	50
PSNR (dB) ↑	35.63	35.60	**35.92**	35.78	35.62
SSIM ↑	0.9905	0.9896	**0.9913**	0.9889	0.9883

For Progressive Fusion Module

To determine d_{max} in progressive fusion in STPI. We used different d_{max} in the experiment and chose the most applicable one to calculate the weighting factor ω for Eq. (3.12) used in the progressive fusion algorithm. The average quantitative results over all the broken landscape videos are given in Table 3.3. According to the results shown in Table 3.3, we finally adopt $d_{max} = 30$ in our work due to the superior performance.

3.3.3 Ablation Study for LTPI module

We conducted ablation experiments to verify the effectiveness of our proposed LTPI module, where the performance of intra GOP inpainting and inter GOP inpainting was

Table 3.4 Ablation study for the proposed LTPI module. Various combinations of intra GOP and inter GOP inpainting, along with forward and backward propagation, are evaluated on the Landscapes dataset. Performance is assessed using PSNR and SSIM metrics, where higher values indicate better reconstruction quality. The best results for each metric are highlighted in bold.

		PSNR (dB) ↑	SSIM ↑
Intra GOP Inpainting	FPI	27.89	0.9412
	BPI	27.43	0.9383
	FPI+BPI	33.56	0.9731
Intra GOP Inpainting +Inter GOP Inpainting	FPI	29.63	0.9557
	BPI	29.35	0.9541
	FPI+BPI	**35.92**	**0.9913**

verified. Meanwhile, the effectiveness of FPI and BPI was also verified. These experiments were conducted on the landscape videos with the 480p resolution. The verification results are given in Table 3.4. It was found from Table 3.4 that completing videos with both the intra GOP inpainting and the inter GOP inpainting demonstrate impressive results and adopting FPI and BPI can bring significant performance gain.

3.3.4 Comparisons

Quantitative Results

In quantitative comparison, we evaluated the performance of our proposed method by comparing it with the SOTA methods, including two propagation-based methods, i.e., TCCDV [3] and FGVC [6], and four deep learning-based methods, i.e., CPNet [95], OPN [5], Fusing fine-grained information in transFormers (FuseFormer) [59] and IIVI [7]. Note that TCCDV [3] was designed on the Matlab platform with CPU, while FGVC [6] was developed based on the Pytorch platform with both CPU and GPU. The four deep learning-based methods were implemented on the Pytorch platform with CPU. The experiment was conducted on two datasets. The first one is the popular video segmentation dataset DAVIS and the second one is our collected dataset Landscapes that is composed by fourty videos without moving objects. We adopted three quantitative metrics, including PSNR (dB), SSIM, and LPIPS, to evaluate the efficiency of different methods.

Firstly, as previous work [6, 8, 9] did, we adopted 50 video clips from DAVIS dataset to evaluate quantitative performance and time efficiency. The resolutions of these videos are 240p (240×432). We generated the stationary square masks and the temporally-varied

Table 3.5 Quantitative comparison on the Densely Annotated VIdeo Segmentation [10] dataset for video inpainting. Two types of masks are used for evaluation: square masks and object-like masks. The masks are selected from background regions without moving targets to facilitate the evaluation of object-removal performance. We employ PSNR, SSIM, and LPIPS for quantitative assessment. ↑ indicates higher is better while ↓ indicates lower is better. The best results for each metric are highlighted in bold.

Methods	Square			Object			Runtime
	PSNR(dB) ↑	SSIM↑	LPIPS ↓	PSNR(dB) ↑	SSIM ↑	LPIPS ↓	(s/frame)
TCCDV [3]	31.53	0.9654	0.034	30.14	0.9509	0.043	4.03
CPNet [95]	30.43	0.9468	0.045	29.65	0.9331	0.051	0.40
OPN [5]	30.58	0.9492	0.047	29.57	0.9364	0.053	0.96
FGVC [6]	32.45	0.9709	0.028	31.16	0.9611	0.035	2.36
IIVI [7]	31.69	0.9628	0.031	31.22	0.958	0.037	110.15
E2FGVI [8]	33.78	0.9809	0.026	33.07	0.9777	0.028	0.16
FGT [9]	33.86	0.9831	0.024	32.89	0.9693	0.031	1.89
SLTPVI (Ours)	**34.04**	**0.9865**	**0.021**	**33.15**	**0.9806**	**0.025**	0.51

irregular masks for video restoration scenario and object removal scenario, respectively. The quantitative evaluation results were given in Table. 3.5. It can be seen from Table. 3.5 that our method outperforms the state-of-the-arts on all the three quantitative metrics. Meanwhile, the runtime of different approaches are also presented in Table. 3.5. Although our method was developed mostly based on CPU, its average processing time is comparable to that of the deep learning-based methods which were implemented based on GPU, which proves the time efficiency of our proposed approach.

Secondly, to verify the robustness of different methods, we applied them to the videos of our Landscapes dataset, where each video is transferred into three versions with three corresponding resolutions, i.e. 480p, 720p, and 1080p. We used the masks offered by DAVIS dataset to remove the content of video and fill the empty regions. Our Landscape dataset can provide ground-truth to evaluate object removal with object-like masks. The corresponding results were offered in Table 3.6. OPN [5] and E2FGVI [8] cannot process the 1080p videos due to the limited GPU memory. One can see from Table 3.6 that our method achieves the best performance in different resolution scenarios with the given marks. This demonstrates its robustness to video resolution and removed content.

Qualitative Results

We carried out qualitative comparison experiments on DAVIS datset. These videos are also used in TCCDV [3] for the quantitative comparison, where the user-specified object

Table 3.6 Robustness verification of different methods for object removal tasks at various resolutions on the Landscapes dataset. The comparison is conducted across three high resolutions: 480p, 720p, and 1080p. PSNR and SSIM are used for quantitative assessment. The best results for each metric are highlighted in bold.

Methods	PSNR(dB) ↑ / SSIM ↑		
	480p	720p	1080p
TCCDV [3]	29.27/0.9644	30.30/0.9719	30.01/0.9673
CPNet [95]	29.21/0.9566	29.89/0.9660	30.68/0.9728
OPN [5]	28.32/0.9589	29.62/0.9657	-/-
FGVC [6]	33.24/0.9871	32.73/0.9875	32.33/0.9832
IIVI [7]	32.12/0.9782	31.84/0.9773	30.77/0.9745
E2FGVI [8]	33.40/0.9832	33.12/0.9827	33.01/0.9821
FGT [9]	35.37/0.9901	-/-	-/-
SLTPVI (Ours)	**35.92/0.9913**	**35.58/0.9907**	**35.23/0.9896**

masks are given, and the videos contain both the single-layer scenes and multi-layer scenes. We demonstrate the inpainting performance of our method by performing it on these videos and make the qualitative comparison with state-of-the-art methods.

Firstly, we present some visual results in Fig. 3.11 to compare our method with the other methods. It is found from Fig. 3.11 that our method produces more pleasant results which contain smoother edges and clearer textures. However, the other methods, especially OPN [5], often generate blurred results and distorted semantic objects. All the results of our proposed method can be found in the website[1].

Secondly, besides visual results, the temporal coherence evaluation adopted in TCCDV [3] is used to compare the temporal consistency of the results obtained by using different methods. In this comparison, a slice of successive frames is selected and the spatio-temporal profile (the yellow line highlighted in frames) is given. We offer the temporal coherence results in Fig. 3.12. One can see from Fig. 3.12 that our result maintains the long-term temporal consistency and accordingly achieves better temporal coherence.

[1] https://drive.google.com/drive/folders/1Qcsn0cy36xRVcgKTLbeFxNpBPLBB-RIa

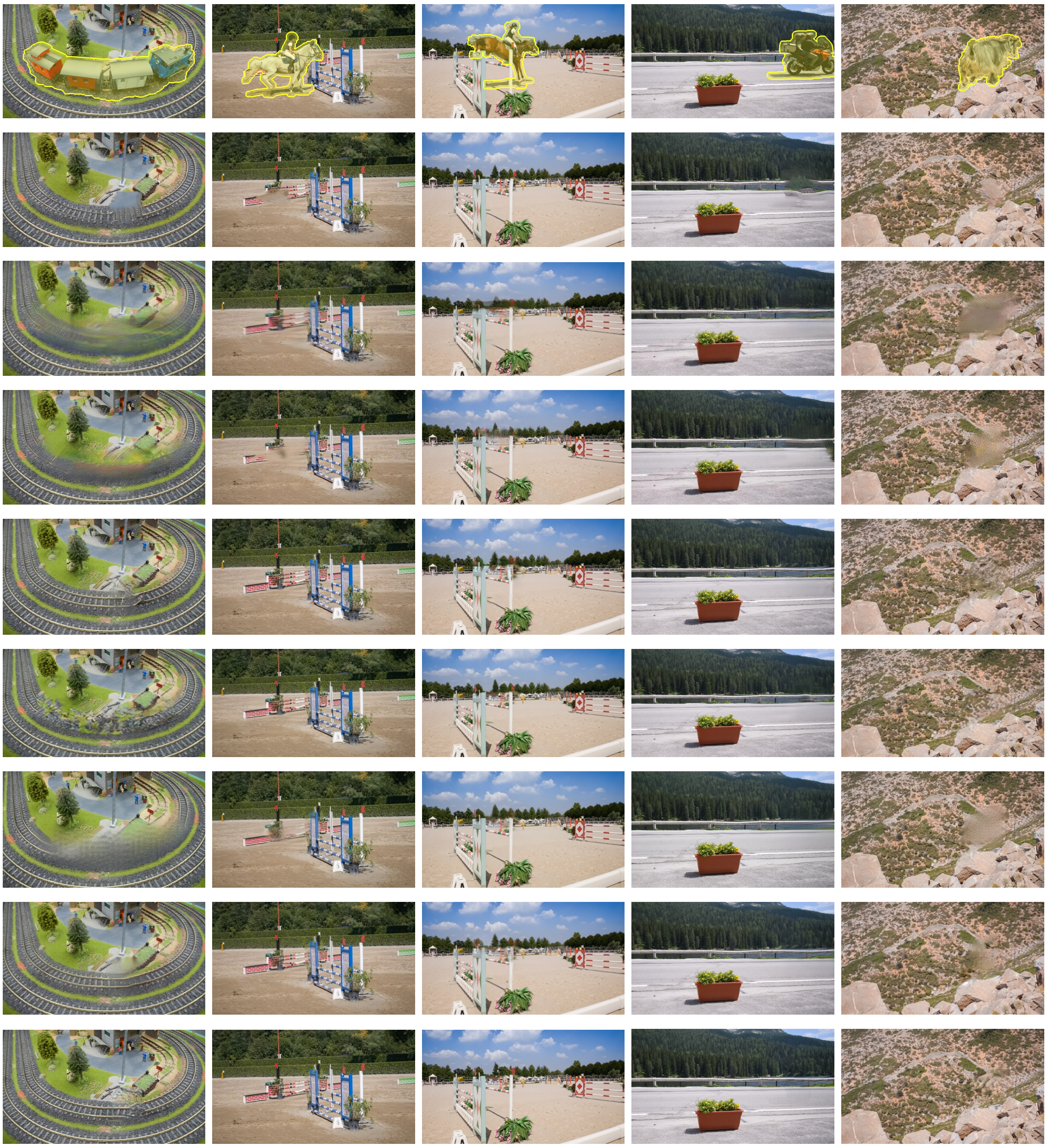


Fig. 3.11 Qualitative comparison of inpainting results for some videos from DAVIS dataset. From left to right: *Train*, *Horsejump-Low*, *Horsejump-High*, *Motorbike* and *Goat*. From top to bottom: Mask, TCCDV [3], CPNet [4], OPN [5], FGVC [6], IIVI [7], E2FGVI [8], FGT [9] and ours.

3.4 Summary

In this paper, we proposed a short-long-term propagation-based method for the object-removal task. The proposed method integrates two inpainting modules, namely the STPI and

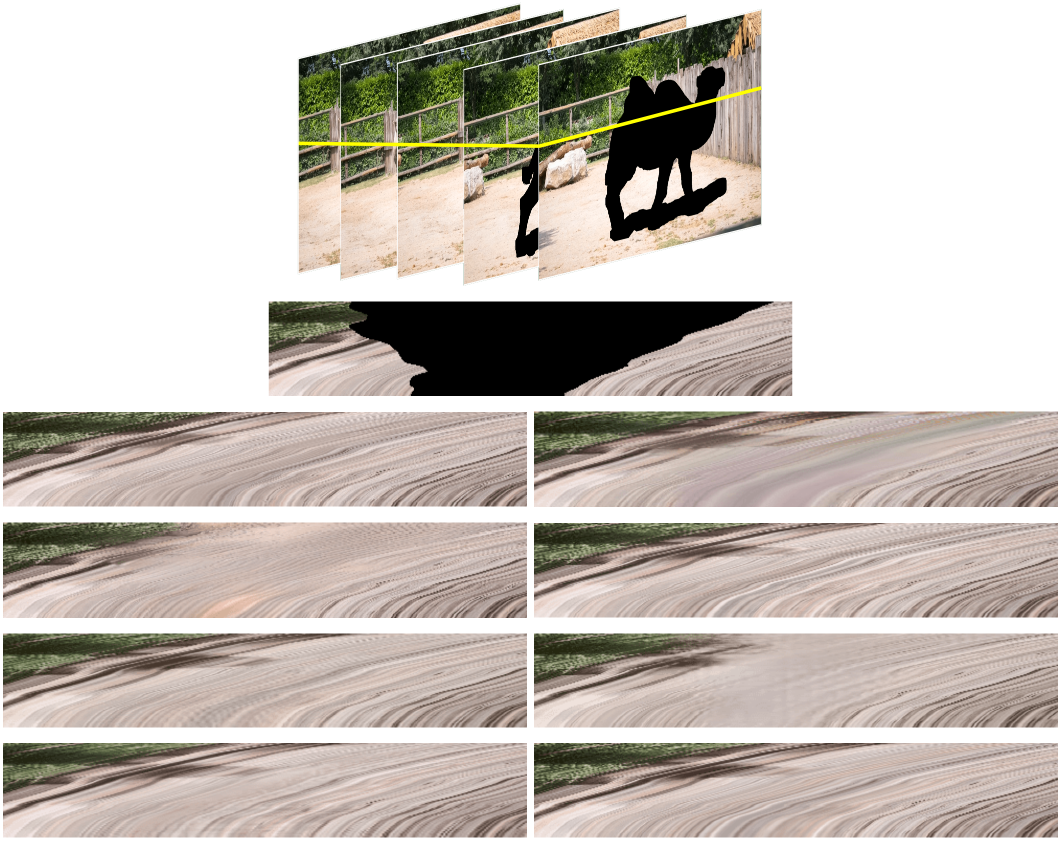


Fig. 3.12 Comparison of temporal coherence. From top to bottom: Temporal slice, Mask, TCCDV [3], CPNet [4], OPN [5], FGVC [6], IIVI [7], E2FGVI [8], FGT [9] and ours.

LTPI modules. The STPI module is designed to inpaint a single frame by leveraging reference information from local adjacent frames, while the LTPI module extends the process to the entire video by progressively applying STPI across all frames. Specifically, STPI comprises three components: source-region acquisition, illumination adaptation, and progressive fusion. Meanwhile, LTPI includes intra-GOP and inter-GOP inpainting, followed by final fusion. Together, these modules propagate correlated spatio-temporal information from one frame to another, ensuring high temporal consistency in the inpainted video.

We conducted experiments on the DAVIS and Landscapes datasets to evaluate the performance of the proposed method for the object-removal task. Comparative analyses were carried out against several SOTA methods [3, 95, 5, 6, 59, 7, 9]. The experimental results, including both qualitative and quantitative evaluations, demonstrate that our method outperforms current state-of-the-art approaches in terms of visual quality and temporal coherence.

While the proposed SLTPVI model effectively and efficiently addresses the object-removal task, producing plausible and visually consistent content for missing regions, it encounters challenges when applied to video restoration tasks involving random missing regions. In scenarios where missing regions span both moving objects and backgrounds, particularly in complex scenes, SLTPVI often generates artifacts and struggles to maintain temporal coherence. To address these limitations, the next chapter introduces a depth-guided deep video inpainting network designed to handle such challenges.

4

Depth-guided Deep Video Inpainting

4.1 Introduction

In the previous chapter, we presented a short-long-term propagation-based video inpainting approach designed for object removal. While this method demonstrated impressive performance in producing inpainted videos for clean backgrounds or street scenes, it exhibits limitations when applied to more complex tasks, such as video restoration and subtitle removal where missing region spans both foregrounds and backgrounds. These scenes often require the reconstruction of content with multiple layers. In this chapter, we will introduce a depth-guided deep video inpainting (DGDVI).

Although numerous video inpainting approaches [6, 58, 59, 8, 9] have been proposed to address complex video completion scenes in the past, there are still challenges in the design of an effective scheme. Notably, it is still a significant challenge to compose the missing regions crossing different depth layers, especially those from dynamic foregrounds and backgrounds. Due to the lack of cues to identify layers for missing regions, foreground and background reference information aliasing frequently happens when information is propagated from reference region to target region for content construction, resulting in blurred edges and details. This aliasing also induces spatial and temporal incoherence between composed frames, generating low-quality inpainted videos.

Over the past years, some methods [6, 9, 51] have been proposed to generate spatial and temporal coherent results by introducing optical flow as guidance. These methods estimated optical flow for missing regions and propagated reference information from frame to frame guided by the flow to construct contents for missing regions. The flow-based methods can be implemented in either pixel domain or feature domain, via content propagation or feature propagation throughout the video for completion. Accurate optical flow is crucial for information propagation in these methods. However, the flow often changes dramatically over a long duration, which makes accurate flow estimation for all missing regions over the whole video difficult when the long-range cues are needed. Meanwhile, estimation error often happens and is propagated during flow estimation, especially on the regions crossing

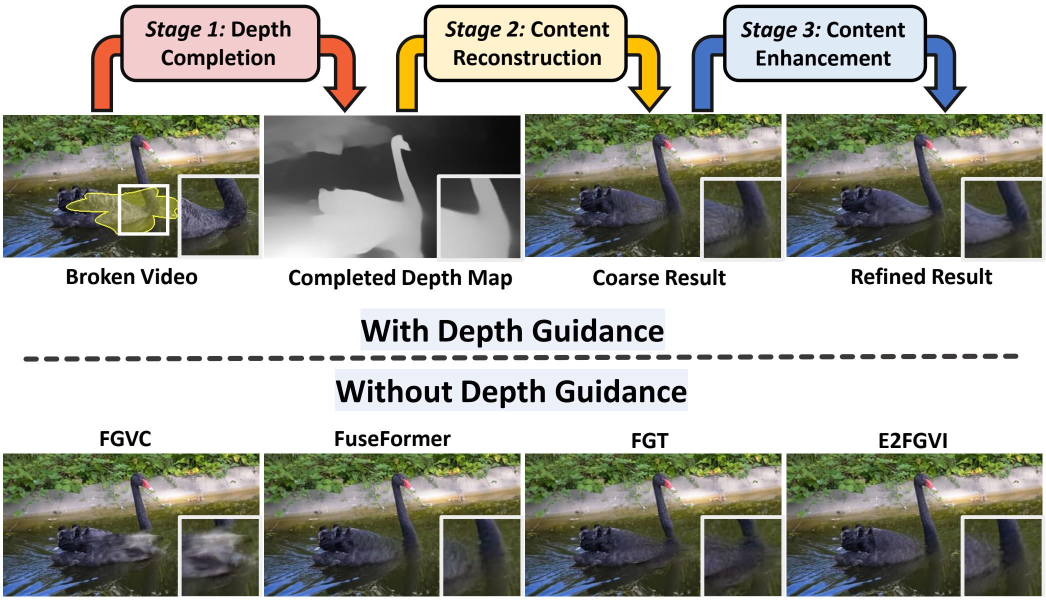


Fig. 4.1 Pipeline of our proposed DGDVI approach. Our approach used the predicted depth map to guide the content reconstruction for missing regions. Compared with the approaches without depth guidance, our method can generate more reliable contents for missing regions that cross foreground and background.

foreground and background layers. The existence of this error will result in information propagation error, thus limiting inpainting efficiency.

To solve the above problem, we adopted depth rather than optical flow to guide the information propagation for video inpainting. Compared with flow, depth is temporally invariant over the whole video, which makes it much easier to be predicted for missing regions. In addition, using depth information can effectively distinguish the foreground and background for the video frame. This indicates that adopting it to guide information propagation may potentially solve the reference information aliasing problem.

In a previous study [131], depth was used to implement the warping of reference region to the target broken region, where the reference region was offered by an external image sharing scene contents with the target image. The warped reference region then offered the scene-consistent information for the completion of the broken region. Additionally, in [14], the depth information acquired from Lidar was used to guide the fusion of multiple source videos, to generate an inpainted clear video without undesired traffic agents. In contrast to previous work, we aimed to use depth to guide the information propagation for the construction of contents for the broken video. We did not use it to either align video frames or fuse videos to implement inpainting.

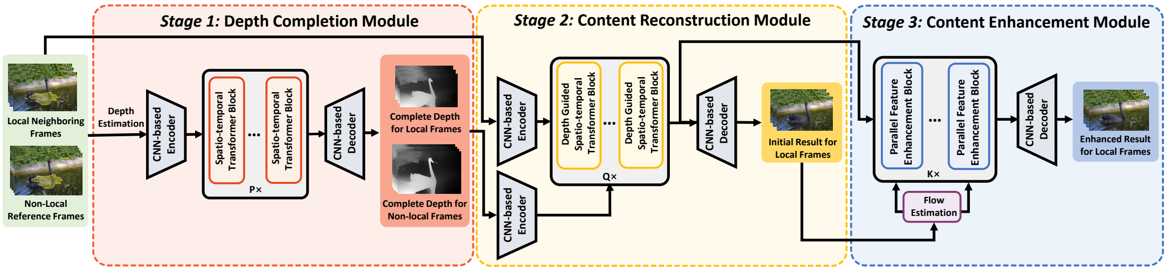

Stage 1: Depth Completion Module
Stage 2: Content Reconstruction Module
Stage 3: Content Enhancement Module
Local Neighboring Frames
Non-Local Reference Frames
Depth Estimation
CNN-based Encoder
Spatio-temporal Transformer Block
...
Spatio-temporal Transformer Block
P×
CNN-based Decoder
Complete Depth for Local Frames
Complete Depth for Non-local Frames
CNN-based Encoder
CNN-based Encoder
Depth Guided Spatio-temporal Transformer Block
...
Depth Guided Spatio-temporal Transformer Block
Q×
CNN-based Decoder
Initial Result for Local Frames
Parallel Feature Enhancement Block
...
Parallel Feature Enhancement Block
K×
Flow Estimation
CNN-based Decoder
Enhanced Result for Local Frames

Fig. 4.2 Framework of the proposed DGDVI.

To achieve effective video inpainting, we proposed a depth-guided method in this work and implemented it in three stages, including the depth completion, content reconstruction, and content enhancement, as illustrated in Fig. 4.1. We designed three corresponding modules for these stages, to predict the depth of video, compose content for the missing region, and enhance the composed content, respectively. These modules were subsequently used to construct our DGDVI, as illustrated Fig. 4.2. Our proposed method aims to achieve high robustness and performance for the challenging inpainting scenes, especially for the filling of region crossing different depth layers. Our contributions are summarized as follows:

- We proposed a video inpainting method with the guidance of depth. The depth information was adopted to guide the information propagation over the video, composing reasonable and reliable results, especially for the completion of multi-layered region.
- We constructed a depth completion module to predict the completed depth for the broken video by using both local and non-local spatio-temporal reference information.
- We designed a content reconstruction module to generate contents for missing regions with the guidance of depth, solving the content aliasing problem.
- We developed a content enhancement module with our proposed parallel feature enhancement network to enhance the temporal coherence and texture quality for the video, guaranteeing to achieve high inpainting quality.

4.2 Methodology

4.2.1 Overview

Our proposed DGDVI method is implemented in three stages, as shown in Fig. 4.2, including depth completion, content reconstruction and content enhancement, to complete broken videos, especially the one with multi-layered contents. Three corresponding modules are designed for these stages and are jointly optimized for the implementation of our proposed model.

In our work, given a broken video that contains N frames $\{X_1, X_2, \ldots, X_N\}$, where $X_i \in \mathbb{R}^{H \times W \times 3}$, we firstly divide the frames into several local frame groups. Each local frame group, denoted as $\mathbf{X}_l$, composed by N_l frames that are obtained by performing a temporal window on the video to select frames. Meanwhile, we construct one non-local frame group, denoted as $\mathbf{X}_{nl}$, that consists of N_{nl} frames obtained by uniformly sampling the video frames with a given step-size. With $\mathbf{X}_l$ and $\mathbf{X}_{nl}$, we compose $\mathbf{X}_{in} = \{\mathbf{X}_l, \mathbf{X}_{nl}\}$.

Then, in the depth completion and content reconstruction stages, we use $\mathbf{X}_{in}$ to produce a rough inpainting result for $\mathbf{X}_l$. Note that $\mathbf{X}_{in}$ consists of local and non-local frames. The employment of non-local frames to produce the inpainted local frames aims at introducing long-range spatio-temporal context to achieve high quality. Finally, we just use the frames of $\mathbf{X}_l$ to enhance the local temporal coherence for it and obtain a refined result in the content enhancement stage.

4.2.2 Depth Completion

The structure, shape and contour cues can be clearly indicated in depth, which makes it be potentially used to enhance the quality of reconstructed contents. In this work, we design the depth completion module to predict depth for damaged video. The predicted depth is then used to guide the content reconstruction. This module is constructed based on the spatio-temporal Transformer with multi-head self-attention and can be used to obtain the local and non-local depth dependencies to generate completed depth.

Given $\mathbf{X}_{in} = \{\mathbf{X}_l, \mathbf{X}_{nl}\} \in \mathbb{R}^{(N_l+N_{nl}) \times H \times W \times 3}$, we use a pre-trained depth estimation network [122] to obtain depth information for the available image regions but leave the other regions of depth empty. Assuming that $\mathbf{D}_{in} \in \mathbb{R}^{(N_l+N_{nl}) \times H \times W}$ is composed of the incomplete depth of $\mathbf{X}_{in}$, the proposed depth completion module is designed to generate complete depth for $\mathbf{X}_{in}$ based on $\mathbf{D}_{in}$.

Inspired by [58, 59], we construct the depth completion module based on the CNN-Transformer hybrid architecture that enables the module to generate accurate and temporally consistent depth information. In addition, the depth completion is implemented in three steps, including feature extraction, feature propagation and depth construction.

Specifically, the features of $\mathbf{D}_{in}$ are firstly extracted by a CNN-based context encoder and converted into token embeddings. Then, the token embeddings are fed into a stack of spatio-temporal Transformer blocks to complete the feature propagation via token updating.

Finally, the updated tokens are converted back into features by a token-to-patch module and a CNN decoder is applied to produce the predicted depth information $\mathbf{D}_c \in \mathbb{R}^{(N_l+N_{nl})\times H\times W}$.

Feature Extraction

We firstly concatenate $\mathbf{D}_{in}$ with the corresponding inpainting mask and then feed them into a CNN-based encoder which consists of five convolutional layers to obtain the features $\mathbf{E}_{dep}$ where the channel number is c_{dep} and the size of feature map is $H/4 \times W/4$. Then, the features $\mathbf{E}_{dep}$ are converted into tokens so that they can be fed into the consequent Transformer blocks. In this work, the Soft Split (SS) [59] operation is applied to split $\mathbf{E}_{dep}$ into overlapped patch embeddings to form the tokens $\mathbf{Z}^0_{dep}$ as

$$\mathbf{Z}^0_{dep} = \mathrm{SS}(\mathbf{E}_{dep}). \tag{4.1}$$

With the soft split operation, the temporally-correlated features extracted from depth are converted into overlapped token embeddings so that we can build up the spatio-temporal correlation among tokens for the updating of them during feature propagation.

Feature Propagation

We adopt feature propagation [58, 59, 8] to propagate the features of depth from available regions to missing regions in the incomplete depth so that we can complete depth. The spatio-temporal Transformer blocks are employed to facilitate this propagation according to the short-long term dependencies and contextual cues in both feature and temporal domains.

To guide the feature propagation, the MSA [65, 96] is adopted in the Spatio-Temporal Transformer Block (STTB) [58, 59] to construct the spatio-temporal dependencies within tokens. To implement the MSA mechanism, the input tokens are firstly transformed into the query, key and value vectors, and then are split into multiple heads to extract more diverse and expressive representations than only using single head. For each head, its attention map is obtained by calculating the attention scores between its query and key vectors, capturing multiple relationships between multiple tokens. With the obtained attention map, we can assign appropriate attention weights to relevant values so as to update the tokens in feature and temporal domains.

Assuming that there are k heads in MSA, the updated value $\hat{\mathbf{V}}_k$ for each head is calculated as

$$\hat{\mathbf{V}}_k = \mathrm{MSA}(\mathbf{Q}_k, \mathbf{K}_k, \mathbf{V}_k) = \frac{softmax(\mathbf{Q}_k\mathbf{K}_k^T)}{\sqrt{d_k}}\mathbf{V}_k, \tag{4.2}$$

where $\mathbf{Q}_k$, $\mathbf{K}_k$, and $\mathbf{V}_k$ are the query, key and value vectors for each head, respectively, and d_k is the feature dimension of $\mathbf{Q}_k$ and adopted as a scaling factor. With the application of MSA,

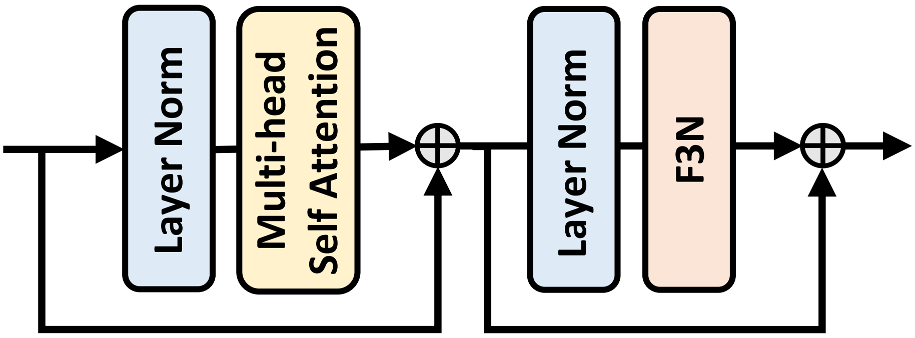


Fig. 4.3 Architecture of STTB.

the module can update the tokens for broken regions during feature propagation, guaranteeing to generate accurate depth information.

Based on MSA, we construct STTB as illustrated in Fig. 4.3 and it is implemented as

$$\begin{aligned}\mathbf{Z}'^{n}_{dep} &= \mathrm{MSA}(\mathrm{LN}(\mathbf{Z}^{n-1}_{dep})) + \mathbf{Z}^{n-1}_{dep} \\ \mathbf{Z}^{n}_{dep} &= \mathrm{F3N}(\mathrm{LN}(\mathbf{Z}'^{n}_{dep})) + \mathbf{Z}'^{n}_{dep},\end{aligned} \tag{4.3}$$

where $\mathbf{Z}^{n-1}_{dep}$ denote the input token embeddings outputted from the $(n-1)^{th}$ Transformer block, $\mathbf{Z}^{n}_{dep}$ represent the output of n^{th} Transformer block, LN denotes the layer normalization [132], and Fusion Feed-Forward Network (F3N) [59] consists of a Soft Composition (SC) [59] and a soft split operation. Note that F3N is adopted in our work to build up the interaction of overlapped token embeddings for effective feature propagation. We stack P spatio-temporal Transformer blocks and use them to implement feature propagation, enabling the module to effectively combine the cues collected from local and non-local depth to generate the complete depth.

Depth Construction

After feature propagation, the token embeddings are accordingly updated. In order to obtain the complete depth, the updated tokens have to be converted back into features. To achieve this goal, the soft composition is applied to convert tokens into the overlapped feature patches. These patches are then used to form the complete features as

$$\hat{\mathbf{E}}_{dep} = \mathrm{SC}(\mathbf{Z}^{N_1}_{dep}), \tag{4.4}$$

where $\mathbf{Z}^{P}_{dep}$ denotes the output of the P^{th} Transformer block and $\hat{\mathbf{E}}_{dep}$ is the composed complete features of depth. Then, the features $\hat{\mathbf{E}}_{dep}$ are decoded by a CNN-based decoder consisting of four convolutional layers to generate the predicted depth $\mathbf{D}_c$ which has the same resolution as the input frame.

4.2.3 Content Reconstruction

The content reconstruction module is designed to generate contents for the broken foreground and background of missing regions. Note that the depth indicates both the contour and content layer information for a picture. Given a picture whose missing content crosses foreground and background, introducing depth as guidance to reconstruct the missing content may produce result with clear shape and structure. In this work, we construct the content reconstruction module based on the depth-guided spatio-temporal Transformer and the proposed multi-head mutual-self-attention. With this module, we can combine the spatial and temporal dependencies of frames with the guidance of depth to produce reasonable and reliable content for the target region.

Our proposed content reconstruction module is also developed based on the CNN-Transformer architecture and consists of feature extraction, depth-guided feature propagation and content composition. Specifically, given broken frames $\mathbf{X}_{in}$ and the corresponding complete depth maps $\mathbf{D}_c$, the lower-resolution features of content and depth are firstly extracted from the broken video frames and the predicted depth via two CNN-based encoders, respectively. Then, these features are converted into tokens and fed into the depth-guided STTB in which the tokens of content are updated through the depth-guided feature propagation. The updated tokens are converted into features and these features are finally used to reconstruct frames $\hat{\mathbf{X}}_{rec}$ via a CNN-based decoder.

Feature Extraction

Firstly, a context encoder[44] which consists of nine convolutional layers takes in the incomplete frames $\mathbf{X}_{in}$ and produces $1/4$ sized feature with C_{cn} channels. And the corresponding predicted depth $\mathbf{D}_c$ is feed into a CNN-based encoder with four convolutional layers to obtain $1/4$ sized feature maps $\mathbf{E}_{dp}$ with C_{dep} channels. Then, we apply the soft split to convert $\mathbf{E}_{cn}$ and $\mathbf{E}_{dp}$ into overlapped token embeddings as

$$\begin{aligned} \mathbf{Z}_{cn}^0 &= \mathrm{SS}(\mathbf{E}_{cn}) \\ \mathbf{Z}_{dp}^0 &= \mathrm{SS}(\mathbf{E}_{dp}), \end{aligned} \tag{4.5}$$

where $\mathbf{Z}_{cn}^0$ and $\mathbf{Z}_{dp}^0$ represent the embedded tokens for $\mathbf{E}_{cn}$ and $\mathbf{E}_{dp}$, respectively.

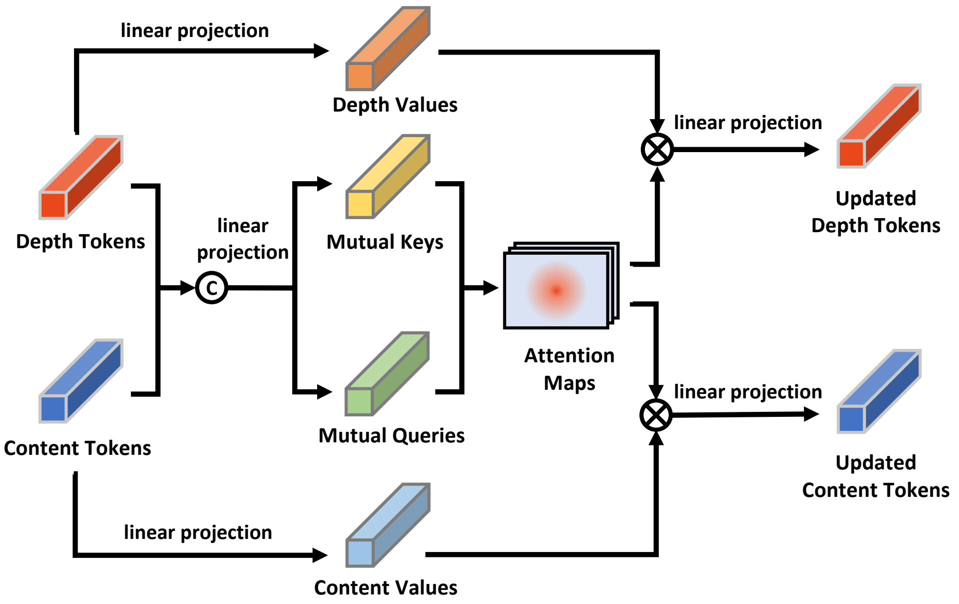


Fig. 4.4 Architecture of MMSA.

Depth-guided Feature Propagation

Since the depth tokens contain enough depth information, we use them to guide the construction of spatio-temporal relationship between content tokens during feature propagation, which makes the model learn how to utilize depth index to assign appropriate attention weights for token updating. To achieve the above goal, we construct a Depth-Guided Spatio-Temporal Transformer Block (DGSTTB) to facilitate the feature propagation. In addition, Mutual Multi-head Self Attention (MMSA) is adopted in DGSTTB to make the content tokens update with the interaction of depth tokens.

To implement MMSA in DGSTTB, the depth tokens just participate in the assignment of attention weights but are not used to determine value vectors. Hence, the module can focus on the reference regions with similar depth to the target region and spread these cues to the target region so that it can effectively update token embeddings for missing regions to achieve better content reconstruction. Different from MSA, we use mutual query $\mathbf{Q}_{mul}$ and mutual key $\mathbf{K}_{mul}$ to obtain the attention map as well as the content value $\mathbf{V}_{cn}$ and the depth value $\mathbf{V}_{dp}$ for value updating in MMSA, as illustrated in Fig. 4.4, where $\mathbf{V}_{cn}$ and $\mathbf{V}_{dp}$ are independent. More specifically, we concatenate $\mathbf{Z}_{cn}$ with $\mathbf{Z}_{dp}$ together and use a linear projection layer f_{kq} to convert them into the mutual query vector $\mathbf{Q}_{mul}$ and mutual key vector $\mathbf{K}_{mul}$, respectively, i.e.,

$$\{\mathbf{K}^{n}_{mul}, \mathbf{Q}^{n}_{mul}\} = f_{kq}(\text{Concat}(\mathbf{Z}^{n-1}_{cn}, \mathbf{Z}^{n-1}_{dp})), \tag{4.6}$$

where $\mathbf{Z}_{cn}^{n-1}$ and $\mathbf{Z}_{dep}^{n-1}$ represent the output content and depth token embeddings of the $(n-1)^{th}$ DGSTTB and $\mathbf{K}_{mul}^{n}$ and $\mathbf{Q}_{mul}^{n}$ are mutual key and query vectors of the n^{th} DGSTTB. In addition, $\mathbf{Z}_{cn}$ and $\mathbf{Z}_{dp}$ are independently converted into the content and depth values

$$\begin{aligned}\mathbf{V}_{cn}^{n} &= f_{vc}(\mathbf{Z}_{cn}^{n-1})\\ \mathbf{V}_{dp}^{n} &= f_{vd}(\mathbf{Z}_{dp}^{n-1}),\end{aligned} \tag{4.7}$$

where f_{vc} and f_{vd} are the linear projection layers used to generate the content and depth values, respectively. In order to capture various relationships between tokens, $\mathbf{Q}_{mul}$, $\mathbf{K}_{mul}$, $\mathbf{V}_{cn}$ and $\mathbf{V}_{dp}$ are then split into multiple heads. For each head, we generate its corresponding attention map by using mutual query and key. Then, we assign the weights for token updating according to the attention map.

Assuming that there are k heads in MMSA, the updated content and depth values for each head, denoted as $\hat{\mathbf{V}}_{cn,k}$ and $\hat{\mathbf{V}}_{dp,k}$, are obtained as

$$\begin{aligned}\{\hat{\mathbf{V}}_{cn,k},\hat{\mathbf{V}}_{dp,k}\} =& \text{ MMSA}(\mathbf{Q}_{mul,k},\mathbf{K}_{mul,k},\mathbf{V}_{cn,k},\mathbf{V}_{dp,k})\\ =& \frac{softmax(\mathbf{Q}_{mul,k}\mathbf{K}_{mul,k}^{T})}{\sqrt{d_k}}\{\mathbf{V}_{cn,k},\mathbf{V}_{dp,k}\},\end{aligned} \tag{4.8}$$

where $\mathbf{Q}_{mul,k}$, $\mathbf{K}_{mul,k}$, $\mathbf{V}_{cn,k}$, $\mathbf{V}_{dp,k}$ are the mutual query, mutual key, content value and depth value for each head, respectively, and d_k is the feature dimension of $\mathbf{Q}_{mul,k}$.

Based on MMSA, DGSTTB is implemented as

$$\begin{aligned}\{\mathbf{Z}_{cn}^{\prime n},\mathbf{Z}_{dp}^{\prime n}\} =& \text{MMSA}(\{\text{LN}(\mathbf{Z}_{cn}^{n-1}),\text{LN}(\mathbf{Z}_{dp}^{n-1})\}) + \{\mathbf{Z}_{cn}^{n-1},\mathbf{Z}_{dp}^{n-1}\}\\ \mathbf{Z}_{cn}^{n} =& \text{F3N}(\text{LN}(\mathbf{Z}_{c}^{\prime n})) + \mathbf{Z}_{cn}^{\prime n}\\ \mathbf{Z}_{dp}^{n} =& \text{F3N}(\text{LN}(\mathbf{Z}_{dp}^{\prime n})) + \mathbf{Z}_{dp}^{\prime n},\end{aligned} \tag{4.9}$$

where $\mathbf{Z}_{dp}^{n-1}$ denotes the output token embeddings of the $(n-1)^{th}$ DGSTTB and $\mathbf{Z}_{dp}^{n}$ represents the output of the n^{th} transformer block. We stack Q DGSTTBs and use them to facilitate the depth-guided feature propagation, enabling the model to effectively update the tokens for broken regions to produce reliable contents for missing regions.

Initial Frame Composition

After the feature propagation, the content tokens are accordingly updated. In order to reconstruct frames, the overlapped token embeddings for contents are converted into features by the CNN-based decoder with the SC operation as

$$\hat{\mathbf{E}}_{rec} = \text{SC}(\mathbf{Z}_{cn}^{N_2}), \tag{4.10}$$

where $\mathbf{Z}_{cn}^{Q}$ denotes the content tokens obtained from the Q^{th} DGSTTB and $\hat{\mathbf{E}}_{rec}$ is the composed features obtained by using SC. Note that the features $\hat{\mathbf{E}}_{rec}$ only contain the features of local frames and the features of non-local frames are discarded. Then we apply the CNN-based decoder which consists of four convolutional layers to progressively upsample $\hat{\mathbf{E}}_{rec}$ and generate initial results for local frames. After that, the composed local frames $\hat{\mathbf{X}}_{rec}$ and the composed features of local frames $\hat{\mathbf{E}}_{rec}$ are fed into the content enhancement module to generate the final inpainting results.

4.2.4 Content Enhancement

After composing the frames for the broken video, we further improve the video quality by introducing optical flow as the guidance to enhance the temporal coherence of neighboring frames. The content enhancement module is accordingly designed to enhance the local temporal consistency and the texture quality for the video. This module is constructed based on a parallel feature fusion network with the flow-guided deformable warping. With this module, we can strengthen the local temporal coherence of the video to enhance the visual consistency and improve both structure and texture details for the final inpainting result.

In this work, we develop a parallel content enhancement module based on the flow-guided deformable convolution [133] to facilitate the content enhancement that consists of flow estimation, feature enhancement and final frame reconstruction. Specifically, given the composed local frames $\hat{\mathbf{X}}_{rec}$, we predict forward and backward flows for them with a flow estimation network. Then, with the features $\hat{\mathbf{E}}_{rec}$ obtained from content reconstruction module, we implement flow-guided deformable warping to simultaneously warp the features of the neighboring frames to the target frame and fuse the warped features for neighboring frames with target feature maps using MLP to enhance the features of target frame. After that, we feed the enhanced features into a CNN-based decoder to generate the final inpainted frames $\bar{\mathbf{X}}_{en}$.

Flow Estimation

We adopt a lightweight flow estimation network SpyNet [134] to predict the forward and backward flows between neighboring frames so as to save the computation cost. We use $F_{t-1\rightarrow t}$ and $F_{t+1\rightarrow t}$ to denote the forward and backward flows, respectively.

Feature Enhancement

As illustrated in Fig. 4.5, we construct a Parallel Content Enhancement Block (PCEB) that is developed based on flow-guided deformable warping to strengthen the temporal coherence of the video. In this work, the proposed PCEB is applied to enhance a group of

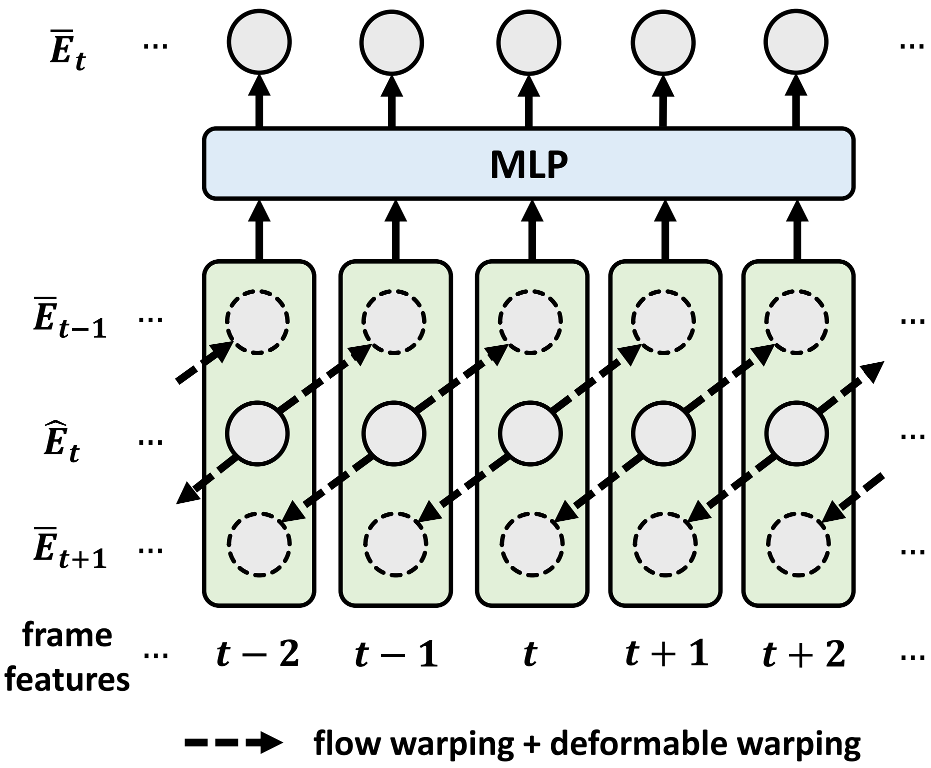


Fig. 4.5 Architecture of PCEB.

local neighboring frames $\{...,\hat{X}_{t-3},\hat{X}_{t-2},\hat{X}_{t-1},\hat{X}_{t+1},\hat{X}_{t+2},\hat{X}_{t+3},...\}$. We describe its implementation using two neighboring frames, $\{\hat{X}_{t-1},\hat{X}_{t+1}\}$, but it can also be extended to work with four frames, $\{\hat{X}_{t-2},\hat{X}_{t-1},\hat{X}_{t+1},\hat{X}_{t+2}\}$, or six frames, $\{\hat{X}_{t-3},\hat{X}_{t-2},\hat{X}_{t-1},\hat{X}_{t+1},\hat{X}_{t+2},\hat{X}_{t+3}\}$, in the implementation of our method. Given the features $\hat{\mathbf{E}}_l$ of the composed frames $\hat{\mathbf{X}}_l$, we stack K PCEBs to produce the enhanced features $\bar{\mathbf{E}}_{en}$ and each PCEB is implemented in four steps.

Firstly, the features $\hat{\mathbf{E}}_{t-1}$ and $\hat{\mathbf{E}}_{t+1}$ extracted from the frames $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ are simultaneously warped to the features $\hat{\mathbf{E}}_t$ of the inpainted frame $\hat{X}_t$ as

$$\begin{aligned}\hat{\mathbf{E}}'_{t-1} &= \mathcal{W}(\hat{\mathbf{E}}_{t-1},\mathbf{F}_{t-1\to t})\\ \hat{\mathbf{E}}'_{t+1} &= \mathcal{W}(\hat{\mathbf{E}}_{t+1},\mathbf{F}_{t+1\to t}),\end{aligned} \tag{4.11}$$

where $\mathcal{W}(\cdot)$ denotes the flow-based warping [133, 135].

Secondly, we predict the offset residuals and modulation masks with several convolution layers so that we can use them to implement the deformable warping for feature enhancement.

The offset residuals and modulation masks are obtained as

$$\{\mathbf{O}_{t-1\rightarrow t},\mathbf{O}_{t+1\rightarrow t},\mathbf{M}_{t-1\rightarrow t},\mathbf{M}_{t+1\rightarrow t}\} = \mathrm{Conv}(\mathrm{Concat}(\mathbf{F}_{t-1\rightarrow t},\mathbf{F}_{t+1\rightarrow t},\hat{\mathbf{E}}^{'}_{t-1}\hat{\mathbf{E}}^{'}_{t+1})), \quad (4.12)$$

where $\mathbf{O}_{t-1\rightarrow t}$ and $\mathbf{O}_{t+1\rightarrow t}$ are the predicted offset residuals, $\mathbf{M}_{t-1\rightarrow t}$ and $\mathbf{M}_{t+1\rightarrow t}$ are the predicted modulation masks, and Conv denotes the application of convolutional layers.

Thirdly, with the predicted offset residuals and modulation masks, we employ the deformable warping to warp features $\bar{\mathbf{E}}_{t-1}$ and $\bar{\mathbf{E}}_{t+1}$ to guarantee the features of neighboring frames can be effectively aligned to $\hat{\mathbf{E}}_t$. The deformable warping is implemented as

$$\begin{aligned}\bar{\mathbf{E}}_{t-1} &= \mathrm{DConv}(\hat{\mathbf{E}}^{'}_{t-1},\mathbf{F}_{t-1\rightarrow t}+\mathbf{O}_{t-1\rightarrow t},\mathbf{M}_{t-1\rightarrow t})\\ \bar{\mathbf{E}}_{t+1} &= \mathrm{DConv}(\hat{\mathbf{E}}^{'}_{t+1},\mathbf{F}_{t+1\rightarrow t}+\mathbf{O}_{t+1\rightarrow t},\mathbf{M}_{t+1\rightarrow t}),\end{aligned} \quad (4.13)$$

where DConv denotes the deformable convolution [136].

Finally, we concatenate $\bar{\mathbf{E}}_{t-1}$, $\bar{\mathbf{E}}_{t+1}$ and $\hat{\mathbf{E}}_t$ to fuse them with MLP to obtain the enhanced features $\bar{\mathbf{E}}_t$ as

$$\bar{\mathbf{E}}_t = \mathrm{MLP}(\mathrm{Concat}(\hat{\mathbf{E}}_t,\bar{\mathbf{E}}_{t-1},\bar{\mathbf{E}}_{t+1})), \quad (4.14)$$

where Concat is the concatenation operation. The features of each frame can be simultaneously fused with the features of its neighboring frames to obtain all the enhanced features $\bar{\mathbf{E}}_{en}$.

Enhanced Frame Reconstruction

After obtaining the enhanced features $\bar{\mathbf{E}}_{en}$ from the K^{th} PCEB, we feed them into a CNN-based decoder that consists of four convolutional layers to progressively increase the resolution of features and generate the enhanced inpainting frames $\bar{\mathbf{X}}_{en}$. All the enhanced frames are used to compose the final output video.

4.2.5 Loss Function

We construct the loss function $\mathcal{L}_{total}$ to train our model and jointly optimizing all the modules. $\mathcal{L}_{total}$ is composed as

$$\mathcal{L}_{total} = \lambda_{dep}\cdot\mathcal{L}_{dep}+\lambda_{con}\cdot\mathcal{L}_{con}+\lambda_{enh}\cdot\mathcal{L}_{enh}+\lambda_{gen}\cdot\mathcal{L}_{gen}, \quad (4.15)$$

where $\mathcal{L}_{dep}$ is the depth completion loss, $\mathcal{L}_{con}$ is the content construction loss, $\mathcal{L}_{enh}$ is the content enhancement loss, $\mathcal{L}_{gen}$ is the T-PatchGAN loss [56], and λ_{dep}, λ_{con}, λ_{enh} and λ_{gen} are the corresponding weighting factors for each loss.

In $\mathcal{L}_{total}$, the depth completion loss measures the difference between the predicted depth information $\hat{\mathbf{D}}$ and the ground-truth depth information $\mathbf{D}$. It is defined as

$$\mathcal{L}_{dep} = ||\hat{\mathbf{D}} - \mathbf{D}||_1. \tag{4.16}$$

The content construction loss $\mathcal{L}_{con}$ measures the difference between the reconstructed video $\hat{\mathbf{X}}$ obtained from the content reconstruction module and the ground-truth video $\mathbf{X}$. It is formulated as

$$\mathcal{L}_{con} = ||\hat{\mathbf{X}} - \mathbf{X}||_1. \tag{4.17}$$

The content enhancement loss $\mathcal{L}_{enh}$ measures the difference between the final output video $\bar{\mathbf{X}}$ obtained from the content enhancement module and the ground-truth video $\mathbf{X}$, i.e.,

$$\mathcal{L}_{enh} = ||\bar{\mathbf{X}} - \mathbf{X}||_1. \tag{4.18}$$

The T-PatchGAN loss [56] evaluates the difference between the final inpainted video $\bar{\mathbf{X}}$ and the ground-truth video $\mathbf{X}$ with a T-PatchGAN discriminator [56] $\mathcal{D}$, where the discriminator makes the model generate high-quality and realistic contents. The T-PatchGAN loss is formulated as

$$\mathcal{L}_{gen} = -\mathbb{E}_{\bar{\mathbf{X}}}[\mathcal{D}(\bar{\mathbf{X}})]. \tag{4.19}$$

Moreover, the T-PatchGAN discriminator consists of six 3D convolution layers and is used to learn the difference between real patches of ground-truth videos and fake patches of inpainted videos. The loss adopted in Chang's work [56] is employed to train the discriminator in this work, making the discriminator correctly classify real and fake samples with a clear margin. It is formulated as

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{X}}(0, 1 - \mathcal{D}(\mathbf{X}))] + \mathbb{E}_{\hat{\mathbf{X}}}[\max(0, 1 + \mathcal{D}(\hat{\mathbf{X}}))]. \tag{4.20}$$

where $\mathcal{D}(x)$ represents the discriminator's output for a real video sample x and $\mathcal{D}(z)$ represents the output for an inpainting video sample z.

4.3 Experimental Results

4.3.1 Settings

Datasets

We evaluated the proposed method on two widely used video object segmentation datasets, YouTube-VOS [11] and DAVIS [10], to demonstrate its effectiveness. The YouTube-VOS dataset consists of 3,471, 474, and 508 video clips for training, validation, and testing,

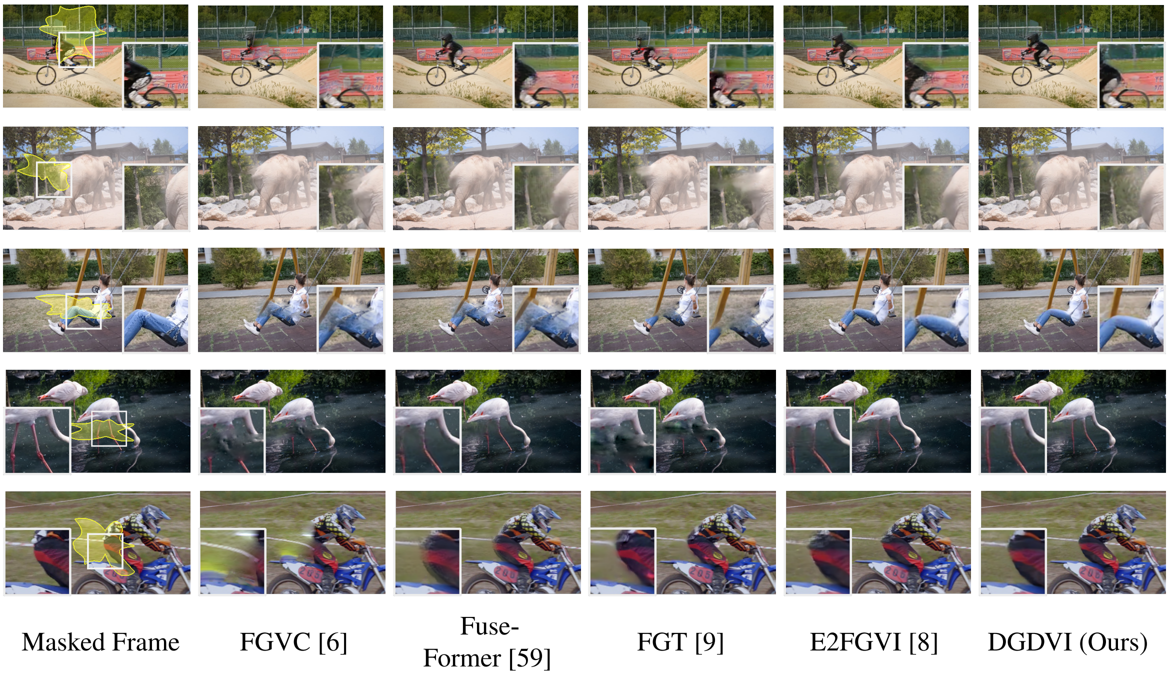


Fig. 4.6 Qualitative results for the video completion scenario that crosses foreground and background. From top to bottom: *Bmx-bumps*, *Elephant*, *Swing*, *Flamingo*, and *Motocross-bumps* videos of the DAVIS [10] dataset.

respectively, covering various scenes. The DAVIS dataset contains 60 videos in the training set and 90 videos in the test set.

We trained our model using the YouTube-VOS dataset and evaluated the performance using both the DAVIS and YouTube-VOS datasets. Specifically, following the initial partitioning of YouTube-VOS dataset, we used its training set to train our model. Moreover, to make our proposed method applicable to different inpainting scenarios, we created both the stationary irregular masks and the dynamic object-shaped masks as [55, 57, 4, 58, 59, 8] did, and applied them to the source videos to produce broken videos by removing the masked contents. To evaluate the performance of the method, we conducted evaluations on the YouTube-VOS test set and 50 video clips from the test set of DAVIS dataset as the previous work [58, 59, 8] did.

Implementation Details

During training, the numbers of local frames N_l and non-local frames N_{nl} were both set to 4. During test, the number of local frames N_l was set to 6, while the step-size to uniformly sample non-local frames N_{nl} was set to 6. In the experiment, the model adopted 8 STTBs in the depth completion module, 8 DGSTTBs in the content reconstruction module and 4

PCEBs in the content enhancement module, i.e., $P = 8$, $Q = 8$, and $K = 4$. The number of content feature C_{cn} was set to 128, while the number of depth feature C_{dep} was set to 64 in the depth completion and 32 in the content reconstruction. The head number k of both MSA and MMSA were set to 4. We first trained the depth completion module independently using $\mathcal{L}_{dep}$ for $300K$ iterations. Then with the depth completion module and the pretrained flow estimation network, SpyNet, frozen, we trained the content reconstruction and content enhancement modules using $\mathcal{L}_{con}$, $\mathcal{L}_{enh}$ and $\mathcal{L}_{gen}$ for $300K$ iterations. And we finetuned three modules together using $\mathcal{L}_{total}$ for $200K$ iteration. The weighting factors for λ_{dep}, λ_{con}, λ_{enh} and λ_{gen} were set to 0.2, 0.2, 1 and $1e{-}3$, respectively. We adopted Adam optimizer [137] to train our network. The initial learning rate was $1e{-}4$, which was divided by 10 after $150K$ iterations. The resolution of training videos were resized to 240×432 and the batch size was set to 4. The training of our network was implemented on Pytorch platform with two NVIDIA GeForce RTX 3090 GPUs, while the test experiments were implemented with one NVIDIA GeForce RTX 3090 GPU.

Evaluation Metrics

We adopted peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [138], video Frechet inception distance (VFID) [113], and flow warping error (E_{warp}) [114] as the quantitative metrics to evaluate the performance of different video inpainting methods. More specifically, PSNR and SSIM are two widely used metrics to assess reconstructed image and video with original ones. Higher value suggests higher similarity. VFID was employed to assess the perceptual similarity of distortion-oriented videos and has been adopted in recent video inpainting approaches [58, 59, 8]. Lower value represents better realism and less distortion compared with natural videos. Flow warping error E_{warp} measures the temporal consistency based on optical flow. Lower score indicates better temporal consistency.

4.3.2 Comparisons

Qualitative Results

We qualitatively compared our method with four latest approaches, including FGVC [6], FuseFormer [59], FGT [9] and E2FGVI [8].

The comparison was conducted on two tasks. The first one was video completion and the second one was object removal, where both the tasks were performed on the videos of the DAVIS dataset. Moreover, we created the stationary irregular mask for the video completion task as the previous methods [58, 59, 61, 8] did. In this task, one static mask was randomly applied to a video to remove the content. In contrast, we produced dynamic object-shaped

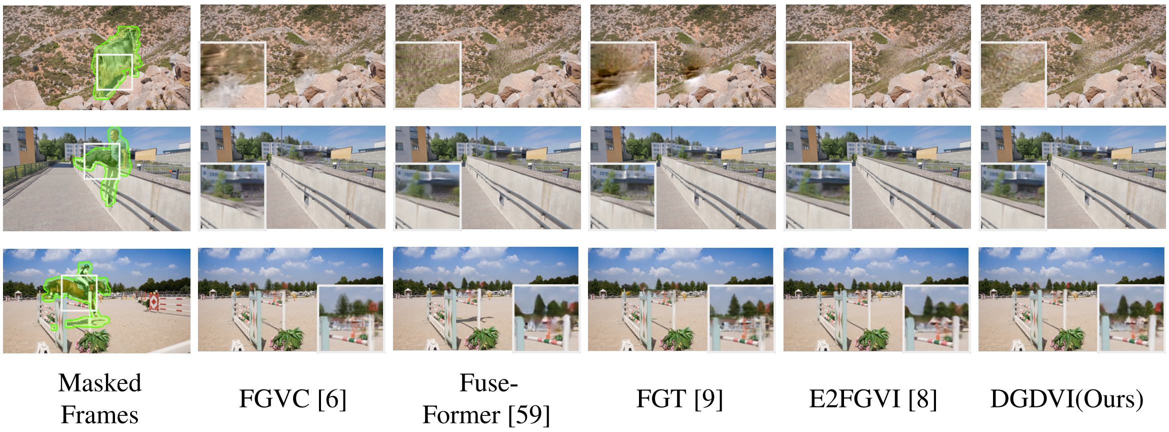


Fig. 4.7 Qualitative results for the object removal scenario. From top to bottom: *Goat*, *Parkour*, and *Horsejump-high* videos of the DAVIS [10] dataset.

masks for the object removal task, where each mask covered one moving object over the whole video.

Some video completion results for the challenging scenes crossing foreground and background were presented in Fig. 4.6 and some object removal results were presented in Fig. 4.7. One could see from Fig. 4.6 and Fig. 4.7 that our method generated more reliable contents and clearer structures than the other approaches, demonstrating its effectiveness.

Quantitative Results

We conducted quantitative comparison on YouTube-VOS and DAVIS for video completion. The resolution of test videos was 432×240. The proposed method was compared to VINet [55], DFVI [51], LGTSM [57], CPNet [4], spatial-temporal transformations for video inpainting (STTN) [58], axial attention-based style Transformer (AAST) [61], FGVC [6], FuseFormer [59], FGT [9], and E2FGVI [8]. The corresponding results were given in Table 4.1. It was found from Table 4.1 that our method significantly outperforms all the state-of-the-art methods evaluated by the four quantitative metrics. These results indicated that our approach could recover the contents with less distortion (PSNR and SSIM), more visually faithful content (VFID), and better spatial and temporal consistency (E_{warp}).

Complexity Analysis

We used Floating Point Operations Per second (FLOPs) and inference time to evaluate the complexity of the compared methods by using the DAVIS dataset. The corresponding results were presented in Table 4.1. The FLOPs of our proposed approach were comparable to VINet [55], LGTSM [57] and CPNet [4] that were developed based on CNN. Meanwhile, the proposed method executed about ×10 faster than DFVI [51], FGVC [6] and FGT [9]. In

Table 4.1 Quantitative comparisons on YouTube-VOS [11] and DAVIS [10] datasets. ↑ indicates higher is better. ↓ indicates lower is better. $E_{warp}{}^{*}$ denotes $E_{warp} \times 10^{-2}$. Each method is evaluated following the procedures in FuseFormer. VINet, DFVI, FGVC, and FGT are not end-to-end training methods. Their Floating Point Operations Per second (FLOPs), thus, are not presented. AAST did not provide the source code. As such its FLOPs and runtime are not provided. The best results for each metric are highlighted in bold.

Methods	YouTube-VOS				DAVIS				FLOPs	Runtime
	PSNR (dB) ↑	SSIM ↑	VFID ↓	$E_{warp}{}^{*}$ ↓	PSNR (dB) ↑	SSIM ↑	VFID ↓	$E_{warp}{}^{*}$ ↓		(s/frame)
VINet [55]	29.20	0.9434	0.072	0.1490	28.96	0.9411	0.199	0.1785	-	-
DFVI [51]	29.16	0.9429	0.066	0.1509	28.81	0.9404	0.187	0.1608	-	2.56
LGTSM [57]	29.74	0.9504	0.070	0.1859	28.57	0.9409	0.170	0.1640	1008G	0.23
CPNet [4]	31.58	0.9607	0.071	0.1470	30.28	0.9521	0.182	0.1533	861G	0.40
FGVC [6]	29.67	0.9403	0.064	0.1022	30.80	0.9497	0.165	0.1586	-	2.36
STTN [58]	32.34	0.9655	0.053	0.0907	30.67	0.9560	0.149	0.1449	1032G	0.12
FuseFormer [59]	33.29	0.9681	0.053	0.0900	32.54	0.9700	0.138	0.1362	752G	0.20
AAST [61]	33.23	0.9669	0.048	0.1396	32.71	0.9720	0.1360	0.1706	-	-
FGT [9]	30.19	0.9536	0.063	0.0968	31.77	0.9639	0.134	0.1483	-	1.89
E2FGVI [8]	33.71	0.9700	0.046	0.0864	33.01	0.9721	0.116	0.1315	682G	0.16
DGDVI (Ours)	**34.07**	**0.9725**	**0.045**	**0.0823**	**33.33**	**0.9740**	**0.111**	**0.1295**	860G	0.21

these methods, the optical flow was adopted to guide the information propagation throughout the frames for inpainting, resulting in rather high complexity. Meanwhile, our method achieved comparable speeds to the Transformer-based approaches, such as STTN [58], FuseFormer [59], and E2FGVI [8].

4.3.3 Ablation Study

We conducted ablation studies to verify the effectiveness of the proposed modules, MMSA and flow-guided deformable warping used in our model. All the studies were performed on the DAVIS dataset for the video completion task.

Effectiveness of the Proposed Modules in DGDVI

Our proposed inpainting model consists of three modules, i.e., the depth completion, content reconstruction, and content enhancement modules. To demonstrate the performance gain offered by them, we conducted an ablation study to verify their effectiveness. The content reconstruction module is the key module in our model. Once it is removed, our proposed inpainting model could not work any longer. Therefore, it was always retained in our model when the ablation study was carried out.

When we conducted this ablation study, we firstly just used the content reconstruction module to construct a baseline model, denoted as *Model-1*, where the content reconstruction

Table 4.2 Ablation study for the proposed modules in DGDVI. Different combinations of the content reconstruction, depth completion, and content enhancement modules are evaluated on the DAVIS [10] dataset. Performance is measured using PSNR and SSIM metrics, with higher values indicating better reconstruction quality. The best results for each metric are highlighted in bold.

	Model-1	*Model*-2	*Model*-3	*Model*-4
Content reconstruction	✓	✓	✓	✓
+ Depth completion	✗	✗	✓	✓
+ Content enhancement	✗	✓	✗	✓
PSNR (dB) ↑/ SSIM ↑	32.54/0.9700	33.04/0.9718	33.08/0.9724	**33.33/0.9740**

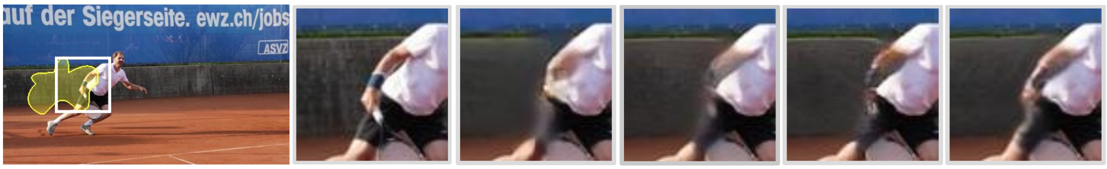


Fig. 4.8 Ablation study for the proposed modules. From left to right: Masked frame of *Tennis* video, portions for ground truth, model-1, model-2, model-3, and model-4.

module is implemented with MSA rather than MMSA (as depth was not available for guidance). Then, we added the content enhancement module to *Model-1* to compose *Model-2* for the verification of effectiveness of this module. Meanwhile, we composed *Model-3* by using the content reconstruction and depth completion modules, where MMSA was adopted in content reconstruction because the depth could be offered by the depth completion module. Finally, we integrated all the modules to build up our proposed inpainting model (denoted as *Model-4* in this experiment). The quantitative and qualitative results for the ablation study were given in Table 4.2 and Fig. 4.8, respectively.

According to the results presented in Table 4.2 and Fig. 4.8, it was found that introducing the depth completion and content enhancement modules to the baseline model, i.e., *Model-1*, could effectively improve the inpainting quality, both quantitatively and qualitatively, When all the modules were adopted, the best quality was achieved. These results demonstrated the effectiveness of the proposed modules.

In addition, we presented some visualized results in Fig. 4.9 to further verify the effectiveness of our proposed modules. Firstly, it was found from Fig. 4.9 that the predicted depth for the broken video was very similar to the depth of the ground-truth video. This accordingly demonstrated the effectiveness of the depth completion module. Secondly, guided by the

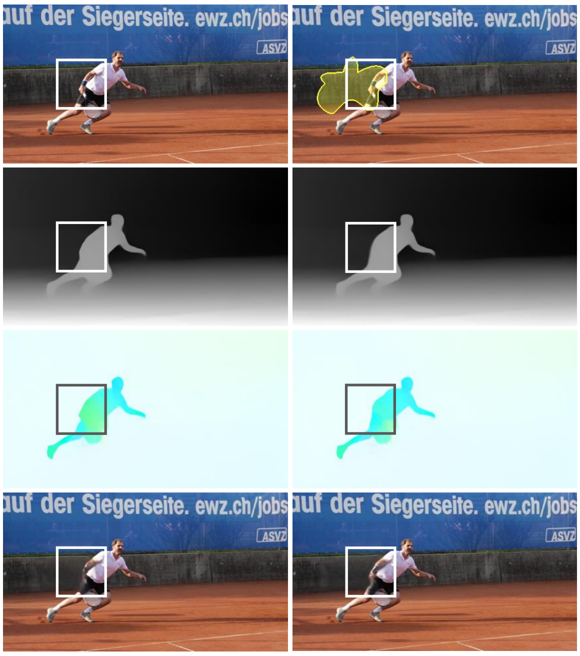


Fig. 4.9 The intermediate results for DGDVI. The example is selected the same frame as Fig. 4.8 from *Tennis* video. From left to right, top to bottom: Ground-truth frame, mask, depth estimation result for ground-truth frame, depth completion result for broken video, visualization of flow estimation result for ground-truth video, visualization of flow estimation for initial inpainted video, initial inpainted result, enhanced inpainted result.

predicted depth, we obtained the initial inpainted video with acceptable quality by using the content reconstruction module. Note that employing this initial result could generate optical flow similar to the one obtained from the ground-truth video. Finally, with the obtained flow, we enhanced the initial result and get the final inpainting result with higher quality, which validated the effectiveness of the content enhancement module.

Effectiveness of MMSA for Content Reconstruction

The depth-guided feature propagation in the content reconstruction was developed based on the proposed MMSA mechanism. To verify the effectiveness of MMSA, we replaced it with MSA in content reconstruction to evaluate the change of inpainting performance. Specifically, the content reconstruction with MSA first fused depth and feature, and then fed them into STTBs for feature propagation. The quantitative results were given in Table 4.3. According to the results in Table 4.3, it was found that the model with MMSA achieves better inpainting performance than using MSA, which demonstrated the superiority of MMSA for the content reconstruction.

Table 4.3 Ablation study for MMSA in depth-guided content reconstruction module. MSA and MMSA are separately applied to the depth-guided content reconstruction module and evaluated on the DAVIS [10] dataset. Performance is measured using PSNR, SSIM, and VFID metrics. The best results for each metric are highlighted in bold.

	PSNR (dB) ↑	SSIM ↑	VFID ↓
MSA	32.74	0.9716	0.124
MMSA	**33.30**	**0.9740**	**0.111**

Table 4.4 Ablation study for flow-guided deformable warping in content enhancement module. Flow-based warping, deformable warping, and flow-guided deformable warping are separately applied to the content enhancement module and evaluated on the DAVIS [10] dataset. Performance is measured using PSNR, SSIM, and VFID metrics. The best results for each metric are highlighted in bold.

	PSNR (dB) ↑	SSIM ↑	VFID ↓
Flow-based	32.75	0.9710	0.121
Deformable	32.61	0.9691	0.125
Flow-guided deformable	**33.30**	**0.9740**	**0.111**

Effectiveness of the Flow-guided Deformable Warping for Content Enhancement

The content enhancement module was developed based on the flow-guided deformable warping. In order to verify the superiority of this warping approach, we compared its performance with two warping methods, flow-based warping and deformable convolution-based warping. Specifically, instead of using flow-guided deformable warping in content enhancement, we implemented flow-based or deformable convolution-based warping to align features for feature enhancement. The corresponding quantitative and qualitative results inpainting results were presented in Table 4.4 and Fig. 4.10, respectively. It could be found from Table 4.4 and Fig. 4.10 that the model with flow-guided deformable warping generated the best results, demonstrating the effectiveness of this warping technique.

Effectiveness of Incorporating Non-local Frames for Video Inpainting

To validate the effectiveness of adopting non-local frames in the inpainting, we firstly conducted an experiment by just employing the local frames to implement the inpainting with our proposed model. Then, we compared the inpainting results with the ones obtained by employing both local and non-local frames to inpaint videos. The corresponding quantitative and qualitative results were given in Table 4.5 and Fig. 4.11, respectively. These results

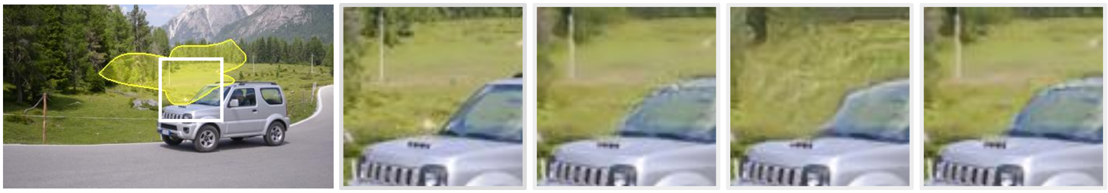


Fig. 4.10 Ablation study for flow-guided deformable warping. From left to right: Masked frame of *Car-turn* video, portions for ground truth, flow-based warping, deformable convolution-based warping, and flow-guided deformable warping.

Table 4.5 Quantitative results verifying the effectiveness of incorporating non-local frames in the inpainting process. Local frames and a combination of local and non-local frames are separately applied to our proposed DGDVI model and evaluated on the DAVIS [10] dataset. Performance is measured using PSNR, SSIM, and VFID metrics, with the best results for each metric highlighted in bold.

	PSNR (dB) ↑	SSIM ↑	VFID ↓
Local frames	32.55	0.9687	0.126
Local + non-local frames	**33.30**	**0.9740**	**0.111**

demonstrated the superior performance of our approach when both local and non-local frames were adopted.

4.4 Summary

In this chapter, we proposed a depth-guided deep video inpainting (DGDVI) network designed to address challenges in complex scenes, particularly for missing regions spanning both moving targets and backgrounds. Previous methods often struggled to determine whether missing pixels belonged to foreground or background layers, resulting in noticeable artifacts in such scenarios. To overcome this limitation, we introduced depth information as a guiding mechanism for the inpainting process. The proposed model comprises three key modules: depth completion, content reconstruction, and content enhancement, implemented in a sequential workflow. Specifically, the depth completion module, built upon a spatio-temporal Transformer architecture, predicts completed depth maps for each video frame. The content reconstruction module, guided by the predicted depth information, generates an initial inpainting result using a depth-guided spatio-temporal Transformer. The content enhancement module, leveraging a flow-guided deformable convolutional network, refines the initial results to improve visual quality and temporal consistency, producing the final

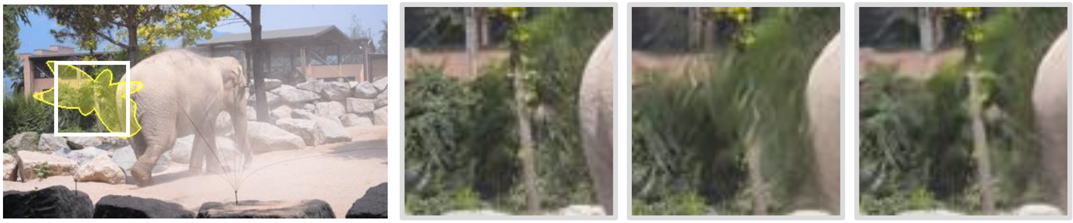


Fig. 4.11 Qualitative results for the effectiveness verification of adopting non-local frames to implement inpainting. From left to right: Masked frame of *Elephant*, portions for ground truth, inpainting just with local frames, and inpainting with both local and non-local frames.

inpainting outputs. These modules are jointly optimized to ensure high efficiency and coherence throughout the inpainting process.

We evaluated the proposed DGDVI on the DAVIS and YouTube-VOS datasets for both video restoration and object removal tasks. The results demonstrate that our method effectively reconstructs clear structures and produces visually plausible results, particularly for scenes with multiple depth layers in the missing regions. Comparative analyses were conducted against SOTA methods [55, 51, 57, 4, 58, 59, 61, 6, 9, 8]. Both qualitative and quantitative evaluations across metrics, including reference techniques, demonstrate that DGDVI consistently outperforms these methods, establishing a new benchmark for video inpainting.

Despite its impressive performance on video restoration and object removal tasks, the proposed DGDVI faces challenges with high-resolution video inpainting. Due to limitations in model design and inefficient computational and memory management, DGDVI is restricted to handling 240p resolution videos. To address this issue, the next chapter introduces a hierarchical sparse Transformer model designed to extend video inpainting capabilities to high-resolution tasks effectively.

5

A Hierarchical Sparse Transformer for High-resolution Video Inpainting

5.1 Introduction

In the previous two chapters, we proposed a short-long-term propagation-based video inpainting approach for achieving visually and contextually consistent object removal, as well as a depth-guided deep video inpainting network for addressing complex scenes with multiple layers. While the former demonstrated impressive performance in target removal tasks, it has limitations when applied to video restoration tasks. In contrast, the latter showed strong robustness in both object removal and video restoration tasks, particularly in challenging scenarios where both foreground and background regions are missing. However, deep learning-based methods often face significant challenges when dealing with high-resolution videos due to inherent limitations in model design and inefficiencies in computation and memory management. To overcome these challenges, this chapter introduces a hierarchical sparse Transformer specifically designed for high-resolution video inpainting.

Despite a number of video inpainting approaches [51, 6, 58, 59, 8, 9, 139] proposed in recent years, it remains as one of the key challenges to handle high computational demand in video inpainting, particularly in high-resolution inpainting scenarios. Images and videos are fundamentally highly sparse matrices, where pixels within local regions exhibit minimal variations, resulting in significant amount of redundant information. As the spatial or temporal dimensions increase in high-resolution videos, the data volume increases exponentially, and the sparsity within video is amplified as well. For example, processing a $1080p$ (1920×1080) high-resolution video at 24 fps generates approximately 1.2×10^9 bits of data per second. This vast amount of data presents two problems. On the one hand, it is difficult for the model to accurately capture the most relevant cues in reconstructing realistic contents amidst the massive data, which often leads to artifacts in inpainting regions. On the other hand, it creates a great burden for the model to manage both computation and memory consumption in processing long and high-resolution videos. Failures of efficient processing of redundant information within videos lead to collapse in many inpainting approaches due to the computation and memory overload. Therefore, it is important to develop an efficient scheme for high-resolution video inpainting.

In recent years, several approaches [58, 60, 59, 8, 9] utilize the Transformer model to reconstruct missing content due to its great ability to model long-range dependencies in videos. These methods typically transform video frames into various token embeddings, with reference information being propagated through interactions of these tokens via MSA in both spatial and temporal domains. To generate visually and contextually coherent results, tokens from missing regions interact with tokens from existed regions in both local and non-local ranges, thereby acquiring short-range and long-range references. Some methods [58, 60, 59] adopt a global manner to make all the tokens interact with each other through MSA. However, a significant increase in token numbers from high-resolution videos leads to a quadratic increase in computational demand. Other methods [8, 9] use optical flow to align corresponding regions between adjacent frames, achieving token interaction within local spatio-temporal windows and thereby reducing computational costs. However, while this design effectively propagates short-term information, it struggles to capture long-range cues from distant frames. Furthermore, effective flow-guided information propagation relies on accurate optical flow prediction for missing regions. Since information is completely lost in these missing regions, flow completion is prone to inaccuracies and errors, especially when dealing with large missing areas in high-resolution videos. Such flow errors can lead to artifacts in the inpainted results.

To address the issues mentioned above, we propose a hierarchical sparse Transformer for high-resolution video inpainting. To efficiently manage the substantial computational cost of capturing spatio-temporal contexts necessary for producing both reliable structures and fine textures in high-resolution inpainting, we process the video in a coarse-to-fine manner using two stages. Initially, the video is processed at a lower resolution to reconstruct the foundational structure of the corrupted regions. By processing a sequence of low-resolution frames sparsely sampled along the temporal dimension, we can efficiently capture the overall context for initial content reconstruction without being overwhelmed by the massive data of high-resolution inputs. Once the low-resolution reconstruction is complete, we further refine the result by enhancing textures and details at higher resolutions under the flow guidance, ensuring that the final output retains both temporal consistency and fine-grained visual quality. The content enhancement is merely performed within local windows of the corrupted regions in consecutive frames, with most valid regions cropped to save computational costs. Additionally, the complete optical flow can be directly obtained from low-resolution results, which eliminates the need for a separate flow completion step to estimate the missing flow information in missing regions. This not only reduces computational complexity but also

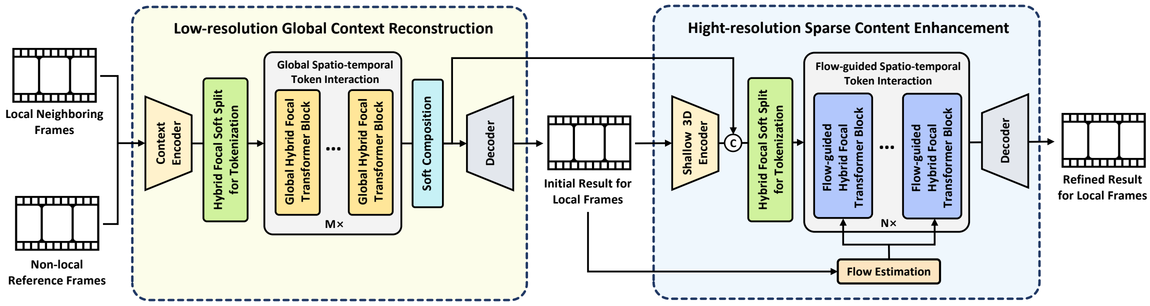

Local Neighboring Frames
Non-local Reference Frames
Low-resolution Global Context Reconstruction
Context Encoder
Hybrid Focal Soft Split for Tokenization
Global Spatio-temporal Token Interaction
Global Hybrid Focal Transformer Block
...
Global Hybrid Focal Transformer Block
M×
Soft Composition
Decoder
Initial Result for Local Frames
Hight-resolution Sparse Content Enhancement
Shallow 3D Encoder
C
Hybrid Focal Soft Split for Tokenization
Flow-guided Spatio-temporal Token Interaction
Flow-guided Hybrid Focal Transformer Block
...
Flow-guided Hybrid Focal Transformer Block
N×
Decoder
Flow Estimation
Refined Result for Local Frames

Fig. 5.1 Framework of the proposed HSTVI.

improves the reliability of optical flow, thereby enhancing the final inpainting performance. Overall, our contributions for this chapter can be summarised as follows.

- We introduce a HSTVI approach, which is designed to address the high computational issue of high-resolution inpainting with two stages, i.e. LGCR and HSCE.
- We develop a LGCR module to synthesize initial results at low-resolutions by using sparse global contexts.
- We design a HSCE module to enhance the details and temporal consistency of inpainting results at high-resolutions within sparse spatio-temporal windows.
- Our approach significantly exceeds the state-of-the-art methods on DAVIS dataset [10] while achieving comparable performance on YouTube-VOS dataset [11], assessed through quantitative and qualitative measurements, which demonstrates the superiority of our approach.

5.2 Methodology

5.2.1 Overview

Given a corrupted video X containing N frames $\{X_1, X_2, \ldots, X_N\}$, where $X_i \in \mathbb{R}^{H\times W\times 3}$, we aims to fill missing regions with reasonable and visually coherent contents. We propose a HSTVI to achieve video inpainting both effectively and efficiently, even for the challenging high-resolution tasks. HSTVI completes video inpainting in a coarse-to-fine manner with two modules, i.e. LGCR and HSCE. Firstly, we employ the LGCR module to generate an initial low-resolution inpainting result by integrating information from neighboring and non-local frames. Subsequently, we calculate optical flow for the initial results. Guided by the optical flows, we then utilize the HSCE module to refine texture details and enhance temporal consistency of the initial results, obtaining the final high-resolution inpainting results. The overall framework is illustrated in Fig. 5.1.

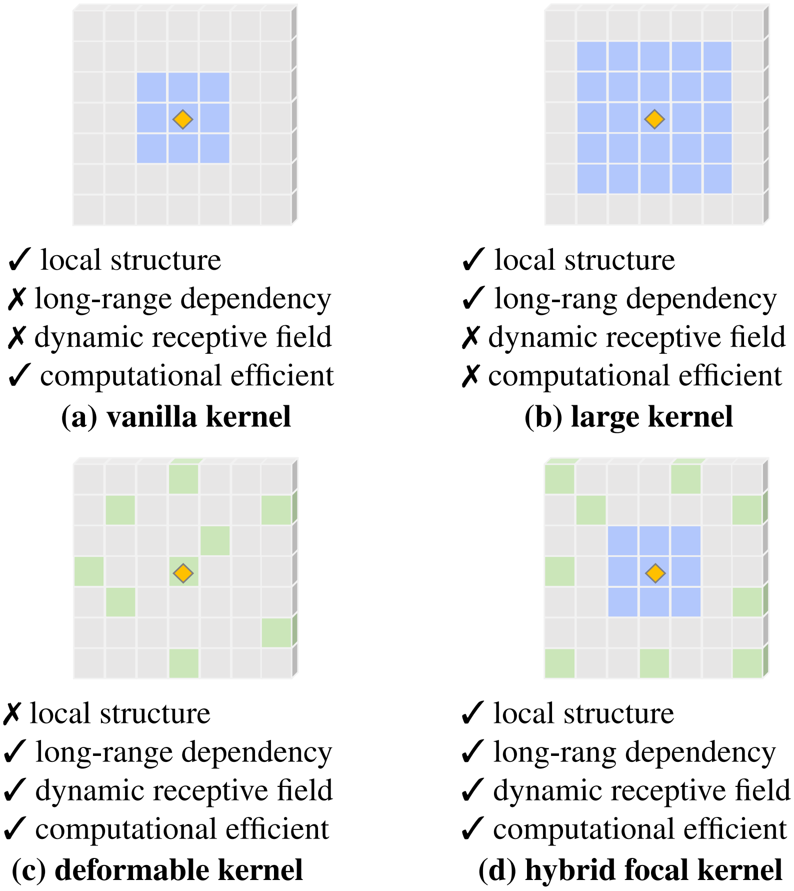
To build the HSTVI, we construct both the LGCR and HSCE modules based on the Transformer model. For each module, the Transformer model begins by converting videos into numerous hybrid focal tokens, allowing for dynamic aggregation of context information both locally and non-locally. With these tokens, the model facilitates token interaction to reconstruct the tokens from corrupted regions, propagating the information throughout the spatio-temporal domain in the video. Once the tokens are updated, they are reversed into videos, allowing the models to achieve context reconstruction and content enhancement, respectively.

In the following subsections, we will delve into the main components of our method. Firstly, we will intorduce the hybrid focal tokenization technique in Sec. 5.2.2, which plays a crucial role in our approach. Following that, in Sec. 5.2.2 and Sec. 5.2.4, we will provide detailed explanations of the designs of LGCR and HSCE, respectively. Finally, in Sec. 5.2.5, we will discuss the training strategy employed for our model.

5.2.2 Hybrid Focal Kernel-based Tokenization

In Transformer model, tokenization segments input data into parts that can be embedded into tokens within a vector space. These tokens interact with each other to build dependencies through MSA, facilitating the propagation of reference information from valid regions to corrupted regions in video inpainting task. Influenced by ViT [65], the tokenization in previous Transformer-based video inpainting approaches [58, 60, 59, 8] typically splits frames into a sequence of rectangular patches for the transformation of embedded tokens. Although vanilla kernel is effective at aggregating contextual information within a local field, it may struggle to capture reference cues in regions with missing contents. The poor-quality tokens extracted from these regions will limit the ability of model to build long-range dependencies, resulting in disappointing results for video inpainting. Employing larger vanilla kernels with wider receptive field might mitigate the issue, but it will significantly increase parameters of model and computational costs.

To tackle this issue, we design a Hybrid Focal Kernel-based Tokenization (HFKT), as illustrated in Fig. 5.2. The hybrid focal kernel combines a fixed dense kernel and a dynamic sparse kernel. The dense kernel aggregates structures and textures within a local area, while the sparse kernel captures long-range contexts within the frame. Inspired by deformable convolution [136, 140], we introduce learnable offsets to create adaptive sampling positions in the dynamic sparse kernel, allowing for a broader receptive field during feature



✓ local structure
✗ long-range dependency
✗ dynamic receptive field
✓ computational efficient

(a) vanilla kernel

✓ local structure
✓ long-rang dependency
✗ dynamic receptive field
✗ computational efficient

(b) large kernel

✗ local structure
✓ long-range dependency
✓ dynamic receptive field
✓ computational efficient

(c) deformable kernel

✓ local structure
✓ long-rang dependency
✓ dynamic receptive field
✓ computational efficient

(d) hybrid focal kernel

Fig. 5.2 Comparison of different kernels for feature extraction, where yellow, blue, and green positions represent feature aggregation centre, fixed dense positions and dynamic sparse positions, respectively.

extraction. When the fixed dense kernel has difficulties in extracting features from corrupted regions with little or no useful information locally, the dynamic sparse kernel helps to adaptively capture long-range contexts from neighboring valid regions. Hence, compared to tokenization adopted in previous approaches [58, 60, 59, 8], HFKT is potential to produce higher quality token embeddings, allowing the model to better establish correlations between tokens from corrupted and valid regions for more effective information propagation. Additionally, the computation and memory costs do not significantly increase when using HFKT for token extraction. Overall, HFKT is designed to efficiently generate superior tokens by aggregating both short-range and long-range information, thereby enhancing the final inpainting performance.

To employ HFKT for token extraction, we firstly determine the dynamic sparse sampling positions by using a convolutional layer Conv_1 to obtain offsets $\mathbf{O}$ and modulated masks $\mathbf{M}$, which can be expressed as

$$\{\mathbf{O}, \mathbf{M}\} = \mathrm{Conv}_1(\mathbf{E}), \tag{5.1}$$

where $\mathbf{E}$ represents the input frame or feature. Then the process of HFKT can be expressed as

$$\mathrm{HFKT}(\mathbf{E}) = \mathrm{Concat}(\mathrm{Conv}_2(\mathbf{E}), \mathrm{DConv}(\mathbf{E}, \mathbf{O}, \mathbf{M})), \tag{5.2}$$

where Concat denotes concatenation and Conv_2, DConv represent convolution and deformable convolution layers, respectively.

5.2.3 Low-resolution Global Context Reconstruction

The LGCR module aims to integrate valid information from both global spatial and temporal domains to generate reasonable and coherent content for missing regions at low resolutions. This stage facilitates the computation of reliable optical flow for further flow-guided content enhancement at high resolutions.

To effectively manage computation and memory costs, the given video clip $\mathbf{X}$ is firstly down-sampled into low-resolution frames and then divided into multiple frame groups for independent content reconstruction. The input to LGCR module, denoted as $\mathbf{X}_{in}$, consists of a local-frame group and a non-local-frame group, ultimately yielding the reconstructed local frame group X_{rec}. Each local frame group is selected using a sliding temporal window in the video to strengthen local spatio-temporal dependencies for inpainting, while the non-local frame group is obtained by uniformly sampling video frames with a specified step size to grasp global context cues for inpainting. By sparsely sampling both local and non-local frames, the LGCR module can efficiently acquire the reference information from the global context to reconstruct the missing region and save the computation cost as well.

LGCR is built upon on a global hybrid focal Transformer, which incorporates hybrid focal tokenization, global spatio-temporal token interaction, and initial frame composition. With this module, both local and non-local frame groups are initially transformed into hybrid focal tokens. Subsequently, all the tokens interact with each other on a global spatio-temporal scale for effective information propagation. Finally, the tokens are composited to reconstruct the initial inpainting result.

Tokenization for Context Reconstruction

To generate high-quality tokens for LGCR, the tokenization module is established with a hybrid focal kernel-based context encoder and a hybrid focal kernel-based soft split layer. Firstly, the context encoder,comprising nine convolutional layers based on the hybrid focal kernel, takes in the corrupted frames $\mathbf{X}_{in}$ and produces $1/4$ sized feature maps $\mathbf{E}_{in}$ with c_{rec}

channels. To transform feature maps into patch tokens, we introduce a Hybrid Focal Soft Split (HFSS) operation based on HFKT. With kernel size $(7,7)$ and stride $(3,3)$ for HFKT, HFSS divides the feature maps into overlapping hybrid focal patches and converts them into token embeddings $\mathbf{Z}_{rec}^{0}$ with c_t channels as

$$\mathbf{Z}_{rec}^{0} = \text{HFSS}(\mathbf{E}_{in}), \tag{5.3}$$

where $\mathbf{Z}_{rec}^{0}$ represents the tokens generated by HFSS for input of global spatio-temporal token interaction phase.

Global Spatio-temporal Token Interaction

Global token interaction aims to reconstruct tokens from corrupted regions by interacting with all other tokens to propagate information across spatio-temporal domains. This interaction is facilitated by a stack of Global Dynamic Sparse Transformer Blocks (GDSTB), which incorporates a Global Spatio-Temporal Multi-head Self-Attention (GST-MSA) and a Hybrid Focal Fusion Feed Forward Network (HF4N).

GST-MSA applies MSA [65, 96] to all tokens, enabling each token to establish dependencies with global context information for content reconstruction. All tokens are linearly projected to generate query, key, and value embeddings. Assuming there are k heads in GST-MSA, the computation for each head is expressed as

$$\text{Attn}(\mathbf{Q}_k, \mathbf{K}_k, \mathbf{V}_k) = \frac{softmax(\mathbf{Q}_k \mathbf{K}_k^T)}{\sqrt{d_k}} \mathbf{V}_k, \tag{5.4}$$

where Attn represents attention mechanism, $\mathbf{Q}_k$, $\mathbf{K}_k$, and $\mathbf{V}_k$ are the query, key, and value vectors for each head, respectively, and d_k is the feature dimension of $\mathbf{Q}_k$ and adopted as a scaling factor. Afterwards, we can obtain the update tokens $\mathbf{Z}_{attn}$ by connecting the result of GST-MSA with the input tokens $\mathbf{Z}_{rec}^{n}$ in residual manner.

After the token interactions via GST-MSA, we employ a hybrid focal fusion feedforward network to enhance the spatial connections between neighboring tokens. HF4N consists of two layers, a SC layer [59] and a HFSS layer. The tokens are firstly converted into overlapped vanilla patches and the patches are composited into feature maps with SC. After that, they are dynamically split into hybrid focal tokens with HFSS. The process of HF4N can be described as

$$\text{HF4N}(\mathbf{Z}_{attn}) = \text{HFSS}(\text{SC}(\mathbf{Z}_{\text{attn}})). \tag{5.5}$$

The whole process of GDSTB can be described as

$$\begin{aligned} \mathbf{Z}_{attn}^{n} &= \text{GST-MSA}(\text{LN}_1(\mathbf{Z}_{rec}^{n-1})) + \mathbf{Z}_{rec}^{n-1}, \\ \mathbf{Z}_{rec}^{n} &= \text{HF4N}(\text{LN}_2(\mathbf{Z}_{attn}^{n})) + \mathbf{Z}_{attn}^{n}, \end{aligned} \tag{5.6}$$

where $\mathbf{Z}_{rec}^{n-1}$ and $\mathbf{Z}_{rec}^{n}$ denote the input and output token embeddings of the n^{th} GDSTB, respectively and LN represents the layer normalization [132]. By stacking M GDSTBs to implement global spatio-temporal token interaction, it enables tokens from missing regions to be reconstructed by integrating the valid information from global context within the video.

Initial Low-resolution Frame Composition

Following global spatio-temporal token interactions, the information within tokens from corrupted regions is updated by integrating cues from both local and non-local frames. To obtain the initial low-resolution inpainting result, we employ SC [59] and a CNN-based decoder. With the SC operation, the tokens from local frames are converted into overlapped vanilla-patches, which are then composited into feature maps as

$$\mathbf{E}_{rec} = \mathrm{SC}(\mathbf{Z}_{rec}^{M}), \tag{5.7}$$

where $\mathbf{Z}_{rec}^{M}$ denotes the output from the M^{th} GDSTB block and $\mathbf{E}_{rec}$ is the composed feature map. Then, E_{rec} residual-connected with $\mathbf{E}_{in}$ is decoded by a CNN-based decoder, consisting of four convolutional layers, to generate the initial inpainting result X_{rec}, which has the same resolution as the input local-frame groups.

5.2.4 High-resolution Sparse Content Enhancement

After the initial context reconstruction of the damaged video at a lower resolution, the structure of missing regions is reconstructed and we further enhance the quality of initial results at high resolutions. Given the reconstructed local-frame group $\mathbf{X}_{rec}$, the content enhancement module is designed to derive the refined result $\mathbf{X}_{enh}$, thereby improving visual consistency and texture details for the inpainted video. Since the global contexts from distant frames have been utilized to reconstruct the overall structure in the initial low-resolution results, our focus shifts to enhancing the quality of the inpainting regions by leveraging spatio-temporal cues from neighboring frames. Additionally, to optimize computational efficiency, we prevent the unnecessary computational and memory costs that would otherwise be incurred by processing the high-resolution valid regions.

We develop HSCE based on a flow-guided hybrid focal Transformer, which consists of dynamic sparse tokenization, flow-guided spatio-temporal token interaction, and enhanced frame composition. The frames are first divided into many large windows, and the windows that contain inpainting regions will be selected for content enhancement. To employ flow guidance for content enhancement in selected window, we first calculate the forward and backward optical flows across neighboring frames of reconstructed local frame groups. After the reconstructed local frames are transformed into hybrid focal tokens, the tokens

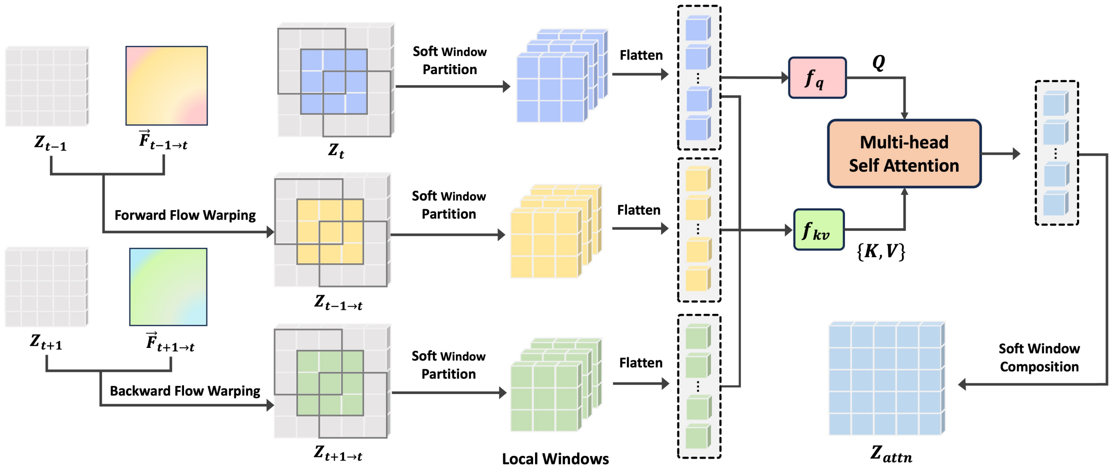


Fig. 5.3 Illustration of Flow-Guided Soft Window Multi-head Self-Attention (FGSW-MSA), in which $Z_t - 1$, Z_t, and Z_{t+1} denote tokens of frame $t-1$ to $t+1$ in E^n_{enh} while $\vec{F}_{t-1\rightarrow t}$ and $\vec{F}_{t+1\rightarrow t}$ represent their forward and backward optical flows.

interact with others and neighboring frames within the window via a flow-guided soft window attention mechanism for information propagation. Finally, the tokens are assembled into refined frames.

Sparse Window Selection

We observe that masked regions typically occupy only a small portion of the frames, whereas valid regions comprise the majority. Since these valid regions retain their original content and do not require enhancement, applying content enhancement to the entire frame results in unnecessary computational overhead. This suggests that the flow-guided soft window attention mechanism may not need to be applied to all regions. To improve efficiency, we selectively apply the attention mechanism only to the masked windows for content enhancement. Specifically, we first sum the masks M_l of neighboring frames in the temporal dimension and then split the summed mask into a grid of $m \times n$ windows. Next, we compute the sum of the mask values within the spatial domain for each window. If the sum for a window is greater than zero, it indicates the presence of masked regions, and the window is selected for content enhancement. If the sum is zero, the window contains only valid regions and can be skipped, thereby reducing unnecessary computation.

Flow Estimation

Unlike previous flow-guided approaches [51, 6, 8, 9], where optical flows are predicted for missing regions, we directly compute optical flow using initial inpainting results, eliminating the need to construct a sophisticated flow completion module. With the initial inpainting results, we directly compute the complete optical flow of the video, eliminating the need to predict flow information for missing regions. We adopt an efficient flow estimation network RAFT [134] to predict the forward and backward flows across neighboring frames, denoted as $\vec{F}_f$ and $\vec{F}_b$, respectively.

Tokenization for Content Enhancement

To facilitate flow-guided information propagation for content enhancement, we employ hybrid focal tokenization for subsequent token interaction. This approach combines a shallow 3D encoder, a linear projection layer, and a hybrid focal soft split layer. Initially, a shallow 3D encoder, consisting of four 3D convolutional layers, converts the cropped regions in reconstructed local frame group into $1/4$ sized feature maps $\mathbf{E}_{3D}$ with c_{enh} channels, ensuring effective representation of temporal dependencies in the token embeddings. Subsequently, the features map $\mathbf{E}_{3D}$ is concatenated with reconstructed feature maps E_{3D} from LGCR and fused with a linear projection layer f_{proj} as

$$\mathbf{E}_{enh} = f_{proj}(\text{Concat}(\mathbf{E}_{rec}, \mathbf{E}_{3D})), \tag{5.8}$$

where $\mathbf{E}_{enh}$ denotes the feature maps with c_{enh} channels for content enhancement. Afterwards, a hybrid focal soft split layer, similar to the tokenization process in LGCR, partitions $\mathbf{E}_{enh}$ into overlapped hybrid focal patches, transforming them into token embeddings as

$$\mathbf{Z}^0_{enh} = \text{HFSS}(\mathbf{E}_{enh}), \tag{5.9}$$

where $\mathbf{Z}^0_{enh}$ denotes the generated tokens for further flow-guided spatio-temporal token interaction.

Flow-guided Spatio-temporal Token Interaction.

It aims to enhance the quality of tokens of target regions by enabling them to interact with tokens within local spatio-temporal range for content enhancement. This process is implemented through a series of Flow-Guided Hybrid Focal Transformer Blocks (FGHFTB), which integrate Flow-Guided Soft Window Multi-head Self-Attention (FGSW-MSA) and Flow-Guided Fusion Feed-Forward Network (FGF3N).

Specifically, FGSW-MSA allows token interaction via multi-head self-attention within a flow-guided 3D spatio-temporal window, as depicted in Fig. 5.3. Given the input tokens of n^{th} FGHFTB, we firstly perform flow-based warping for motion compensation between corresponding tokens of neighboring frames as

$$\mathbf{Z}^n_{f1} = \mathcal{W}(\mathbf{Z}^n_{enh}, \vec{F}_f), \quad \mathbf{Z}^n_{b1} = \mathcal{W}(\mathbf{Z}^n_{enh}, \vec{F}_b), \tag{5.10}$$

where $\mathcal{W}$ denotes flow-based warping and $\mathbf{Z}^n_{f1}$, and $\mathbf{Z}^n_{b1}$ represents the warped tokens with forward and backward flow-based warping operations. Subsequently, we partition the token maps $\mathbf{Z}_{enh}$, $\mathbf{Z}_f$, and $\mathbf{Z}_b$ into overlapped windows using a sliding rectangular window with size P and stride Q. We then employ the attention mechanism in each window, with queries obtained from Z_{enh} while keys and values gathered from both $\mathbf{Z}_{enh}$ and aligned tokens $\mathbf{Z}_{f1}$ and $\mathbf{Z}_{b1}$. Query, key, and value embeddings are transformed from the tokens via two linear projection functions f_q and f_{kv} as

$$\mathbf{Q}^n_{enh} = f_q(\mathbf{Z}^n_{enh}), \quad \{\mathbf{K}^n_{enh}, \mathbf{V}^n_{enh}\} = f_{kv}(\{\mathbf{Z}^n_{enh}, \mathbf{Z}^n_{f1}, \mathbf{Z}^n_{b1}\}). \tag{5.11}$$

These embeddings are then used to calculate MSA following Eq. 5.4. The resulting overlapped windows are composited together with SC, yielding the result $\mathbf{Z}^n_{attn}$.

To further enhance temporal coherence for content enhancement, we introduce FGF3N to establish connections between attention layers. FGF3N comprises a Flow-Guided Deformable Warping (FGDW) layer [133, 135, 139], a MLP layer, and an HFSS layer. Initially, tokens from neighboring frames are aligned using flow-guided deformable warping as

$$\mathbf{Z}^n_{f2} = \mathrm{FGDW}(\mathbf{Z}^n_{attn}, \vec{F}_f), \quad \mathbf{Z}^n_{b2} = \mathrm{FGDW}(\mathbf{Z}^n_{attn}, \vec{F}_b), \tag{5.12}$$

where $\mathbf{Z}^n_{f2}$, $\mathbf{Z}^n_{b2}$ denote the forward and backward warping results. Subsequently, the MLP layer fuses the tokens with neighboring aligned tokens to obtain Z^n_{fuse} as

$$\mathbf{Z}^n_{fuse} = \mathrm{ReLU}(\mathrm{MLP}(\mathrm{Concat}(\mathbf{Z}^n_{attn}, \mathbf{Z}^n_{f2}, \mathbf{Z}^n_{b2}))), \tag{5.13}$$

where ReLU denotes the rectified linear unit activation function. Finally, the tokens are updated using an HFSS layer, generating new hybrid focal tokens $\mathbf{Z}^n_{enh}$ for the next FGHFTB.

The entire process of FGHFTB can be described as

$$\begin{aligned} \mathbf{Z}^n_{attn} &= \text{FGSW-MSA}(\mathrm{LN}_1(\mathbf{Z}^{n-1}_{enh})) + \mathbf{Z}^{n-1}_{enh} \\ \mathbf{Z}^n_{enh} &= \mathrm{FGF3N}(\mathrm{LN}_2(Z^n_{attn})) + \mathbf{Z}^n_{attn}, \end{aligned} \tag{5.14}$$

where $\mathbf{Z}^{n-1}_{enh}$ and $\mathbf{Z}^n_{enh}$ represent the input and output token embeddings of the n^{th} FGHFTB, respectively. By stacking N FGHFTBs to implement flow-guided spatio-temporal token interaction, the quality of tokens from corrupted regions is enhanced by incorporating neighboring reference information, thereby strengthening the local spatio-temporal dependencies and improving the visual consistency of the refined inpainting results.

Final High-resolution Frame Composition

After flow-guided spatio-temporal token interaction, we can enhance the quality of tokens by gathering cues from local spatio-temporal ranges. We adopted the same decoder structure as the one used in initial result composition to reverse tokens back to pixels, obtaining the enhanced cropped regions containing inpainting regions. We can obtain the final high-resolution inpainting results by combining the selected windows composite with the other valid regions.

5.2.5 Training Strategy

To effectively train the progressive video inpainting model, we adopt a two-stage training strategy. Initially, we train the LGCR module independently. Subsequently, with the LGCR module frozen, we proceed to train the HSCE module. Both modules are optimized by minimizing a composite loss function defined by

$$\mathcal{L}_{total} = \lambda_R \cdot \mathcal{L}_R + \lambda_{adv} \cdot \mathcal{L}_{adv}, \tag{5.15}$$

where $\mathcal{L}_R$ represents the reconstruction loss, $\mathcal{L}_{adv}$ denotes the T-PatchGAN loss [56], and λ_R and λ_{adv} are their respective weighting factors.

The reconstruction loss $\mathcal{L}_R$ uses L_1-loss to measure the difference between the reconstructed or enhanced frames $\hat{\mathbf{X}}$ and the ground-truth video $\mathbf{X}$. It is formulated as

$$\mathcal{L}_R = \|\hat{\mathbf{X}} - \mathbf{X}\|_1. \tag{5.16}$$

The T-PatchGAN loss [56] evaluates the difference between the reconstructed or enhanced result $\hat{\mathbf{X}}$ and the ground-truth video $\mathbf{X}$ using a T-PatchGAN discriminator $\mathcal{D}$, which ensures the generation of high-quality and realistic contents. The T-PatchGAN loss is formulated as

$$\mathcal{L}_{adv} = -\mathbb{E}_{\hat{\mathbf{X}}}[\mathcal{D}(\hat{\mathbf{X}})]. \tag{5.17}$$

Moreover, the T-PatchGAN discriminator learns to differentiate between real patches from ground-truth videos and fake patches from inpainted videos, thereby ensuring an accurate classification of real and fake samples with a discernible margin. The loss function employed for training the discriminator is formulated as

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{X}}(0, 1 - \mathcal{D}(\mathbf{X}))] + \mathbb{E}_{\hat{\mathbf{X}}}[\max(0, 1 + \mathcal{D}(\hat{\mathbf{X}}))]. \tag{5.18}$$

5.3 Experiments

5.3.1 Experiment Setup

Datasets

We evaluated the effectiveness of our proposed method on two widely used video object segmentation datasets, i.e. YouTube-VOS [11] and DAVIS [10]. The YouTube-VOS dataset comprises 3,471 training, 474 validation, and 508 testing video clips, covering diverse scenes. Meanwhile, the DAVIS dataset includes 60 videos in the training set and 90 videos in the test set with dynamic moving targets.

Following the original partition of YouTube-VOS dataset, we used its training set to train our model. During training process, we created both stationary irregular masks and the dynamic object-shaped masks as [55, 57, 4, 58, 59, 8] did to produce corrupted videos, making the model robust to various inpainting scenarios. To evaluate the performance of the method, we conducted evaluations on YouTube-VOS test set and 50 video clips from the test set of DAVIS dataset as the previous work [58, 59, 8] did.

Evaluation Metrics

We utilized PSNR, SSIM [138], VFID [113], and E_{warp} [114] as the quantitative metrics to evaluate the performance of various video inpainting methods. Specifically, PSNR and SSIM are widely accepted metrics for comparing the similarity between reconstructed images or videos and their originals. A higher value indicates a higher degree of similarity. VFID was employed to assess the perceptual similarity of distortion-oriented videos and has been incorporated in recent video inpainting approaches [58, 59, 8]. A lower VFID value suggests better realism and less distortion when compared to natural videos. Flow warping error E_{warp} measures the temporal consistency based on optical flow. Lower score indicates better temporal consistency.

Implementation Details

The LGCR module was constructed with 8 GDSTBs, with the channels set to $c_{rec} = 128$, $c_t = 512$ and the hybrid kernel size set to 7. For the HSCE module, we employed $m \times n = 256 \times 256$, 6 FGHFTBs with $c_{enh} = 128$ and the hybrid kernel size set to 3. We set the scaling factors for training both modules, λ_R and λ_{adv}, to 1 and 1×10^{-3}, respectively. During training, the LGCR module was trained with 8 local frames or non-local frames for 500K iterations, while HSCE was trained with 6 local frames for an equivalent number of iterations. The initial learning rate was set to 1×10^{-3} and progressively decreased to

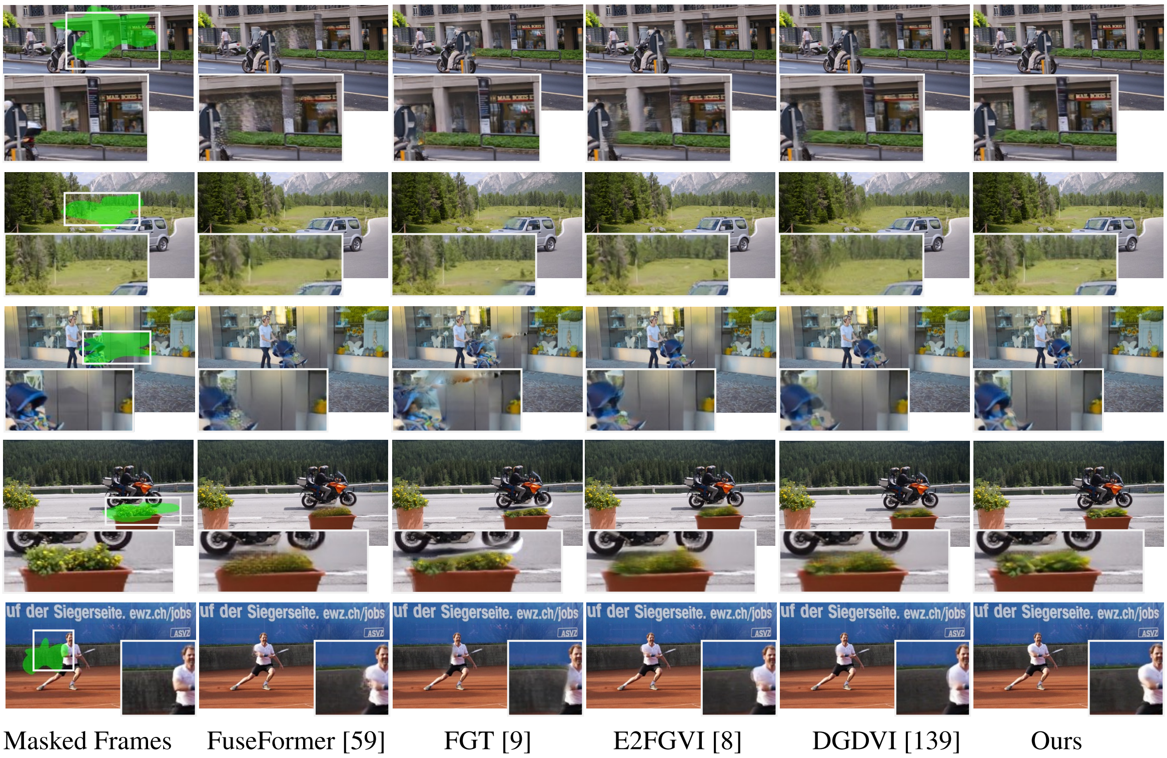


Fig. 5.4 Qualitative comparative results for video completion task. From top to bottom: *Scooter-gray*, *Car-turn*, *Stroller*, *Motorbike*, and *Tennis* videos from the DAVIS dataset.

1×10^{-4} until 400K iterations. For testing, we used 10 local frames and set the step-size to uniformly sample non-local frames to 10 as well. We conducted training using PyTorch platform on 8 NVIDIA GeForce RTX 4090 GPUs (24G). For testing, we used a single NVIDIA A100 Tensor Core GPU (80G).

5.3.2 Comparison with the State-of-the-Art

Qualitative Comparison

We conducted a qualitative comparison between our method and four state-of-the-art approaches, i.e. FGVC [6], FGT [9], E2FGVI [8], and DGDVI [139]. The comparison was performed on two tasks, video completion and object removal, using videos from the DAVIS dataset. For the video completion task, we created stationary irregular masks, following the approach of previous methods [58, 59, 61, 8]. In this task, a static mask was randomly applied to a video to remove content. Conversely, for the object removal task, we employed dynamic object-shaped masks, with each mask covering a moving object throughout the video. Fig. 5.4 and Fig. 5.5 presented some results from both video completion and object removal tasks, focusing on challenging scenes with dynamic moving objects. One could

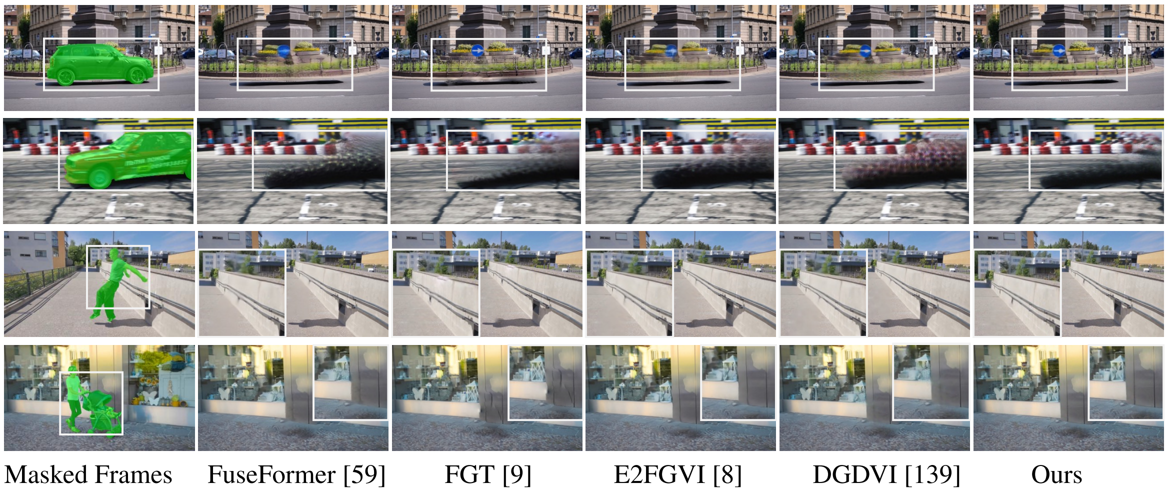


Fig. 5.5 Qualitative comparative results for object removal task. From top to bottom: *Car-roundabout*, *Drift-straight*, *Parkour*, and *Stroller* videos from the DAVIS dataset.

see from these figures that our method produced more reliable results and clearer structures compared to other approaches, showcasing its effectiveness.

Quantitative Comparison

We performed a quantitative evaluation of our proposed method through two experiments. First, we compared the performance of our method with several state-of-the-art methods on the YouTube-VOS and DAVIS datasets. The resolution of the test videos was set to 432×240, a common standard for quantitative evaluation in this field. Our method was compared with previous SOTA methods, including VINet [55], DFVI [51], LGTSM [57], CAP [4], STTN [58], FGVC [6], FuseFormer [59], FGT [9], E2FGVI [8], and DGDVI [139]. The results were provided in Table 5.1. As shown in Table 5.1, our method significantly outperformed all the state-of-the-art methods in terms of the four quantitative metrics on the DAVIS dataset. While we observed a minor gap with DGDVI on SSIM, our approach still produced impressive results on the YouTube-VOS dataset. These results demonstrated that our method was capable of recovering content with less distortion (PSNR and SSIM), more visually faithful reconstructions (VFID), and better spatial and temporal consistency (E_{warp}).

Additionally, we evaluated the performance of our video inpainting approach across various resolutions on the DAVIS dataset. The tested resolutions ranged from low to super high, including 240p (432×240), 480p (864×480), 720p (1280×720), 1080p (1944×1080), and 2K (2160×1200). We compared our method with several recent state-of-the-art methods that have demonstrated strong performance at 240p, including STTN [58], FuseFormer [59], E2FGVI [8], FGT [9], and DGDVI [139]. The results were presented in Table 5.2. Most of

Table 5.1 Quantitative comparisons on YouTube-VOS [11] and DAVIS [10] datasets. ↑ indicates higher is better. ↓ indicates lower is better. $E_{warp}{}^*$ denotes $E_{warp} \times 10^{-2}$. Each method is evaluated following the procedures in FuseFormer. VINet, DFVI, FGVC, and FGT are not end-to-end training methods. Their FLOPs, thus, are not presented. SAVIT did not provide the source code. As such its FLOPs and runtime are not provided. The best results for each metric are highlighted in bold.

Methods	YouTube-VOS				DAVIS				FLOPs	Runtime
	PSNR (dB) ↑	SSIM ↑	VFID ↓	$E_{warp}{}^*$ ↓	PSNR (dB) ↑	SSIM ↑	VFID ↓	$E_{warp}{}^*$ ↓		(s/frame)
VINet [55]	29.20	0.9434	0.072	0.1490	28.96	0.9411	0.199	0.1785	-	-
DFVI [51]	29.16	0.9429	0.066	0.1509	28.81	0.9404	0.187	0.1608	-	2.56
LGTSM [57]	29.74	0.9504	0.070	0.1859	28.57	0.9409	0.170	0.1640	1008G	0.23
CAP [4]	31.58	0.9607	0.071	0.1470	30.28	0.9521	0.182	0.1533	861G	0.40
FGVC [6]	29.67	0.9403	0.064	0.1022	30.80	0.9497	0.165	0.1586	-	2.36
STTN [58]	32.34	0.9655	0.053	0.0907	30.67	0.9560	0.149	0.1449	1032G	0.12
FuseFormer [59]	33.29	0.9681	0.053	0.0900	32.54	0.9700	0.138	0.1362	752G	0.20
FGT [9]	30.19	0.9536	0.063	0.0968	31.77	0.9639	0.134	0.1483	-	1.89
E2FGVI [8]	33.71	0.9700	0.046	0.0864	33.01	0.9721	0.116	0.1315	682G	0.16
DGDVI [139]	34.07	**0.9725**	0.045	0.0823	33.33	0.9740	0.111	0.1295	860G	0.21
HSTVI (Ours)	**34.30**	0.9721	**0.044**	**0.0396**	**34.52**	**0.9768**	**0.103**	**0.483**	798G	0.19

the evaluated methods struggled to handle higher-resolution videos, revealing limitations in video inpainting at high resolutions. STTN, FuseFormer, and DGDVI were restricted to processing videos at 240p due to the limitations of absolute positional encoding in their algorithm designs, while E2FGVI and FGT were unable to manage some higher resolutions due to memory constraints, even with 80GB of memory on a NVIDIA A100 GPU. However, under the same memory constraints, our method successfully processed all resolutions and consistently outperforms other methods across all metrics, demonstrating both the robustness and efficiency of our proposed approach.

Complexity Analysis

We used FLOPs and inference time to evaluate the complexity of the compared methods by using the DAVIS dataset. The corresponding results were presented in Table 5.1. The FLOPs of our proposed approach were comparable to VINet [55], LGTSM [57] and CAP [4] that were developed based on CNN. Meanwhile, the proposed method executed about ×10 faster than DFVI [51], FGVC [6] and FGT [9]. Meanwhile, our method achieved comparable speeds to the Transformer-based approaches, such as STTN [58], FuseFormer [59], E2FGVI [8] and DGDVI [139].

Table 5.2 Quantitative comparisons for videos at different resolutions from the DAVIS [10] dataset. The test video resolutions range from low (240p and 480p) to high and super resolutions (720p, 1080p, and 2K). A “-” indicates no available results. STTN, FuseFormer, and DGDVI only process videos at 240p resolution, while E2FGVI and FGT fail to handle some higher resolutions due to memory limitations in their algorithm design. The best results for each metric are highlighted in bold.

Methods	**240p (432*240)**			**480p (864*480)**			**720p (1296*720)**			**1080p (1944*1080)**			**2K (2160*1200)**		
	PSNR	SSIM	VFID	PSNR	SSIM	VFID	PSNR	SSIM	VFID	PSNR	SSIM	VFID	PSNR	SSIM	VFID
STTN [58]	30.67	0.956	0.149	-	-	-	-	-	-	-	-	-	-	-	-
FuseFormer [59]	32.54	0.97	0.138	-	-	-	-	-	-	-	-	-	-	-	-
E2FGVI [8]	33.01	0.9721	0.116	32.88	0.969	0.042	32.32	0.9637	**0.019**	30.44	0.9565	0.014	-	-	-
FGT [9]	31.76	0.9639	0.134	32.75	0.97	**0.04**	-	-	-	-	-	-	-	-	-
DGDVI [139]	33.33	0.9740	0.111	-	-	-	-	-	-	-	-	-	-	-	-
HSTVI (Ours)	**34.52**	**0.9768**	**0.103**	**33.89**	**0.9745**	0.043	**33.88**	**0.9725**	**0.019**	**33.67**	**0.9692**	**0.013**	**33.61**	**0.9682**	**0.010**

Table 5.3 Ablation study for the components in HSTVI. The results produced by the single LGCR module and the combination of LGCR and HSCE modules are evaluated on the DAVIS [10] dataset. Performance is measured using PSNR and SSIM metrics, with higher values indicating better reconstruction quality. The best results for each metric are highlighted in bold.

	PSNR (dB) ↑	SSIM ↑	VFID ↓
LGCR	33.26	0.9731	0.111
LGCR + HSCE	**34.52**	**0.9768**	**0.103**

5.3.3 Ablation Studies

In this subsection, we conducted a series of ablation experiments to verify the effectiveness of the proposed modules, hybrid focal kernel-based tokenization and the adoption of non-local frames in our model. All the studies were performed on the DAVIS dataset for the video completion task.

Effectiveness of the Components in HSTVI

Our proposed inpainting model, HSTVI, was constructed with two modules, LGCR and HSCE. To demonstrate the performance gain offered by these modules, we conducted an ablation study to verify their effectiveness. In the experiment, we always retained the LGCR module, as it is the basic component in our model. Hence, the ablation study compared the performance of solely using the LGCR module with incorporating both the LGCR and HSCE modules. The corresponding quantitative and qualitative results were illustrated in Table 5.3. It was observed from Table 5.3 that introducing the content enhancement module

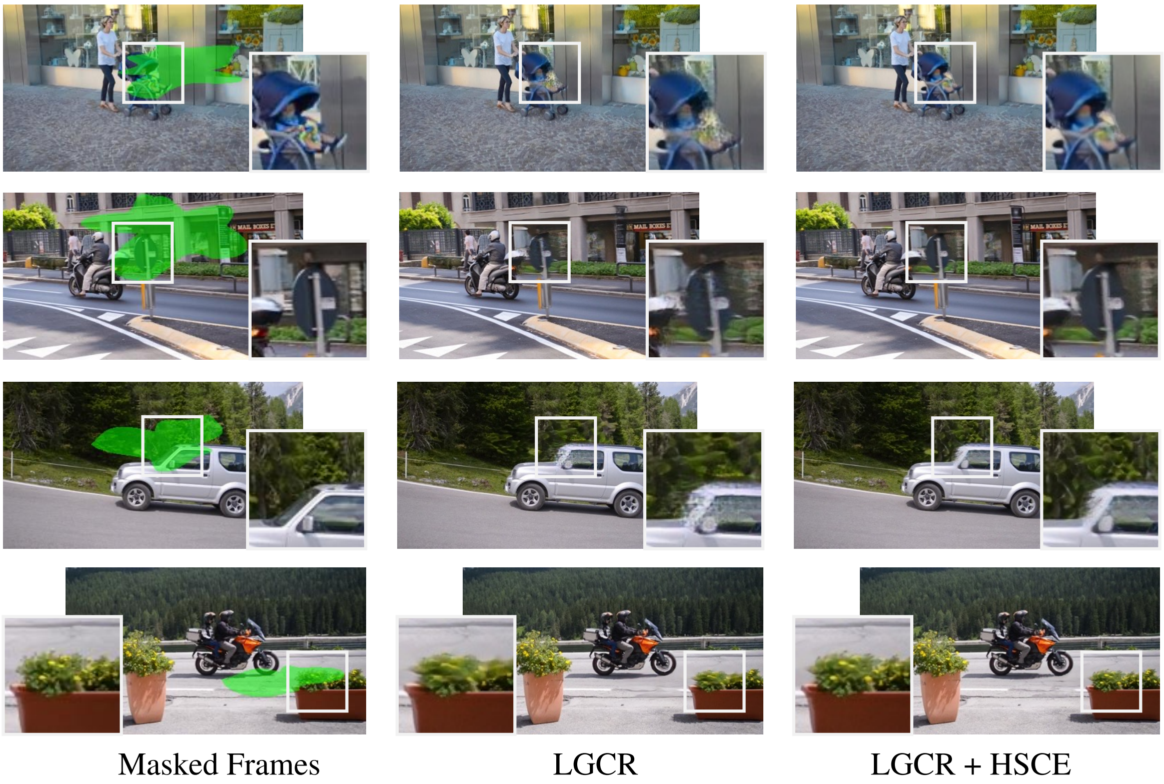


Fig. 5.6 Ablation study for the components in HSTVI. We qualitatively compare the ground truth, initial inpainting results produced by the single LGCR module, and final results generated by the combination of LGCR and HSCE modules. From top to bottom: *Stroller*, *Scooter-gray*, *Car-turn*, and *Motorbike* videos from the DAVIS dataset.

brought improvements of nearly 1.4 dB in PSNR, about 0.04 in SSIM and approximately 0.01 in VFID. Furthermore, it was evident from Fig. 5.6 that the model with the content enhancement module produced results with finer structures and textures. This demonstrated that utilizing the progressive video inpainting framework could effectively enhance local details and improve video perceptual quality.

Effectiveness of Hybrid Focal Kernel for Tokenization

To assess the effectiveness of employing a hybrid focal kernel for tokenization in our proposed method, we conducted an experiment comparing the model's performance using different kernels for tokenization, including vanilla kernel, deformable kernel, and hybrid focal kernel. We replaced all the operations in LGCR and HSCE that used focal kernels with these two kernels, respectively. The corresponding quantitative results were presented in Table. 5.4. One could see from Table. 5.4 that utilizing hybrid focal kernel-based tokenization could significantly enhance the performance for both content reconstruction and content enhancement in the proposed progressive inpainting approach.

Table 5.4 Comparison of different kernels used for tokenization in the HSTVI model. The vanilla kernel, deformable kernel, and hybrid focal kernel are separately applied to the tokenization process in the LGCR and HSCE modules and evaluated on the DAVIS [10] dataset. Performance is measured using PSNR, SSIM, and VFID metrics, with the best results for each metric highlighted in bold.

	LGCR			HSCE		
	PSNR (dB) ↑	SSIM ↑	VFID ↓	PSNR (dB) ↑	SSIM ↑	VFID ↓
vanilla kernel	32.54	0.9700	0.136	33.38	0.9721	0.117
deformable kernel	32.81	0.9712	0.121	33.73	0.9740	0.111
hybrid focal kernel	**33.26**	**0.9731**	**0.111**	**34.52**	**0.9768**	**0.103**

Table 5.5 Ablation study on the utilization of non-local frames in the HSTVI model. Local frames and a combination of local and non-local frames are separately applied to our proposed HSTVI model and evaluated on the DAVIS [10] dataset. Performance is assessed using PSNR, SSIM, and VFID metrics, with the best results for each metric highlighted in bold.

	w/o non-local frames		w/ non-local frames	
	LGCR	LGCR + HSCE	LGCR	LGCR + HSCE
PSNR (dB) ↑	32.67	33.67	33.26	**34.52**
SSIM ↑	0.9690	0.9720	0.9731	**0.9768**
VFID ↓	0.121	0.113	0.111	**0.103**

Effectiveness of Adopting Non-local Frames

To validate the effectiveness of incorporating non-local frames in video inpainting, we conducted a comparative experiment between using only local frames and integrating both local and non-local frames for inpainting videos. The non-local frames were sampled linearly from the entire video sequence. The corresponding quantitative and qualitative results were presented in Table. 5.5. These results revealed that the adoption of non-local frames improved the performance for both content reconstruction and content enhancement stages. Hence, this indicated its significance in overall inpainting efficacy.

5.4 Summary

In this paper, we introduced a hierarchical sparse Transformer (HSTVI) for high-resolution video inpainting. Our method generates coherent and visually plausible results in a coarse-to-fine manner through two key modules: LGCR and HSCE. Specifically, the LGCR module, constructed with spatio-temporal Transformer, leverages global spatio-temporal reference information from sparsely sampled frames to produce an initial low-resolution inpainting result, while the HSCE module, built upon a flow-guided sparse Transformer, refines texture details and enhances visual consistency for high-resolution outputs by focusing on local spatio-temporal dependencies. Moreover, we proposed a hybrid focal kernel to extract token embeddings, which can effectively capture features from both local and non-local ranges of contexts and be applied to both LGCR and HSCE.

We evaluated the proposed HSTVI on the DAVIS and YouTube-VOS datasets across a range of resolutions, from 240p to 2K videos, for both object removal and video restoration tasks. Experimental results demonstrated that our method effectively handles videos across various resolutions, maintaining both computational efficiency and high-quality outputs. Furthermore, we compared our approach with several SOTA methods [55, 51, 57, 4, 58, 59, 6, 9, 8]. The results show that HSTVI not only processes a broader range of high-resolution videos but also consistently outperforms SOTA methods based on quantitative and qualitative evaluations, underscoring its robustness and effectiveness.

6

Conclusions and Future Expectations

6.1 Conclusions

In this Ph.D. thesis, we focus on the development of AI-enabled video inpainting techniques using spatio-temporal correlations. With a deep discussion of the significance of video inpainting and a comprehensive literature review for the related works, our research explored starting from three key issues within video inpainting, including visual and contextual inconsistency, difficulties in complex scenes with multiple layers, and high computational demand. To address these challenges, we proposed the corresponding approaches. The contributions of my research can be summarized as follows.

- We developed a short-long-term propagation-based video inpainting approach for object removal. This method integrates short-term and long-term propagation mechanisms, ensuring seamless and coherent inpainting results by progressively propagating reference information within the video.

- We constructed a depth-guided deep video inpainting network, which utilizes depth information to guide the inpainting of multiple layers within complex scenes. This approach effectively addresses the artifacts due to a lack of understanding in layer relationships, providing more accurate and reliable inpainted content.

- We designed a hierarchical video inpainting approach for high-resolution videos. By separating the inpainting process into low-resolution contextual reconstruction and high-resolution texture enhancement, this method efficiently manages computation and memory demand, enabling the inpainting of high-resolution videos without scarifying quality.

The experiments showed that our proposed methods outperform the SOTA for their corresponding targets, respectively. Therefore, our contributions may provide potential solutions for future developments in video inpainting and offer practical technical support for various applications in multimedia, art, historical preservation, medical healthcare, and beyond.

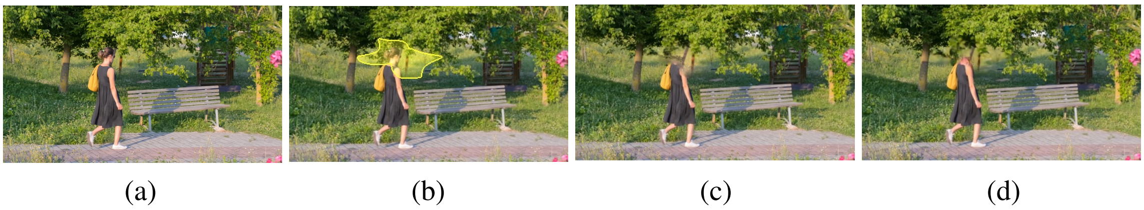


Fig. 6.1 An example of limited cases for uncertain scenes. From left to right: (a) original video, (b) masked video, (c) result of SLTPVI, (d) result of DGDVI.

6.2 Limitations and Discussions

Despite addressing several key issues in video inpainting, our research has limitations that present opportunities for future improvement.

Limitations in Understanding User's Intention

Since video inpainting is an inherently ill-posed problem with no unique solution, AI-enabled video inpainting algorithms occasionally fail to generate content aligned with user expectations in uncertain cases. For example, consider a scenario where a girl's head was masked in a video, as shown in Fig. 6.1. Our approaches generated headless walking figures. While these results might be appropriate for an object-removal task in special effects, they were unsuitable for a video restoration task aimed at recovering the original appearance or generating a plausible new face. Incorporating user intentions into the inpainting process offers a promising solution to this issue. By enabling users to provide textual guidance or instructions about their intentions, integrating video inpainting with NLP, such as Contrastive Language-Image Pre-training (CLIP) [141], could facilitate more accurate and reliable conditional editing and content generation.

Limitations in Real-time Computational Efficiency

While the proposed methods have improved computational efficiency, our approach currently achieves only 5 fps on an NVIDIA 3090Ti GPU, falling short of real-time processing requirements. This limitation underscores the need for further algorithmic optimization to enhance processing speed. Future work could focus on lightweight model architectures and efficient computational strategies, such as leveraging video coding information. Instead of estimating optical flow across consecutive frames, we may use motion vectors from video compression frameworks, such as Advanced Video Coding (AVC) [142], High Efficiency Video Coding (HEVC) [143], and Versatile Video Coding (VVC) [144], to build corresponding relationships in temporal dimensions, which helps to propagate reference information to the target regions. If the video inpainting module could be integrated into the video compression framework, the inpainting of corrupted regions can be implemented simultaneously with video encoding/decoding process. It will enable faster results, paving the way for real-time applications.

Limitations in Advanced Quality Assessment

One limitation in evaluating the performance of video inpainting methods is the lack of subjective analysis of high-level visual information. An important direction for future work is the incorporation of advanced Video Quality Assessment (VQA) tools to provide more comprehensive evaluations of video inpainting methods. In this regard, the Video Multi-method Assessment Fusion (VMAF) [145, 146], proposed by Netflix, represents a promising avenue for assessing the quality of inpainted videos. VMAF combines several video quality metrics, including perceptual and structural characteristics, to predict human perception of video quality more accurately. Unlike traditional metrics such as PSNR or SSIM, which primarily focus on pixel-level differences, VMAF considers higher-level visual factors such as texture and motion, making it particularly relevant for evaluating the subjective quality of reconstructed videos.

By applying VMAF in future studies, we can obtain a better understanding of how different inpainting algorithms affect video quality from the viewer's perspective. This tool could be used to benchmark inpainting methods, offering deeper insights into their strengths and limitations in terms of perceptual quality. Moreover, integrating VMAF into the optimization pipeline could enhance algorithm design by providing more perceptually aligned feedback during training. Moreover, incorporating VMAF into the evaluation and development of video inpainting methods can improve their robustness and align their outputs more closely with human expectations in real-world applications.

6.3 Future Research Directions

Looking ahead, video inpainting and editing fields will witness transformative developments in the next 5–10 years. The advent of large generative models, multi-modality learning, and responsible AI will reshape image and video edition tasks and bring opportunities to other disciplines, especially medical and health applications.

For example, by integrating NLP and large generative models, video inpainting systems could allow users to specify their desired edits through textual prompts, enabling intuitive and precise control over the content generation process. Additionally, advances in multi-modal learning may lead to systems capable of simultaneously processing video, audio, and textual data, providing comprehensive tools for complex multimedia editing tasks. However, ensuring the reliability and interpretability of AI-generated content is crucial, especially in sensitive domains such as medical imaging and legal forensics. Robust validation frameworks and explainable AI techniques will be essential to build trust in AI-driven editing tools. Moreover, ethical concerns regarding privacy and intellectual property must be carefully addressed. As video inpainting systems become more powerful, they may inadvertently enable misuse, such as creating deepfakes or altering videos in deceptive ways. Developing regulatory frameworks and implementing safeguards against misuse will be vital to mitigate these risks. In the following parts, we will provide more detailed discussions from these aspects.

Medical and Healthcare Applications

In the medical imaging and diagnostics field, video inpainting has significant potential to enhance the quality of recorded procedures and diagnostic videos, ensuring that critical details are preserved and clearly visible for educational and training purposes. Additionally, it can improve communication efficiency by removing artifacts that may interfere with remote diagnostics. For instance, generative models can enhance medical imaging by reconstructing missing or low-quality data, producing high-resolution images from noisy or incomplete scans, and even simulating diverse pathological scenarios to improve diagnostic accuracy. In radiology, they could assist in detecting subtle abnormalities in X-ray [147, 148] or Magnetic Resonance Imaging (MRI) [149–151] data, thereby reducing the likelihood of missed diagnoses. Furthermore, generative models can synthesize realistic training data [152, 153], mitigating the challenges of limited datasets in medical research while preserving patient privacy through the creation of anonymized, synthetic medical data.

As a future work direction, a specialized medical image dataset can be constructed for video inpainting, focusing on improving the recording quality of procedures such as endoscopy [17, 18]. Video inpainting technologies can be applied to remove specular highlights or medical instruments, producing better views without obstructions. Moreover, to address privacy concerns associated with medical data, we suggest integrating federated learning [154, 155] with video inpainting techniques. This integration would allow for a secure, smart medical imaging system that enables collaborative learning across multiple institutions without compromising patient privacy. Federated learning would facilitate the training of robust video inpainting models on diverse datasets while keeping sensitive patient information decentralized and protected. This combined approach could revolutionize the field by providing high-quality, secure video inpainting solutions tailored to the unique requirements of medical and healthcare applications.

Diffusion Models

Diffusion models [156] has emerged as a powerful framework for image and video generation, rivaling or surpassing earlier generative approaches such as GAN [157] and Variational Auto-Encoder (VAE) [158]. Diffusion models, including popular implementations like DALL-E 2 [159] and Stable Diffusion [160], operate by progressively refining noise through a learned reverse diffusion process. This iterative framework has shown exceptional capability in generating high-fidelity, diverse, and coherent visual content.

In the context of video inpainting and editing, diffusion models have great potential in generating temporally consistent and spatially coherent inpainting results. By incorporating spatio-temporal priors and noise scheduling tailored for video data, these models can address common challenges like flickering and motion discontinuities. Their probabilistic foundation allows for multi-modal outputs, enabling the generation of plausible variations for missing regions or edits based on textual or other contextual prompts. For example, diffusion models can restore missing frames in damaged videos or even generate entirely new sequences that align seamlessly with the existing footage. Additionally, conditional generation, a strong advantage of diffusion models, will play an important role in future image and video inpainting. By conditioning the diffusion process on external inputs, such as masks, optical flow, or depth maps, these models can accurately propagate structural and motion cues throughout the content. This makes diffusion-based approaches particularly effective in scenarios involving complex edits, such as object removal, style transfer, or video restoration.

Multi-Modality Learning

In recent years, multi-modality learning [161, 162] represents another groundbreaking advancement in image and video generation, enabling models to understand and synthesize content across multiple data types, such as text, images, audio, and video. Models like CLIP [141] and Flamingo [163] have shown how vision-language pre-training can significantly improve the alignment between textual and visual domains, resulting in more intuitive and flexible editing capabilities.

In image and video inpainting applications, multi-modality learning can be used to facilitate tasks such as text-guided image and video editing, where users can describe desired changes in natural language. For instance, a user can input "remove the car and replace it with a tree" or "change the sky to a sunset" to achieve complex scene modifications without requiring technical expertise. The synergy between multi-modality models and diffusion frameworks has been particularly transformative, as demonstrated by tools like Adobe Firefly and Runway ML, which leverage this combination for user-friendly creative workflows.

Reliability of Generated Results

The rise of generative models has brought transformative potential to image and video , but it also necessitates careful consideration of responsibility, reliability, and ethics. Ensuring the trustworthiness of AI-generated content is paramount, particularly in sensitive domains such as medical imaging, transportation, and education, where the stakes are high, and errors can have serious consequences.

Reliability in AI-generated content [164, 165] involves producing accurate, consistent, and dependable outputs that align with real-world requirements. For instance, in medical imaging [149–151], reconstructed data directly impacts patient health and clinical decisions. Misleading or incorrect reconstructions could lead to misdiagnosis or suboptimal treatment plans. Similarly, in autonomous driving systems [166, 167], generative models used for enhancing street maps or filling in occlusions in sensor data must produce flawless outputs, as any error could jeopardize the safety of drivers and pedestrians. In education [168], the quality and accuracy of AI-generated content influence the knowledge and understanding of students, underscoring the need for highly reliable systems. To achieve this, robust validation frameworks are critical. When generative models are applied for image and video inpainting tasks in real applications, especially for medical and health applications [147, 148, 17, 18]. These frameworks involve rigorous testing of models under diverse scenarios to ensure consistent and accurate performance. Additionally, explainable AI techniques play a vital role in building trust. By providing insights into how generative models make decisions or generate content, explainable AI can help end-users understand the system's reliability and limitations.

Responsible Generative Model

Ethical questions are raised with the development of image and video editions techniques and generative models, particularly concerning privacy, intellectual property, and misuse. Hence, it is crucial to develop responsible generative models for inpainting tasks in real-world applications. The generative models with responsibility could inadvertently enable harmful applications, such as creating deepfakes or altering videos in deceptive ways. Such misuse has far-reaching implications, from spreading misinformation to violating personal or corporate integrity. To prevent these issues, the technologies [169, 170] to detect inpainted or generative contents will be further developed, protecting the public data security. Additionally, privacy is another critical issue. Even with regulations like Generation Data Protection Regulation (GDPR) and personal data protection laws, generative models can sometimes reconstruct private information based on seemingly innocuous input data. For example, a system trained on anonymized medical images might inadvertently regenerate identifiable details, posing risks to patient confidentiality. Moreover, generative models often require significant computational resources, leading users to rely on cloud-based services. This reliance raises concerns about data transmission and storage, as sensitive personal information could be exposed during processing. Ensuring secure encryption, data minimization, and on-device generative capabilities are some potential approaches to mitigate these risks.

References

[1] Z. Wan, B. Zhang, D. Chen, and J. Liao, “Bringing old films back to life,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 17694–17703, 2022.

[2] W. Yan, L. Xu, W. Yang, and R. T. Tan, “Feature-aligned video raindrop removal with temporal constraints,” *IEEE Trans. Image Process.*, vol. 31, pp. 3440–3448, 2022.

[3] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Temporally coherent completion of dynamic video,” *ACM Trans. Graph.*, 2016.

[4] S. Lee, S. W. Oh, D. Won, and S. J. Kim, “Copy-and-paste networks for deep video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[5] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim, “Onion-peel networks for deep video completion,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[6] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf, “Flow-edge guided video completion,” in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[7] H. Ouyang, T. Wang, and Q. Chen, “Internal video inpainting by implicit long-range propagation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 14579–14588, 2021.

[8] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng, “Towards an end-to-end framework for flow-guided video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 17562–17571, 2022.

[9] K. Zhang, J. Fu, and D. Liu, “Flow-guided transformer for video inpainting,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 74–90, 2022.

[10] 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 *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016.

[11] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, “Youtube-vos: Sequence-to-sequence video object segmentation,” in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[12] Z. Ji, Y. Su, Y. Zhang, J. Hou, Y. Pang, and J. Han, “Raformer: Redundancy-aware transformer for video wire inpainting,” *arXiv preprint arXiv:2404.15802*, 2024.

[13] S. Tsogkas, F. Zhang, A. Jepson, and A. Levinshtein, “Efficient flow-guided multi-frame de-fencing,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, pp. 1838–1847, 2023.

[14] M. Liao, F. Lu, D. Zhou, S. Zhang, W. Li, and R. Yang, “Dvi: Depth guided video inpainting for autonomous driving,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 1–17, 2020.

[15] A. Renaudeau, F. Lauze, F. Pierre, J. Aujol, and J. Durou, “Alternate structural-textural video inpainting for spot defects correction in movies,” in *Scale Space Var. Methods Comput. Vis.*, vol. 11603, pp. 104–116, Springer, 2019.

[16] A. Renaudeau, T. Seng, A. Carlier, F. Pierre, F. Lauze, J.-F. Aujol, and J.-D. Durou, "Learning defects in old movies from manually assisted restoration," in *Proc. IEEE Int. Conf. Pattern Recognit.*, pp. 5254–5261, IEEE, 2021.

[17] Q. Wang, Y. Chen, N. Zhang, and Y. Gu, "Medical image inpainting with edge and structure priors," *Meas.*, vol. 185, p. 110027, 2021.

[18] F. X. Zhang, S. Chen, X. Xie, and H. P. Shum, "Depth-aware endoscopic video inpainting," *arXiv preprint arXiv:2407.02675*, 2024.

[19] Y. Liu, S. Dutta, A. W. K. Kong, and C. K. Yeo, "An image inpainting approach to short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 38, no. 1, pp. 177–187, 2022.

[20] A. Geiss and J. C. Hardin, "Inpainting radar missing data regions with deep learning," *Atmos. Meas. Tech.*, vol. 14, no. 12, pp. 7729–7747, 2021.

[21] A. Bakushinsky and A. Goncharsky, *Ill-posed problems: theory and applications*, vol. 301. Springer Science & Business Media, 2012.

[22] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, 2013.

[23] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, pp. 417–424, 2000.

[24] J. Shen and T. F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2002.

[25] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.

[26] D. Jin and X. Bai, "Patch-sparsity-based image inpainting through a facet deduced directional derivative," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1310–1324, 2018.

[27] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 5505–5514, 2018.

[28] H. Liu, B. Jiang, Y. Xiao, and C. Yang, "Coherent semantic attention for image inpainting," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4170–4179, 2019.

[29] C. Xie, S. Liu, C. Li, M.-M. Cheng, W. Zuo, X. Liu, S. Wen, and E. Ding, "Image inpainting with learnable bidirectional attention maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 8858–8867, 2019.

[30] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 7508–7517, 2020.

[31] 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 *Proc. Eur. Conf. Comput. Vis.*, pp. 85–100, 2018.

[32] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4471–4480, 2019.

[33] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, “Resolution-robust large mask inpainting with fourier convolutions,” in *Proc. IEEE Winter Conf. App. Comput. Vis.*, pp. 2149–2159, 2022.

[34] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, “Edgeconnect: Generative image inpainting with adversarial edge learning,” *arXiv preprint arXiv:1901.00212*, 2019.

[35] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo, “Foreground-aware image inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 5840–5848, 2019.

[36] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, “Structureflow: Image inpainting via structure-aware appearance flow,” in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 181–190, 2019.

[37] H. Wu, J. Zhou, and Y. Li, “Deep generative model for image inpainting with local binary pattern learning and spatial attention,” *IEEE Trans. Multimedia*, vol. 24, pp. 4016–4027, 2021.

[38] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, “Spg-net: Segmentation prediction and guidance network for image inpainting,” in *Proc. Brit. Mach. Vis. Conf.*, p. 97, 2018.

[39] Y. Zhang, Y. Liu, R. Hu, Q. Wu, and J. Zhang, “Mutual dual-task generator with adaptive attention fusion for image inpainting,” *IEEE Trans. Multimedia*, vol. 26, pp. 1539–1550, 2023.

[40] H. Sun, W. Li, Y. Duan, J. Zhou, and J. Lu, “Learning adaptive patch generators for mask-robust image inpainting,” *IEEE Trans. Multimedia*, 2022.

[41] Y. Yu, F. Zhan, R. Wu, J. Pan, K. Cui, S. Lu, F. Ma, X. Xie, and C. Miao, “Diverse image inpainting with bidirectional and autoregressive transformers,” in *ACM Int. Conf. Multimedia*, pp. 69–78, 2021.

[42] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, “Mat: Mask-aware transformer for large hole image inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 10758–10768, 2022.

[43] Q. Dong, C. Cao, and Y. Fu, “Incremental transformer structure enhanced image inpainting with masking positional encoding,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 11358–11368, 2022.

[44] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 2536–2544, 2016.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[46] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, “Video inpainting of occluding and occluded objects,” in *Proc. IEEE Int. Conf. Inf. Process.*, vol. 2, pp. II–69, IEEE, 2005.

[47] Y. Wexler, E. Shechtman, and M. Irani, “Space-time completion of video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, 2007.

[48] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, “Video inpainting of complex scenes,” *SIAM J. Imaging Sci.*, 2014.

[49] T. T. Le, A. Almansa, Y. Gousseau, and S. Masnou, “Motion-consistent video inpainting,” in *Proc. IEEE Int. Conf. Inf. Process.*, pp. 2094–2098, IEEE, 2017.

[50] A. Aldahdooh, M. Barkowsky, D. R. Bull, and P. Le Callet, “Inpainting-based error concealment for low-delay video communication,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 1632–1636, IEEE, 2017.

[51] R. Xu, X. Li, B. Zhou, and C. C. Loy, “Deep flow-guided video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019.

[52] X. Zou, L. Yang, D. Liu, and Y. J. Lee, “Progressive temporal feature alignment network for video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2021.

[53] K. Zhang, J. Fu, and D. Liu, “Inertia-guided flow completion and style fusion for video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 5982–5991, 2022.

[54] J. Kang, S. W. Oh, and S. J. Kim, “Error compensation framework for flow-guided video inpainting,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 375–390, Springer, 2022.

[55] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, “Deep video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019.

[56] Y. Chang, Z. Y. Liu, K. Lee, and W. Hsu, “Free-form video inpainting with 3d gated convolution and temporal patchgan,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[57] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, “Learnable gated temporal shift module for deep video inpainting,” in *Proc. Brit. Mach. Vis. Conf.*, 2019.

[58] Y. Zeng, J. Fu, and H. Chao, “Learning joint spatial-temporal transformations for video inpainting,” in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[59] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, “Fuseformer: Fusing fine-grained information in transformers for video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021.

[60] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, and L. Hongsheng, "Decoupled spatial-temporal transformer for video inpainting," *arXiv preprint arXiv:2104.06637*, 2021.

[61] M. S. Junayed and M. B. Islam, "Consistent video inpainting using axial attention-based style transformer," *IEEE Trans. Multimedia*, 2022.

[62] E. Lee, J. Yoo, Y. Yang, S. Baik, and T. H. Kim, "Semantic-aware dynamic parameter for video inpainting transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 12949–12958, 2023.

[63] S. Zhou, C. Li, K. C. Chan, and C. C. Loy, "Propainter: Improving propagation and transformer for video inpainting," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 10477–10486, 2023.

[64] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," in *Proc. AAAI Conf. Artif. Intell.*, 2019.

[65] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[66] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[67] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, pp. 1150–1157, IEEE, 1999.

[68] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, pp. 404–417, Springer, 2006.

[69] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," *Int. Conf. Comput. Vis. Theory Appl.*, vol. 2, no. 331-340, p. 2, 2009.

[70] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[71] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A practical approach to microarray data analysis*, pp. 91–109, Springer, 2003.

[72] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis: proceedings of the biennial Conference*, pp. 105–116, Springer, 2006.

[73] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 2339–2346, 2013.

[74] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1-3, pp. 185–203, 1981.

[75] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. Artif. Intell.*, vol. 2, pp. 674–679, 1981.

[76] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2758–2766, 2015.

[77] 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 *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 2462–2470, 2017.

[78] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, pp. 402–419, Springer, 2020.

[79] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[80] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 807–814, IEEE, 2005.

[81] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2007.

[82] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[83] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[84] R. Ji, K. Li, Y. Wang, X. Sun, F. Guo, X. Guo, Y. Wu, F. Huang, and J. Luo, "Semi-supervised adversarial monocular depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2410–2422, 2019.

[85] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 6647–6655, 2017.

[86] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon, "Robust semi-supervised monocular depth estimation with reprojected distances," in *Conf. Robot Learn.*, pp. 503–512, PMLR, 2020.

[87] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 270–279, 2017.

[88] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.*, pp. 740–756, Springer, 2016.

[89] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 340–349, 2018.

[90] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, 1959.

[91] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, pp. 541–551, 1989.

[92] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[93] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[94] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. C. Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019.

[95] S. Lee, S. W. Oh, D. Won, and S. J. Kim, "Copy-and-paste networks for deep video inpainting," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 4413–4421, 2019.

[96] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 5998–6008, 2017.

[97] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 10012–10022, 2021.

[98] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 22–31, 2021.

[99] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 10231–10241, 2021.

[100] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.

[101] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 7262–7272, 2021.

[102] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 12077–12090, 2021.

[103] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021.

[104] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu, "Vit-yolo: Transformer-based yolo for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2799–2808, 2021.

[105] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, "Transformer for single image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 457–466, 2022.

[106] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 5728–5739, 2022.

[107] X. Xu, R. Wang, C.-W. Fu, and J. Jia, "Snr-aware low-light image enhancement," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 17714–17724, 2022.

[108] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[109] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2223–2232, 2017.

[110] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 4401–4410, 2019.

[111] B. Jähne, *Digital image processing*. Springer Science & Business Media, 2005.

[112] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 586–595, 2018.

[113] T. Wang, M. Liu, J. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1152–1164, 2018.

[114] W. Lai, J. Huang, O. Wang, E. Shechtman, E. Yumer, and M. Yang, "Learning blind video temporal consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[115] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[116] 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," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[117] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017.

[118] M. Ebdelli, O. Le Meur, and C. Guillemot, “Video inpainting with short-term windows: application to object removal and error concealment,” *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3034–3047, 2015.

[119] Y.-L. Chang, Z. Yu Liu, and W. Hsu, “Vornet: Spatio-temporally consistent video inpainting for object removal,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, pp. 0–0, 2019.

[120] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-stokes, fluid dynamics, and image and video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. I–I, IEEE, 2001.

[121] T. K. Shih, N. C. Tang, and J.-N. Hwang, “Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 347–360, 2009.

[122] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[123] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[124] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, “Content-preserving warps for 3d video stabilization,” *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–9, 2009.

[125] S. Liu, L. Yuan, P. Tan, and J. Sun, “Bundled camera paths for video stabilization,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, 2013.

[126] F. Zhang and F. Liu, “Parallax-tolerant image stitching,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 3262–3269, 2014.

[127] M. Fischer, “Randam sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, pp. 381–385, 1981.

[128] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[129] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[130] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median,” *J. Exp. Soc. Psychol.*, vol. 49, no. 4, pp. 764–766, 2013.

[131] Y. Zhou, C. Barnes, E. Shechtman, and S. Amirghodsi, “Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 2266–2276, 2021.

[132] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[133] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, "Basicvsr++: Improving video super-resolution with enhanced propagation and alignment," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022.

[134] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017.

[135] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, "Vrt: A video restoration transformer," *arXiv preprint arXiv:2201.12288*, 2022.

[136] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 764–773, 2017.

[137] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[138] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, 2004.

[139] S. Li, S. Zhu, Y. Ge, B. Zeng, M. A. Imran, Q. H. Abbasi, and J. Cooper, "Depth-guided deep video inpainting," *IEEE Trans. Multimedia*, vol. 26, pp. 5860–5871, 2024.

[140] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019.

[141] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, pp. 8748–8763, PMLR, 2021.

[142] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[143] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "Hevc complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, 2012.

[144] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[145] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock, "Vmaf: The journey continues," *Netflix Technology Blog*, vol. 25, no. 1, 2018.

[146] R. Rassool, "Vmaf reproducibility: Validating a perceptual practical video quality metric," in *Proc. IEEE Int. Sym. Broad. Mul. Sys. Broad.*, pp. 1–2, IEEE, 2017.

[147] E. Sogancioglu, S. Hu, D. Belli, and B. van Ginneken, "Chest x-ray inpainting with deep generative models," *arXiv preprint arXiv:1809.01471*, 2018.

[148] H. Esfandiari, S. Weidert, I. Kövesházi, C. Anglin, J. Street, and A. J. Hodgson, "Deep learning-based x-ray inpainting for improving spinal 2d-3d registration," *Int. Journal Medic. Robot. Comp. Assist. Surg.*, vol. 17, no. 2, p. e2228, 2021.

[149] J. V. Manjón, J. E. Romero, R. Vivo-Hernando, G. Rubio, F. Aparici, M. de La Iglesia-Vaya, T. Tourdias, and P. Coupé, "Blind mri brain lesion inpainting using deep learning," in *Simul. and Synth. in Medic. Imag. Workshop*, pp. 41–49, Springer, 2020.

[150] B. Nguyen, A. Feldman, S. Bethapudi, A. Jennings, and C. G. Willcocks, "Unsupervised region-based anomaly detection in brain mri with adversarial image inpainting," in *Proc. IEEE. Int. Cof. Sympos. Biomed. Imag.*, pp. 1127–1131, IEEE, 2021.

[151] K. Armanious, Y. Mecky, S. Gatidis, and B. Yang, "Adversarial inpainting of medical image modalities," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3267–3271, IEEE, 2019.

[152] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," *arXiv preprint arXiv:1806.03384*, 2018.

[153] N. K. Singh and K. Raza, "Medical image generation using generative adversarial networks: A review," *Health informatics: A computational perspective in healthcare*, pp. 77–96, 2021.

[154] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Trans. Intel. Sys. Tech.*, vol. 13, no. 4, pp. 1–23, 2022.

[155] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, *et al.*, "The future of digital health with federated learning," *NPJ Digit. Medic.*, vol. 3, no. 1, pp. 1–7, 2020.

[156] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850–10869, 2023.

[157] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Sig. Process. Mag.*, vol. 35, no. 1, pp. 53–65, 2018.

[158] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *arXiv preprint arXiv:2008.12595*, 2020.

[159] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[160] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 10684–10695, 2022.

[161] Z. Zhao, H. Bai, Y. Zhu, J. Zhang, S. Xu, Y. Zhang, K. Zhang, D. Meng, R. Timofte, and L. Van Gool, “Ddfm: denoising diffusion model for multi-modality image fusion,” in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 8082–8093, 2023.

[162] Q. Sun, Y. Cui, X. Zhang, F. Zhang, Q. Yu, Y. Wang, Y. Rao, J. Liu, T. Huang, and X. Wang, “Generative multimodal models are in-context learners,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 14398–14409, 2024.

[163] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, “Flamingo: a visual language model for few-shot learning,” *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 23716–23736, 2022.

[164] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, “Can ai-generated text be reliably detected?,” *arXiv preprint arXiv:2303.11156*, 2023.

[165] D. Johnson, R. Goodman, J. Patrinely, C. Stone, E. Zimmerman, R. Donald, S. Chang, S. Berkowitz, A. Finn, E. Jahangir, *et al.*, “Assessing the accuracy and reliability of ai-generated medical responses: an evaluation of the chat-gpt model,” *Research square*, 2023.

[166] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen, “Genad: Generative end-to-end autonomous driving,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 87–104, Springer, 2025.

[167] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, “Gaia-1: A generative world model for autonomous driving,” *arXiv preprint arXiv:2309.17080*, 2023.

[168] I. Jurenka, M. Kunesch, K. R. McKee, D. Gillick, S. Zhu, S. Wiltberger, S. M. Phal, K. Hermann, D. Kasenberg, A. Bhoopchand, *et al.*, “Towards responsible development of generative ai for education: An evaluation-driven approach,” *arXiv preprint arXiv:2407.12687*, 2024.

[169] H. Wu and J. Zhou, “Iid-net: Image inpainting detection network via neural architecture search and attention,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1172–1185, 2021.

[170] B. Yu, W. Li, X. Li, J. Lu, and J. Zhou, “Frequency-aware spatiotemporal transformers for video inpainting detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 8188–8197, 2021.