Generating Virtual Camera Views Using Generative Networks

Violeta Menéndez González

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

September 2023

Abstract

The challenge of novel view synthesis involves generating new synthetic images from different points of view based on existing views. This task is essential for various creative endeavours, like computer graphics, free viewpoint video, and media production. The primary motivation driving our research lies within television production and event coverage of resource-constrained events, where ideal camera positions are often limited. Our goal is to overcome physical limitations by computationally generating optimal camera views. However, this task comes with its own set of challenges, including constraints on the number of available cameras and specialised equipment, feasible camera positions, and computational resources. The task of novel view synthesis is intrinsically ill-posed, as there exist many equally valid solutions to a single problem, and the presence of these physical constraints further amplifies the uncertainty and ambiguity of the problem. Traditional computer vision methods struggle to capture scene semantics and reconstruct high ambiguity areas given a sparse set of inputs. Instead, we harness the power of generative deep learning algorithms to recreate realistic and consistent novel content.

We first delve into inpainting techniques designed to fill the information gaps resulting from disocclusions caused by the camera change of perspective. Our research begins with a stereo camera scenario, wherein the inpainting model learns to see behind objects and reconstruct the disocclusion holes while ensuring consistency across cameras. We introduce a novel deep learning framework that transforms the inpainting problem into a self-supervised task. This involves the use of a bank of geometrically-meaningful object masks, comprising context and synthesis areas used to create virtual objects on our datasets. The network then recovers image content by propagating textures and structures in these synthesis areas, conditioned on the information available in the context regions. Multi-camera consistency is achieved by leveraging the stereo-camera object context and the introduction of a disparity-based training loss.

Subsequently, our research explores more complex and realistic event capture scenarios and camera arrangements, including larger baseline changes and different camera planes. This new setting results in more significant disocclusions in areas unseen by any camera input. To address this, we study sparse scene-agnostic representation methods based on Neural Radiance Fields, which facilitate generalisation from a limited number of input views, suitable for both static and dynamic scenes. We employ convolutional encoding volumes to aid the networks in learning general geometry and motion priors. To enhance our system's creative capabilities, we also leverage adversarial training to hallucinate novel and diverse content in missing areas. This allows us to represent and render new points of view without the need for scene-specific optimisation.

Key words: Novel View Synthesis, Image Inpainting, Stereo Inpainting, Adversarial Training, Neural Radiance Fields, Neural Rendering, Computational Wizardry

Email: violeta.menendez.gonzalez@gmail.com

WWW: https://violetamenendez.github.io/

Acknowledgements

I am deeply grateful for the unwavering support and guidance I have received on this journey. First and foremost, I extend my heartfelt appreciation to my principal supervisor, Dr. Simon Hadfield, whose tireless support and patience have been instrumental in my successful completion of this PhD. He generously shared his time and devoted considerable effort to support me both academically and personally, offering much-needed encouragement during moments of faltering motivation. I also wish to express my gratitude to my secondary supervisor, Dr. Andrew Gilbert, whose invaluable assistance has greatly contributed to my research and professional growth. I have had the privilege of learning from his extensive experience, and his support has been indispensable. I extend my sincere thanks to my industrial supervisors, Dr Graeme Phillipson and Dr Stephen Jolly, for believing in me and providing the exciting opportunity to embark on this PhD journey. Their unwavering commitment ensured I had the necessary resources and moral support to thrive. I consider myself exceptionally fortunate to have been part of such a supportive academic network, and I owe much of my personal and academic development to the dedication of my advisors.

I would also like to thank all my colleagues at the BBC and CVSSP, who made me feel at home in both the North and the South. Our shared experiences, from fun and enlightening lunchtime discussions to bake-offs, made returning to work a daily joy. Special thanks to the friends I made there, Celyn and Guillaume, for the enriching GAN masterclasses while climbing; Yusuf, for being a supportive presence in my life; Tommy, for being my office bestie and top 5 enemy, dedicating me with enormous patience and continuous motivation to keep going; Nika, for being the best desk neighbour in the North and South, and giving me the girl power I needed. We all shared the burdens of PhDs and global pandemics together, and I would not have survived without their support. I am also thankful for the friendships I made at conferences and the memorable times we had together.

Special thanks to my friends in Manchester, Madrid, and all around the world – Danny, Ben, Tom, Javier, Lewis, Charlotte, Shaz, Hector, Luis, Sara, and all the rest – who put up with me during the hard times and celebrated my successes. To my friend and proofreader Dani, who has more patience than a saint and has always been keen to help me in every aspect of my life. To my partner in adventures, Estefanía, whose long audio notes (and pictures of morritos) kept me afloat and made me laugh when I wanted to cry. To my Spanscouse friends, Javi and Gala (and Balbinka), who give me a safe space in their home and hearts. To my lifelong friends, Marta and Eva, for their free therapy sessions and hugs, and their endorsement of my procrastibaking.

And finally, to all my family. To my parents Beatriz and Rafael, whose selfless love kept me warm and strong even when far. To the best little brother Javi, who is taller than me, but I am OK with it. To Indy for being cute and sleeping on my suitcase when I visit. And to the best and smartest twin, Iris, who assumed my role of the elder sister and provided me with the strength, the moral support, and the academic and graphic designer advice I needed.

I cannot fathom this journey without each and every one of you by my side.

Contents

Nomenclature

BMVC	British Machine Vision Conference
CVPR	Conference on Computer Vision and Pattern Recognition
AI4CC	AI for Content Creation
VUA	Video Understanding and its Applications
SSIM	Structural Similarity Index
PSNR	Peak Signal-to-Noise Ratio
LPIPS	Learned Perceptual Image Patch Similarity
DNN	Deep Neural Network
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
MLP	Multi-Layer Perceptron
NeRF	Neural Radiance Field
PE	Positional Encoding
GAN	Generative Adversarial Network
VAE	Variational Auto Encoder
NSFF	Neural Scene Flow Field
AI	Artificial Intelligence
DL	Deep Learning
VR	Virtual Reality
AR	Augmented Reality
XR	Extended Reality
MVS	Multi-View Stereo
FVV	Free Viewpoint Video
NVS	Novel View Synthesis
SVS	Sparse View Synthesis
IBR	Image-Based Rendering
N-IBR	Neural Image-Based Rendering

MPI	Multi-Plane Image
LDI	Layered Depth Image
SRN	Scene Representation Network
GDL	Geometric Deep Learning
PSV	Plane Sweep Volume
SDE	Stochastic Differential Equation
NCC	Normalised Cross-Correlation
BBC	British Broadcasting Corporation
SfM	Structure from Motion
RGB	Red, Green and Blue
RGB-D	Red, Green, Blue, and Depth

List of Figures

List of Tables

Declaration

The work presented in this thesis is also present in the following manuscripts:

- Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. SaiNet: Stereo aware inpainting behind objects with generative networks. *AI for Content Creation (AI4CC) at CVPR*, 2022
- Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. SVS: Adversarial refinement for sparse novel view synthesis. In *Proceedings of the 33rd British Machine Vision Conference (BMVC)*, 2022
- Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. ZeST-NeRF: Using temporal aggregation for zero-shot temporal NeRFs. In *Proceedings of the 34th British Machine Vision Conference Workshops (BMVC)*, 2023.

Chapter 1

Introduction

Creativity is an essential characteristic of human intelligence. It allows us to come up with ideas, alternatives, or possibilities, to be able to create, communicate, discover, and solve problems. Artificial Intelligence (AI) has come a long way in recent years trying to mimic human perception, understanding, reasoning, and creativity. In particular, the field of computer vision is booming with the rise of Deep Learning (DL) and generative techniques. These approaches use the power of Deep Neural Networks (DNNs) to automatically extract features from data, reducing the need for complex feature engineering, and allowing AI models to better *understand* and create content.

Generative networks are used to generate new data samples that are similar to a given dataset. These models can be used to generate new data based on certain cues, edit existing data, or recover missing information. Notably, deep generative methods have facilitated advancements in many topics which were traditionally challenging for machine learning approaches, such as image restoration, style transfer, 3D modelling and inpainting. These techniques have a wide application in many creative areas like computational photography, virtual and augmented reality (VR/AR), and media production.

For this work, we focus on one such application, Novel View Synthesis (NVS). The objective of this task is to generate new synthetic images from new points of view, given one or many views of a scene. This task poses many challenges, as it is an intrinsically ill-posed problem which suffers from uncertainty and ambiguity. Images are two-dimensional projections of the three-dimensional real world. It is our own human stereo cameras (the eyes) that allow us to have a sense of depth and 3D space. Although optical illusions exist for human perception, we can usually rely on our understanding of the world to reason about scenes. A lot of effort has been directed towards helping AI replicate this reasoning. To tackle these problems, new DL approaches need to gather a deep understanding of the scene by encoding the semantics of the input views. This is necessary to be able to extract and extrapolate information from the inputs, to reason about the structure and appearance of a scene.

In this thesis, we aim to approach the problem of NVS as one of image generation. Our objective is to build a DL pipeline that can take advantage of nearby views to reconstruct missing content. More precisely, we focus on developing techniques that can be applied to video, and can work using only a few input views and generalise well to unseen scenes.

1.1 Motivation

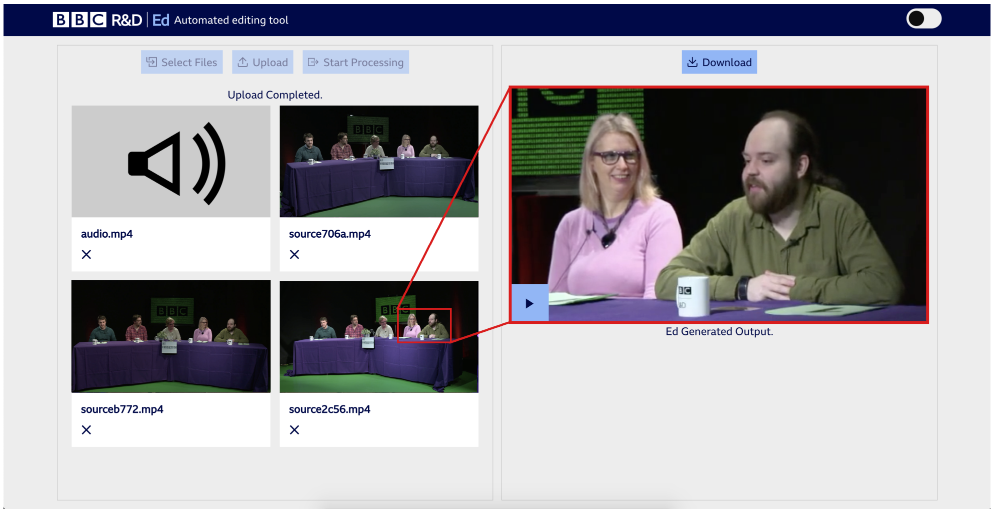


Figure 1.1: **Ed system.** BBC's automated editing tool which combines different machine learning algorithms for creating edited coverage of live events. Ed takes raw camera feeds, and cuts and crops between them to create a "virtual camera" view.

The ideal staging for television production can only be found in professional TV studios. In these studios, producers have control over many variables that they modify to create the best results. This can be the lighting, the arrangement of the stage and people, or the camera positions for optimal framing. But access to these studios is not always an option. For example, for live events recorded in venues like theatres, the best viewpoints may be taken by the audience (Figure 1.2). For more independent or low-budget events, the resources are limited, and producers have to work with what they have. It would be very convenient if there were a computational way of overcoming these physical limitations.

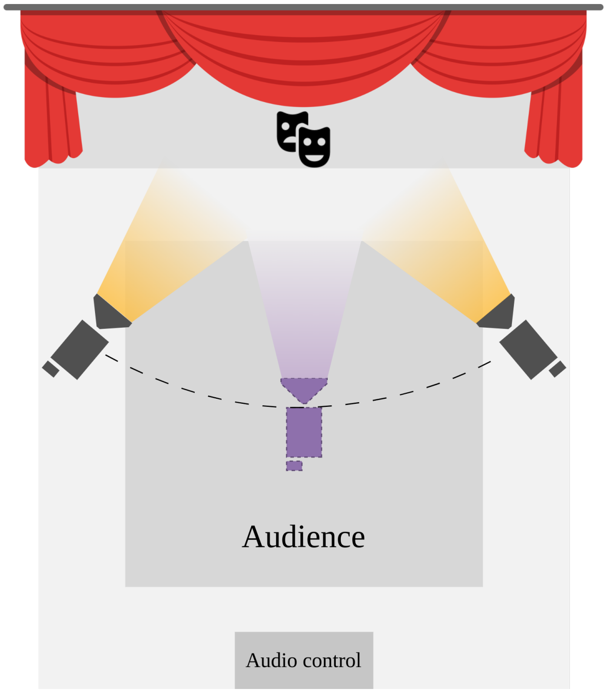
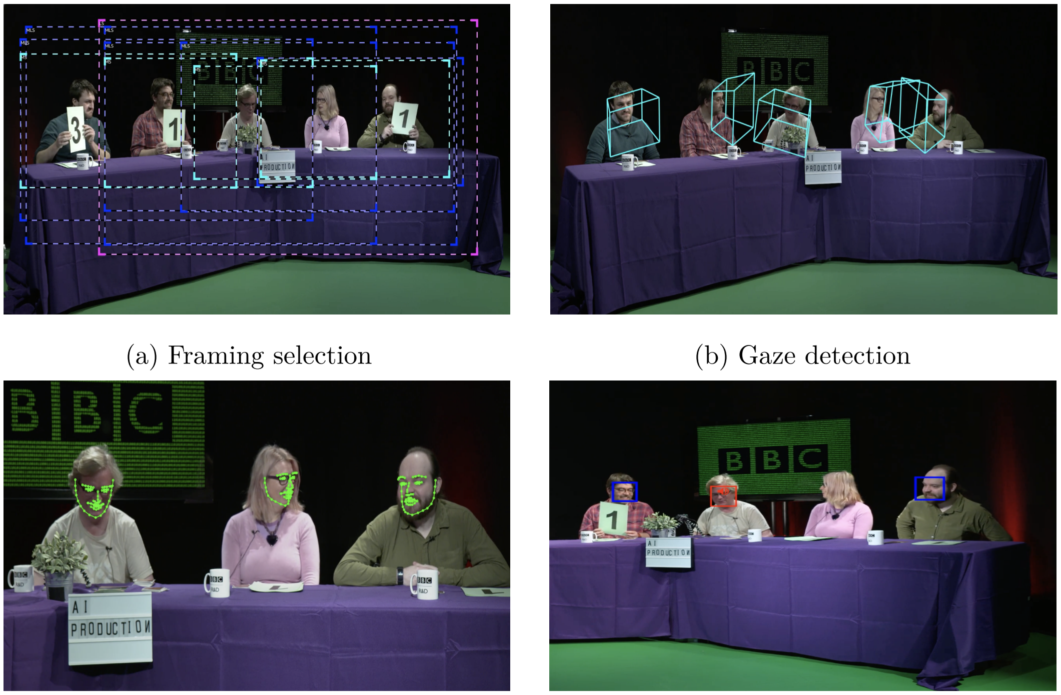


Figure 1.2: **Radio theatre example.** Optimal camera viewpoint, indicated in purple, is located in the audience.

This project is funded by the British Broadcasting Corporation (BBC), seeking a potential extension to their *Ed* system [1, 2] (Figure 1.1). *Ed* is an AI system that combines different machine learning tools for creating edited coverage of live events (Figure 1.3). Its inputs are high-resolution locked-off cameras that are deployed at venues with live audiences. The system then creates "virtual camera" views by cropping these raw camera feeds and cuts between them to produce output. In real-world deployments, camera positioning is commonly constrained to suboptimal positions and is very dependant on the venue and space available. Having the ability to synthesise more favourable camera views that were physically untenable would give the *Ed* system more directorial control (for example, positioning the view from the middle of the audience). Furthermore, it would enable producers to perform more complicated and artistic shots that may not be feasible with their limited resources (e.g. aerial images).



(a) Framing selection

(b) Gaze detection

(c) Face detection and landmarking

(d) Speech detection

Figure 1.3: **Ed system features.** The Ed system can take decisions about shots and framing depending on multiple factors, e.g., who is speaking, where people are looking, the arrangment of people and objects of the scene.

The motivations for this system are to make professional video production more accessible to smaller independent crews with limited budgets, simplify production for novice users, and increase the scope and scale of the BBC's coverage. In particular, the BBC are interested in applying this system to cover large-scale independent events like the Edinburgh festival.

Looking into the future, AI-based production tools benefiting from NVS technologies promise a radical transformation in content creation and consumption. These may be used in producing all kinds of media, from panel shows to high-end dramas, to allow for viewpoints to be chosen independently of camera location. Similar techniques are currently used in sport replays, but as the technology matures, we might expect it to eventually become suitable for presentation of live material. This would revolutionise how audiences engage with content, which may include free-viewpoint consumption of live events and the use of Extended Reality (XR) devices.

1.2 Challenges and objectives

This PhD will focus on two main applications of computer vision: Novel View Synthesis (NVS) and inpainting. We will use NVS techniques to be able to create new virtual camera points of view from the available cameras. The change of perspective from one camera view to another creates image disocclusions, the gap of information that was occluded by the scene's objects and is now visible from another point of view. Recovering consistent content from partial-occlusions, where content may be occluded in some input views but visible in others, is challenging. The task is even more problematic when these occlusions are full, creating areas where no input view can see. The recovery of this completely unseen content by using techniques like image inpainting is a complex endeavour that requires a deep understanding of the scene. The goal of inpainting is to fill in the gaps with realistic and plausible content that is consistent and pleasant to the human eye.

As this work's motivation lies within television production, and, in particular, we are interested in contributing towards accessible and cost-effective AI video editing tools, we need to find an approach that can be easily applied to video recorded with limited resources. Factors such the availability of cameras or specialised equipment, recording locations, and computational resources, are key challenges that our proposed system needs to overcome. Therefore, we are looking at efficient approaches that can operate in general settings, and that can create novel content from very few and sparse input views. Unfortunately, the essentially ill-defined NVS problem suffers from even more ambiguity and uncertainty when the number of available views is reduced, and the camera baseline expands, resulting in larger disocclusions. This complexity is particularly pronounced when modelling the temporal component of videos, having to ensure the consistency of the generated content not only across viewpoints, but also over time.

These challenges provide a natural set of objectives which set a path to the overall aim of this thesis. These objectives are:

1. Explore the use of generative inpainting techniques to the application of novel view synthesis.
2. Develop a generative deep learning framework that exploits multi-view camera information and can generate novel content in unobserved regions avoiding visual artefacts.
3. Provide view-synthesis models with a generalisation mechanism to be applicable to diverse and sparse camera setups in varied scenes.
4. Explore the temporal consistency and generalisation of videos for the view synthesis problem.

1.3 Contributions

To fulfil these objectives, this thesis introduces a series of technical contributions. This section will provide an overview of the chapters in this thesis and outline our contributions and how they relate to the proposed objectives.

Chapter 2 – Literature Review

In this chapter we present a comprehensive survey of literature on inpainting and novel view synthesis. We start by giving an overview of traditional inpainting methods and how techniques have evolved with the introduction of DL approaches. We follow a similar outline for NVS methods, and focus on scene representations apt for this task. Then we finish the chapter with a section that follows dynamic approaches to applications from Free Viewpoint Video (FVV) to NVS.

Chapter 3 – SaiNet: Stereo aware inpainting behind objects with generative networks

To address Objective 1, we present an end-to-end network for stereo-consistent image inpainting with the aim of recovering large missing regions behind objects. The proposed model consists of an edge-guided UNet-like network [134] using Partial Convolutions [87]. We enforce multi-view stereo consistency by introducing a disparity loss. More importantly, we develop a training scheme where the model is learned from realistic stereo masks representing object occlusions, instead of the more common random masks. The technique is trained in a self-supervised way. Our evaluation shows competitive results compared to previous state-of-the-art techniques.

This work was accepted and presented at the AI for Content Creation (AI4CC) workshop at the Conference on Computer Vision and Pattern Recognition (CVPR) 2022 [42].

Chapter 4 – SVS: Adversarial refinement for sparse novel view synthesis

To tackle Objectives 2 and 3, we introduce the challenge of Sparse View Synthesis (SVS). This is a view synthesis problem where the number of reference views is limited, and the baseline between target and reference view is significant. Neural rendering approaches like NeRFs [104] have become very popular due to their photo-realistic results. However, they tend to be very expensive, requiring a multitude of input views and a very long per-scene optimisation process. Furthermore, under the SVS conditions, current radiance field methods fail catastrophically due to inescapable artefacts such 3D floating blobs, blurring and structural duplication.

Advances in network architecture and loss regularisation are unable to satisfactorily remove these artefacts. The large occlusions within the scene ensure that the true contents of these regions is simply not available to the model. In this work, we instead focus on hallucinating plausible scene contents within such regions, while additionally tackling the need for overfitting to specific scenes. To this end we unify general radiance field models [15] with adversarial learning [45] and perceptual losses. The resulting system provides up to 60% improvement in perceptual accuracy compared to current state-of-the-art radiance field models on this problem.

This work has been published at the British Machine Vision Conference (BMVC) 2022 [43] and presented at AI4CC at CVPR 2023.

Chapter 5 – ZeST-NeRF: Using temporal aggregation for Zero-Shot Temporal NeRFs

Finally, to approach Objective 4, we extend our static NVS model to the processing of dynamic scenes. Previous models have focused on implicitly representing static and dynamic scenes using NeRF [104]. These models achieve impressive results but are costly at training and inference time. They overfit a MLP to describe the scene implicitly as a function of position. In this chapter, we propose ZeST-NeRF, a new approach that can produce temporal NeRFs for new scenes without retraining. We can accurately reconstruct novel views using multi-view synthesis techniques and scene flow-field estimation, trained only with unrelated scenes. We demonstrate how existing state-of-the-art approaches from a range of fields cannot adequately solve this new task and demonstrate the efficacy of our solution. The resulting network improves quantitatively by 15% and produces significantly better visual results.

This work was published and presented at the Video Understanding and its Applications (VUA) workshop at BMVC 2023 [44].

Chapter 6 – Conclusion and Future Work

Lastly, we summarise the findings of this thesis in the conclusion chapter, and discuss potential directions for future work.

Chapter 2

State-of-the-Art

The contributions of this thesis involve the use of Novel View Synthesis (NVS) techniques, to interpolate and extrapolate virtual camera views from new perspectives, and image inpainting, to recover the missing information caused by dis-occlusions. In this chapter, we present a comprehensive review of the techniques in each of these fields. Then, in the last section of this review, we provide an insight on dynamic approaches that bring these techniques to the application of video processing.

2.1 Inpainting

Image inpainting is the task of filling in the missing regions of an image plausibly. It has many vital applications in computer vision, and image and video processing: the removal of unwanted objects (e.g. superimposed text), image and film restoration (e.g. scratches or cracks), image completion (e.g. dis-occlusions), cinema post-production, among others. Despite the impressive advancements in the field, image inpainting is still a challenging problem. It requires a high-level understanding of scenes to be able to synthesise visually realistic and semantically coherent missing regions, which becomes progressively more difficult with the size increase of the missing holes and the complexity of the scene.

Traditionally, inpainting has been divided focusing on two main image characteristics, structure (edges/corner) and texture (surface patterns).

2.1.1 Structural inpainting

Structural inpainting centres on the uniformity of the geometrical structure of the images, trying to preserve limits and edges. For this purpose, the seminal work of Bertalmio *et al.* [7] presented a *diffusion-based* technique (note this was prior to the growth of deep learning and is not a diffusion network) that attempted to replicate the strokes of professional painting conservators. Their approach uses Partial Differential Equations to propagate local image structures from the exterior to the interior of the missing hole, following the direction of isophotes. Classical diffusion-based methods are useful for completing small or narrow regions, like cracks or scratches on old photographs, and in general, preserve unconnected edges. But they are not suitable for recovering the missing information over large areas, as they cannot capture the texture of objects and instead produce blurred outcomes.

2.1.2 Texture inpainting

Textured images contain rich statistical content; therefore, methods that aim to reconstruct texture need to exploit image statistical and self-similarity priors. To synthesise a new texture, *statistical-based* methods like Heeger and Bergen [52] typically start with an output image containing pure noise, and keep perturbing that image until its statistics match the estimated statistics of the input texture. However, they can synthesise only textures which are highly stochastic and usually fail to do so for inhomogeneous textures containing structure. The seminal work of Efros and Leung [32] introduced a method where the texture to be synthesised is learned from similar regions (samples) from a known part of the image, and stitched back together, growing the reconstructed texture one pixel at a time. However, this technique suffers from a high computational cost, error propagation and the generation of unnatural repetitive patterns.

2.1.3 Hybrid models

Nonetheless, natural images contain a mixture of complex structures and textures, consequently creating the need for hybrid methods which combine both types of inpainting.

Bertalmio *et al.* [8] proposed a hybrid algorithm which decomposes the original image into two components, one with structural attributes and another with textural qualities, which are then inpainted using structural [7] and texture inpainting [32] respectively, and subsequently combined into an output image. A similar but more straightforward and faster hybrid approach was proposed by Criminisi *et al.* [24], where they proved that exemplar-based texture synthesis is sufficient for propagating extended linear image structures (like object boundaries). They prioritised the synthesis of regions on the continuation of image structures in their filling algorithm. One standard handicap of these techniques lies on the greediness of the algorithms, once the filling-in of the patch is done, that patch remains unchanged, which may lead to visual inconsistencies.

2.1.4 Patch-based approaches

Instead of synthesising the regions pixel by pixel, *patch-based* solutions are introduced to copy entire areas in one step. To improve the searching of suitable patches, Simakov *et al.* [146] use bidirectional similarity to capture the visual information of the patches and to avoid introducing visual artefacts. However, this technique is relatively slow and cannot be used for real-time applications. To reduce the computational cost, PatchMatch [3] designed a fast nearest neighbour searching algorithm using natural coherence in the imagery as prior information to find the best fitting patch. Patch-based methods can generate photorealistic textures for relatively large missing holes, and they may work well in some images with repetitive structures, but they lack semantic awareness. They assume that the non-hole regions have similar semantic contents with the missing regions, which may not be accurate in some tasks with unique structures, such as face image inpainting. To find patches that are similar and contain different but valid semantics, some approaches have included the use of external databases [49, 172]. These techniques were meant for removing whole objects and do not perform well filling arbitrary holes that partially occluded objects. Also, they depend significantly on the contents of the databases that may not contain a similar image to the target image.

2.1.5 Semantic inpainting

With the advancements of Deep Learning and the availability of large-scale training data, deep Convolutional Neural Networks (CNNs) became a popular tool for image processing. Initial CNN models attempted to perform image inpainting by using feature learning with Denoising Autoencoders [177], translation variant interpolation [131], or exploiting the shape of the masks [76]. Yet all these methods were only applicable to tiny and thin masks and lacked semantic understanding of the scenes. With the addition of Generative Adversarial Networks (GANs) [45], CNN architectures were able to extract meaningful semantics from images and generating novel content. One of the early attempts by Pathak *et al.* [123] takes advantage of GANs for inpainting by feature learning. Their Context Encoder (CE) architecture consisted of an encoder-decoder, where the decoder generated the contents of a missing image region conditioned on its surroundings. It is based on a latent feature representation created by the encoder and connected to the decoder through a channel-wise fully-connected layer, which allows each unit in the decoder to reason about the entire image content. They found this model captured the semantics of structures, as well as their appearance. By jointly training their network with an adversarial loss in addition to a reconstruction (L_2) loss, the model learns sharper more realistic patches. CE only takes advantage of the structure of holes during training but not during inference, resulting in blurry or unrealistic images. Instead, Yeh *et al.* [183] perform semantic inpainting using the latent space of a trained GAN to find the closest encoding to the corrupted image, which is then used to reconstruct the image using a generator. One of the significant advantages of their method is that it does not require the masks for training and can be applied for arbitrarily structured missing regions during inference. Although compared with CE, this method generates more realistic images with sharper edges, it relies too strongly on the training process, making it difficult to represent complex scenes in the world. While the CE discriminator did not take into account the context boundary, so it struggled to maintain global consistency and often generated results with visual artefacts. Iizuka *et al.* [60] solved this problem by using both local and global context discriminators, which helped the local consistency of generated patches and still maintained overall image coherence. Their work handled arbitrary resolutions and image holes by using a fully convolutional network with dilated convolutions, unlike CE [123] which only applies to 128×128 fixed-size images with 64×64 centred holes. Despite the visual improvements concerning previous works, their model sometimes generated colour inconsistencies, which they solved by applying a post-processing step of fast marching method and Poisson blending. Dilated convolutions are not very effective for representing spatial correlations, and may not be able to weigh the importance of some features over others. Based on a similar architecture to [60], Yu *et al.* [187] added a contextual attention layer to aid the modelling of long-term correlations between the distant contextual information and the hole regions. They substituted the post-processing blending with a refinement network (that adds the detailed information) as well as using a spatially discounted reconstruction loss. This gives more weight to the pixels closer to the border, which improves training stability and speed, being able to train the network in a week instead of months.

2.1.6 Learnable convolutions

Traditional vanilla convolutions are applied naively to both image and inpainting holes, consequently, their learned values depend on the initialisation values of both the images (valid pixels) and the holes (invalid pixels). This process usually leads to visual artefacts (lack of texture, obvious colour contrasts, or mismatched edges), needing the application of expensive post-processing or refinement networks. To produce an integrated standalone smooth model, Liu *et al.* [87] proposed a U-Net-like architecture [134], with *partial convolutions*: masked and re-normalised convolutional filters conditioned only on valid pixels. Previous works focused on centred rectangular holes, which may cause methods to overfit to this kind of mask. Their method was generalised to handle more realistic irregular holes, collecting a large benchmark of irregular masks with varying sizes and locations, and studying the effects when the holes are in contact with the image border. They significantly improved results from previous baselines, but still failed especially on cases when holes are large and involve different segments of an image, or when they have strong structure or contours. Yu *et al.* [186] extended the idea of partial convolutions with *gated convolutions* by generalising to a learnable dynamic features selection mechanism for each location across all layers. Unlike Liu *et al.* [87]

who adopted a U-Net-like network, Yu *et al.* [186] continued with a similar architecture from their previous contextual attention work [187]. Whilst they moved away from global and local GANs, which are more relevant for single regular masks, and introduced SN-PatchGAN, a patch-based GAN loss with spectral-normalisation. Another main contribution was their inclusion of a sketch as a guide to the network to generate more user-desired results. They evaluated their model on Places2 and CelebA-HQ, showing that it performs well in natural scenes and faces. They also introduced a simple algorithm to automatically generate random free-form masks on-the-fly during training, which creates masks that are similar to those drawn in real use-cases, efficient, and diverse to avoid over-fitting, while Liu *et al.* [87] collected a fixed set of irregular masks from an occlusion estimation method.

2.1.7 Structural deep learning

Although these methods show impressive improvements in image inpainting, many fail to reconstruct reasonable structures and usually over-smooth surfaces. With a two-stage adversarial model, EdgeConnect [109] tries to generate structures by using additional prior information, firstly recovering edges of the missing regions, and then filling in the fine detail using the edge maps as a priori. However, the distribution of edge images differs substantially from the distribution of the target images, discarding too much useful information (e.g. colours), making it challenging to generate detailed textures especially in the boundaries of objects. To solve this, StructureFlow [132] presents a similar model to EdgeConnect but using edge-preserved smooth images as the global structure information. They also introduce the use of appearance flow to model long-term corrections between missing regions and existing regions for vivid texture generation.

2.1.8 Diffusion models

In the last few years, diffusion probabilistic models have appeared as a popular new trend of generative models showing impressive results. These models have proved to produce higher-quality content compared to other generative methods like Variational

Auto Encoders (VAEs) and normalising flows. Furthermore, they are very simple to train, as opposed to GANs which are highly unstable and can cause problems with vanishing gradients or mode collapse. In the original paper, Sohl-Dickstein *et al.* [150] propose using the diffusion process from nonequilibrium thermodynamics principles to model data. This approach uses tractable small steps to destroy structure in a data distribution by adding Gaussian noise. Then it uses a neural network to learn a reverse diffusion process, where the network restores the structure in the data. More recently, Ho *et al.* [54] take this concept to the field of image synthesis by combining diffusion probabilistic models with denoising score matching. They reparameterise diffusion models to learn the noise of an image at a certain time step, by using a U-Net [134] architecture with group normalisation [174] and self-attention layers [165]. To feed the time information into the network, they apply a sinusoidal embedding [165] to the time step. The noise is sampled at each time step by scaling the variance and mean using a linear scheduler to avoid the variance exploding. Nichol and Dhariwal [112] further improve this model by using a cosine scheduler instead of a linear one. This destroys the image information more slowly, allowing the model to retain more information in the later stages of diffusion. In addition, they learn both the variance and the mean of the image noise, which leads to improvements in the log-likelihood. In a follow-up paper, Dhariwal and Nichol [27] heavily improve the outcome images by improving the overall architecture. They change the size of the network, increase the attention blocks, and use residual blocks from bigGAN [10] for upsampling and downsampling. They also propose Adaptive Group Normalisation, which adds a linear projection of the time step and class label onto the group normalisation layer. Moreover, they use a separate classifier to guide the diffusion network generating images from a given class.

Even though diffusion models are used for all kind of generative modelling tasks, there has been a significant surge of applications in the field of image synthesis. Many approaches focus on tackling either one problem, or come up with a multi-task framework for the tasks of text-to-image synthesis, image-to-image translation, image editing, image super resolution, and image inpainting. Song *et al.* [151] proposes a pipeline based on Stochastic Differential Equations (SDEs) and score-matching for noise injecting and corresponding reverse-time noise removal, which can tackle a variety of inverse problems including inpainting. Likewise, Meng *et al.* [100] uses SDEs to build their SDEdit model, which permits inpainting regions of an image conditioned on a rough sketch or set of colours. Ho *et al.* [55] introduce a new framework for high resolution image synthesis called Cascaded Diffusion Models (CDM). This process consists of multiple diffusion model networks of increasingly higher resolution which are conditioned on ImageNet [136] classes and the previous lower-resolution generated image. Saharia *et al.* [137] propose Palette, a unified multi-task framework and evaluation protocol applied to four different image-to-image translation tasks. In their experiments they show that using an L_2 loss leads to a higher degree of diversity in model samples, whereas L_1 produces more conservative outputs. They also show the importance of self-attention layers in the U-Net architecture of diffusion models. When trained directly on the inpainting task, diffusion models can smoothly inpaint regions of an image without edge artifacts. The GLIDE model by Nichol *et al.* [113] is a classifier-free conditional model which provides better results than the alternative conditional models. Guidance of the image generation process is simplified when conditioning on information that is difficult to predict with a classifier. RePaint [92] instead features an enhanced denoising strategy that uses resampling iterations to enhance the overall conditioning of the image.

Unfortunately, diffusion models have a critical downside, they are inherently slow to sample from, needing a few thousand steps of iteration to generate images from pure Gaussian noise. Chung *et al.* [21] suggest that starting the diffusion process from pure Gaussian noise is unnecessary and slows the process. They show that it is enough to start the process from a closer step where the target image has not been completely corrupted with noise. Instead, Jing *et al.* [67] try to reduce the duration of the sampling process by reducing dimensionality of the subspace onto which diffusion is realised at each time step. With Denoising Diffusion Restoration Models (DDRM), Kawar *et al.* [72] propose an efficient and unsupervised posterior sampling method. The model takes advantage of a pre-trained denoising diffusion generative model for solving any linear inverse problem. In order to allow for the training of diffusion models with limited computational resources, some methods like Latent Diffusion Model (LDM) [133] have shifted the diffusion process to the latent space using pre-trained autoencoders. ImageBART [34] improve the sampling speed by using autoregressive factorised models to learn the inversion of a multinomial diffusion process. The individual transition probabilities are modelled using independent autoregressive encorder-decoder transformers, which can attend to global context from its preceding representation. ImageBART is capable of solving free-form image inpainting problems given user-provided masks, as well as text-guided image modification. Hu *et al.* [56] propose a non-autoregressive text-to-image model based on a discrete Vector Quantised VAE where the latent space is modelled by a conditional Denoising Diffusion Probabilistic Model [54]. This approach takes advantage of the global context, which can be applied to irregular mask inpainting without re-training for the specific task.

2.1.9 Stereo consistent inpainting

There is little research done on stereoscopic image inpainting in the framework of deep learning. Following a similar trajectory to monocular approaches, traditional patch-based methods [105, 106, 169] find example patches from the available parts of the image and fill the holes applying consistency constraints. Wang *et al.* [169] simultaneously inpaint colour and depth images using a greedy segmentation-based approach, inpainting first partial occlusions using warping, and total occlusions with a depth-assisted texture synthesis technique. Morse *et al.* [105] extend PatchMatch [3] to cross-image searching and matching without explicit warping, using a completed disparity map to guide the colour inpainting. Multi-view inpainting techniques such as Gilbert *et al.* [40] create a dictionary of patches from multiple available viewpoints that are then coherently selected and combined to recover the missing region.

The first stereo inpainting approach using deep learning was made by Luo *et al.* [93]. They use a double reprojection technique to generate image occlusion masks from several new viewpoints, then apply Partial Convolutions [87] to inpaint the holes, and aggregate the results in a layered depth image. This technique shows good visual results on their own Keystone B&W dataset, but they do not take into account multi-view consistency and rely on depth maps to reconstruct the image. Conversely, other stereo techniques take advantage of both left and right views. Chen *et al.* [16] extend Context Encoder [123] to inpaint left and right views simultaneously encoding both views and aggregating them at the feature level. In addition, they introduced a local consistency loss which helps preserve the inpainting consistency at a pixel level. This model is applied to inpainting regular holes at the centre of the image. Ma *et al.* [94] use a similar architecture for two different tasks: reconstructing missing objects in one view that are available on the other view, and coherently inpainting the same holes in both views similar to Chen *et al.* [16]. To do this, they use two different stereo consistency losses, a warping-based consistency loss and a stereo-matching PSMNet-based [13] disparity-reconstruction loss. However, because of a lack of ground-truth data for object removal, they only train their model on corruption restoration data.

2.2 Novel View Synthesis

Novel View Synthesis (NVS) is the problem of generating new camera viewpoints of a scene given a fixed set of views of the same scene. NVS methods thus deal with image and video synthesis conditioned on camera pose. Key challenges underlying Novel View Synthesis are inferring the scene's 3D structure given sparse observations, as well as inpainting of occluded and unseen parts of the scene. This task has wide applications in image and video editing, animating still photographs or viewing RGB images in 3D.

It is difficult to formulate a single taxonomy of NVS methods, as they can be classified in many ways. We start showing how learned strategies have been introduced, partially or entirely replacing the more classical pipelines, and then focusing on different types of scene representations.

2.2.1 Image-based rendering

In classical computer vision methods, Image-Based Rendering (IBR) typically relies on optimisation-based Multi-View Stereo (MVS) algorithms to reconstruct scene geometry and re-project observations into the target view. As opposed to geometry-based methods, the underlying data representation of image-based methods is composed of a set of photometric observations. Chen and Williams [17] compute arbitrary intermediate frames from an image array by using image flow fields and morphing techniques, which combines interpolation of texture maps and their shape. McMillan and Bishop [99] instead attempt to reconstruct the 5D plenoptic function using resampling, avoiding the ambiguity introduced by flow fields. Gortler *et al.* [46] present The Lumigraph, a subset of the plenoptic function that describes the flow of light at all positions in all directions, which does not rely on geometric representations. Levoy and Hanrahan [80] also tackle the problem of image interpolation by reconstructing a 4D function of light fields by blending and resampling, which does not use any depth information or feature matching. Debevec *et al.* [25] implement view-dependent texture-mapping with approximately known geometry, by using projective texture-mapping and a view map that selects the best input views to recover each area. While Chaurasia *et al.* [14] attempt to provide plausible free-viewpoint navigation over datasets with missing or unrealisable geometry. IBR approaches produce high-quality photorealistic results when the images have reasonable baselines or are captured with depth sensors. However, in general, these methods struggle with sparse input observations, view-dependent effects, and large occlusions, leading to results with artefacts and holes.

2.2.2 Neural rendering

With the advances in the field of Deep Learning (DL), many of these issues were tackled by using learning-based models. Neural Image-Based Rendering (N-IBR) methods replace some of the heuristics often found in classical IBR pipelines with Deep Neural Networks (DNNs). Some approaches like Deep Blending [51] remove the need for hand-crafted blending functions by learning blending weights of the projected input images for compositing in the target image space. Others apply corrections that take into account view-dependent effects, like relighting [181], or completely remove specular effects before reprojecting to the target view like EffectsNet [160].

Another approach to aid IBR models to look more photorealistic is to use Neural Re-rendering, which employs DNNs to reprocess the render produced by the classical method. N-IBR approaches generally assume scenes to have a static appearance, which may not be the case in scenes captured outdoors or in motion. With this aim, *Neural Re-rendering in the wild* [101] introduces a rendering technique that tries to model appearance to re-render a realistic view of the scene. Pittaluga *et al.* [125] use a U-Net-based [134] re-rendering model to invert Structure from Motion (SfM) reconstructions and recreate colour images of novel views from very sparse point clouds. Meanwhile, Yu and Smith [188] borrow from both N-IBR and re-rendering methods to create a pipeline that can generate novel views from a single uncontrolled image. By combining monocular depth estimation and inverse rendering they predict the scene geometry, which can then be rendered with controlled lightning changes.

2.2.3 Multi-layer representations

All these previous models reconstruct a proxy 3D geometry to prescribe an internal representation of the scene. To explicitly model the scene with a representation that contains visibility information, DeepStereo [36] introduces Plane Sweep Volumes (PSVs). These PSVs remove the need to supply a pose input, as this information is implicit in the construction of the volume. Xu *et al.* [181] use PSV to reconstruct scene depth and appearance using 3D CNNs and aim to model the light transport of the scene to be able to handle view-dependent effects. Also, they predict attention maps to capture the visibility of pixels from different viewpoints to handle occlusions. Zhou *et al.* [193] introduce a similar representation called Multi-Plane Images (MPIs), which consist of a set of fronto-parallel planes at fixed depths from a reference camera coordinate frame, where each plane encodes an RGB image and appearance at the corresponding depth. DeepView [35] approach uses an MPI representation estimated by a learned gradient descent method, which learns to take larger, parameter-specific steps compared to standard gradient descent, converging significantly faster. This representation is good for capturing semi-reflective and semi-transparent surfaces. Srinivasan *et al.* [154] focus on generating high-quality view extrapolations given a narrow-baseline pair of images. They achieve this by inpainting the layers of an MPI using opacity and 2D flow prediction. Mildenhall *et al.* [103] proposes a 3D CNN architecture that dynamically adjusts the number of depth planes. They predict multiple MPIs, one per input view, and use a blending procedure with alpha compositing which allows them to render new views from larger-baseline view interpolation. Still, because of the fixed depth discretisation, sloped surfaces do not reproduce well, and it is memory and storage inefficient, and thus, costly to render.

To address the efficiency issue of scene representation, a more sparse layered representation is used. Layered Depth Images (LDIs) [143] are a generalisation of depth maps that represent a scene using several layers of depth maps and associated colour values. More recent work by Shih *et al.* [144] extends the original work to explicitly store pixel connectivity information, which can then locally adapt to any depth-complexity and generate a varying number of layers across the image. LDIs are naturally more memory and storage efficient and can be easily converted into a textured mesh for efficient rendering.

2.2.4 Learning-based representations

While methods based on multi-layer representations have achieved impressive results, they prescribe the model's internal representation of the scene. They do not allow the model to learn an optimal representation of the scene's geometry and appearance. A recent line of work in novel view synthesis is, therefore, to build models with learned, feature-based representations of scene properties. Generative Query Network [33] learns a low-dimensional feature embedding of a scene, explicitly modelling the stochastic nature of such neural scene representation due to incomplete observations. On the other hand, some methods choose to take advantage of the 3D information by enforcing a 3D structure in the learnt representation. Some models use 3D-explicit voxel grids [142, 155, 78, 71], which achieve a multi-view consistent modelling of a scene with limited training data. Tulsiani *et al.* [163] introduce a differentiable ray consistency term for visual coherence to recover a 3D shape from a single arbitrary 2D observation using an occupancy voxel volume. Penner and Zhang [124] aggregate stereo cost into a surface-confidence voxel grid to aid their occlusion aware reconstruction. DeepVoxels [148] is based on a Cartesian 3D grid of persistent embedded features that make use of the underlying 3D structure without having to model its geometry explicitly. Using a similar representation, HoloGAN [111] manages to disentangle shape and appearance, which allows for precise control over the pose of generated objects through rigid-body transformations of the learned 3D features. HoloGAN learns these representations from single natural images in an entirely unsupervised manner. Using an unconditional generative model to generate a representation of shapes and images, Visual Object Networks [194] manages to disentangle object texture, shape and pose. NSVF [88] defines a set of voxel-bounded implicit fields organised in a sparse voxel octree that model local properties in each cell. Due to the sparsity of this representation, it can render higher resolution free-viewpoints faster than other voxel techniques.

With voxel volumes, the fidelity of 3D information that can be represented is tied to the volume dimensions. Wiles *et al.* [173] use a more flexible approach that generalises naturally to varying resolutions, using a latent 3D point cloud of features that can be projected onto the target view. They train their end-to-end model with pairs of views in a self-supervised fashion, needing only a single image of an unseen scene at test time for view synthesis. The 3D component in their model allows for interpretable manipulation of the latent feature space at test time.

2.2.5 Implicit function

Another problem with 3D-voxel representations is that the models scale cubically with spatial resolution. This makes them exceptionally costly to run and to be unable to parametrise large scenes with sufficient resolution, as this requires a finer sampling of 3D space. These methods struggle to render scenes smoothly and fail to generalise shape and appearance across scenes. In Geometric Deep Learning (GDL), models have tried to overcome these problems by parametrising surface geometry via an implicit function. In [138], Saito *et al.* use the PIFu model to manage the 3D reconstruction of humans representing object colour as an implicit function. Niemeyer *et al.* [118] use implicit shape and texture representations. They represent surfaces as 3D occupancy fields and calculate surface intersection using numerical methods and implicit differentiation. Then they use a neural 3D texture field to predict a diffuse colour for that intersection point. Scene Representation Networks (SRNs) [149] aims to describe a scene implicitly as a continuous, differentiable function that maps a 3D world coordinate to a feature-based representation of the scene properties at that coordinate. While these techniques can in principle represent complex and high-resolution geometry, they have so far been limited to simple shapes with low geometric complexity and oversmooth renderings.

2.2.6 Neural Radiance Fields

Mildenhall *et al.* [104] proposed an entirely neural implicit scene representation with Neural Radiance Fields (NeRFs). NeRF encodes a *continuous* volume within the parameters of a deep fully-connected neural network. An MLP is used to parameterise a function to model density and colour by querying 3D location and viewing direction. These querying points are sampled along rays using ray marching. This approach revolutionised the field, as it could represent higher-resolution geometry and appearance to render photorealistic novel views of complex scenes. However, in its original form, NeRF is very costly to run, and has a long per-scene optimisation process, which prevents it from being useful in many important applications. In addition, this model exhibits some representation limitations, which are present when the scene and data are not ideal. Many following approaches have worked towards improving performance and increasing flexibility of NeRF [157, 66, 84, 129, 102, 178]. Some focus on alleviating the need for a large number of input views [64] by applying data augmentation [18] or by introducing regularisations [74, 128, 114, 26]. Other methods focus on disentangling and controlling shape and appearance using adversarial methods [141, 12, 116, 47, 115, 180]. Tancik *et al.* [156] use meta-learning to initialise NeRF weights on image classes, allowing it to converge faster and generalise better. Using this technique they can pre-train a network to represent a scene from a varied set of input images with different lightning conditions and camera settings. Then they can optimise the pre-trained NeRF to a new image with a specific style and render new views with the new style. NeRF-W [97] extends NeRF with appearance and transient embeddings to be able to apply this model to sets of uncontrolled images captured in the wild. NeRF-OSR [135] take this application further and allows the network to have semantically meaningful editing of the scene illumination. Instead of training on ideal images, with completely static content and constant lighting effects, NeRF-W is able to train over input images that suffer from variable illumination, change in colour and exposure, and transient occluders. By rendering conical frustums instead of single-pixel rays, mip-NeRF [4] reduces aliasing artefacts, excessive blurriness, and manages to improve in speed and size with respect to NeRF. NeRV [153] models reflectance and visibility fields to produce arbitrary lighting conditions and indirect illumination effects on the scene renderings. Their MLP estimates surface normals, material parameters, distance to surface intersection, and visibility in any direction, in addition to the volume density. EfficientNeRF [57] aims to improve NeRF performance by improving sampling efficiency and using a caching structure during testing. Regardless, these models require dense input views, are costly, and do not generalise to new scenes.

2.2.7 Zero-shot Novel View Synthesis

To overcome the limitations posed by the scene-specific implicit representation, some approaches have sought to combine the geometry learning capabilities of MVS and IBR techniques with the power of neural rendering models. By doing so, these approaches can acquire geometry information from a limited number of input images and apply it "zero-shot" to novel scenes without requiring time-consuming per-scene training. GRF [162] proposes a geometry-aware attention module, where visual occlusions are implicitly considered, to combine the general 2D local features from multi-views. IBRNet [170] follows concepts of IBR methods by drawing from the source views at render time. The network aggregates 2D feature information from source views along a given ray to compute its final colour. Then a ray transformer is applied along the ray samples to incorporate long-range contextual information to predict the density for each sample. SRF [20] emulates classical stereo matching techniques by learning an ensemble of pair-wise feature similarities. Even though this method works to recover certain geometry and generalise results, it still requires at least 10 inputs to generate pleasant results. In addition, these results are blurry, cannot handle specularities, and the model is generally very expensive to run. PixelNeRF [185] manages to generalise to new scenes using as few as one input image, relying on 2D pixel features and no explicit geometry-aware 3D structures. However, it tends to overfit to the training set, failing to generalise well. On the other hand, PointNeRF [179] leverages 3D information to reconstruct local geometry using neural point clouds. This allows for fast reconstruction of 360 radiance fields from an arbitrary number of input views. However, only the geometry is recovered "zero-shot", needing further training and fine-tuning to retrieve appearance.

MVSNeRF [15] also takes advantage of 3D geometry-aware structures by building a 3D voxel encoding volume based on a feature Plane Sweep Volume [36]. This model works on as few as three input images and is generalisable to different complex scenes. Further developments were made based on geometric constraints [68] and recurrent aggregation [191]. However, in all these systems only the scene content visible from the reference view is well reconstructed. The outputs contain significant artefacts in challenging or occluded regions which require further fine-tuning per scene. These techniques also lack any mechanism to generate image content in areas which are occluded in all inputs.

2.3 Dynamic scene representation

2.3.1 Classical video rendering

Classical video reconstruction methods have focused on the problem of Free Viewpoint Video (FVV) [158], where a dynamic scene captured in real-time can be played back in real-time from a new point of view. These models are usually applied to volumetric performance capture, a multi-view camera acquisition of shape and texture of objects, traditionally captured and experienced from the outside of a 360-degree volume [70]. Early methods follow principles of IBR and MVS. Kanade *et al.* [70] generate new views by triangulating merged depth maps captured with a dome of cameras. Narayanan *et al.* [108] extends the system with explicit surface reconstruction by combining dense depth maps to form a Visible Surface Model. Zitnick *et al.* [195] capture the scene with an 8 camera array and create a pipeline of instance segmentation, segment disparity estimation, and refinement using reprojection and smoothing. Labatut *et al.* [79] attempt to capture more complex and cluttered scenes by modelling a surface graph by applying 3D Delaunay triangulation to a rough scene point cloud estimated by stereo matching. Vlasic *et al.* [167] reconstruct a detailed mesh of a human in different lightning conditions using the photometric normals. Liu *et al.* [89] also use a stereo matching approach to build a point cloud, then merging and downsampling to obtain a mesh and final surface of the object. Using a similar approach, Collet *et al.* [22]

improve results by leveraging infrared, RGB, and silhouette information. Most of these methods require a large number of input views, expensive optimisation, or are applied to the reconstruction of single objects instead of complex scenes.

Zollhofer *et al.* [196] proposed the first real-time template-based approach that can be applied to general objects. They use a custom RGB-D stereo camera to create a coarse volumetric tetrahedral template of the deformable object, and then render object dynamics in real-time. DynamicFusion [110] is the first approach to achieve template-free reconstruction at real-time. VolumeDeform [62] extend this model by using a fine-scale deformation lattice instead of a coarse deformation graph, allowing for higher reconstruction quality. Fusion4D [30] employs a key volume strategy of updating the reference frame, which makes this approach more robust to tracking failures, and able to model large frame-to-frame motion and topology changes. Performance capture systems require either pre-scanned actors, large number of cameras or active sensors [48]. Additionally, volumetric methods lack view dependent effects. Furthermore, inaccuracies in the derived geometry often result in blurred texture maps. These systems fall short of achieving photorealism due to the absence of high-frequency details or embedded textures.

2.3.2 Deep Learning for dynamic scenes

As with the classical approaches, many deep learning dynamic novel view systems concentrate on recovering a single non-rigid 3D object or subject [120, 145, 29, 90, 130, 117]. Martin-Brualla *et al.* [96] perform neural re-rendering over incomplete renderings of human bodies. The application of completion, super resolution, and denoising stages help ameliorate the artefacts in geometry and texture, holes, noise, poor lighting, and low-resolution. Huang *et al.* [58] introduce the use of CNNs to the capture of detailed human shapes from highly sparse camera views without the need for manual image processing, marker tracking, texture cues, or pre-scanned mesh templates. They map 2D images to a 3D volumetric field to encode the probabilistic distribution of surface points of the captured object. Even though they can render bodies from as few as three or four Red, Green and Blue (RGB) cameras, they are still constrained by the use of synthetic datasets without complex backgrounds. DeepDeform [9] proposes a learning-based RGB-D correspondence matching and reconstruction approach based on a Siamese network [166]. Lombardi *et al.* [91] create 3D neural volumes representations of scenes using an encoder-decoder network. They do not explicitly model any temporal dynamics, but the latent space enables them to generate dynamic content.

Bemana *et al.* [6] propose X-Field, a 2D neural field parameterisation that can be interpolated at different view, time or light coordinates. This is implemented as a 2D CNN with hard-coded graphics principles in a differentiable way. Unfortunately, X-Field struggles to recover disoccluded regions, and can show further artefacts created by a lack of training data, even though they require many input camera views. Their approach also struggles with stochastic variation, considerable changes in brightness and other reflectance effects. Yoon *et al.* [184] suggest a self-supervised depth fusion network DFNet. This model combines dense view-dependent monocular depth and sparse view-independent multi-view stereo to explain dynamic scene geometry. However, DFNet requires synchronised annotated data and produces significant artefacts in disocclusions. CaSPR [130] recovers spatio-temporal point cloud representations of dynamically moving objects using an additional embedding which encodes frame information.

2.3.3 Dynamic Neural Radiance Fields

Although the original NeRF [104] is designed for static scenes and takes solely a 3D spatial point and viewing direction as input, it can be easily extended to incorporate temporal variation by conditioning the NeRF on a vector that represents the deformed state (time). The conditioning can be performed implicitly by a time input that can be potentially positionally encoded [82, 176, 37, 31, 126] or learned latent codes per time step [81, 121, 161, 122]. DyNeRF [81] extends NeRF to the time dimension by conditioning the MLP on a learned time-variant latent code. They show this approach performs better than naively adding a temporal dimension to the NeRF input. In this approach, they also propose a more efficient training scheme based on hierarchical training of key-frames, and importance ray sampling to prioritise the learning of the rays that carry more motion information. Even though this approach shows high quality results on long video sequences, they require a grid of 18 time-synchronised and calibrated cameras. Addressing the task of reconstructing dynamic scenes from few shots or monocular videos without prior knowledge of object type or 3D guidance is an extremely ill-posed problem. Consequently, many models are aided by additional geometric regularisers rooted in physical intuitions, and condition learning on additional image-based data like depth maps or optical flow. Several methods learn scene-flow mappings between temporally neighbouring time steps [82, 176, 37, 31]. Li *et al.* [82] propose Neural Scene Flow Fields (NSFFs), a method that models dynamics by predicting a scene flow along the NeRF appearance output. They use this scene flow as a regulariser, warping input frames into the target frame to apply image reconstruction losses. In addition, they enforce other consistency losses on their scene flow by encouraging spatial or temporal smoothness, or forward-backward cycle consistency. To make sure their model does not converge to sub-optimal local minima, they follow an image-based initialisation using depth and optical flow predictions from pre-trained models. NeRFlow [31] models deformations with infinitesimal displacements integrated by Neural ODE to obtain offsets. Their method also applies reconstruction losses to warped frames, and leverages estimated depth maps to supervise the geometry reconstruction, as well as tracking backprojected keypoints in 3D and other regularisations. Although this helps with geometric consistency within the scenes, the quality of the rendering may be hindered by the accuracy of the monocular depth estimation methods used. Furthermore, this method struggles to preserve static backgrounds and handling complex scenes and rendering novel views that differ substantially from the input camera views. Xian *et al.* [176] introduce a soft regularisation loss to constrain static scene content, while Li *et al.* [82] and Gao *et al.* [37] make use of a second static MLP to exploit multi-view background information from frames across time, allowing the dynamic MLP to focus on learning the dynamics of objects. Following up on Xian *et al.* [176], Gao *et al.* [37] train the static NeRF on observations that do not contain moving and deforming parts with the help of a binary segmentation mask which has to be provided by the user. This allows their method to produce more accurate results than its predecessors, but it still suffers from lack of fidelity in handling arbitrary non-rigid deformations, and suffers from a strong reliance on optical flow, and hand-crafted inputs. STaR [189] models motion as a global rigid transformation of a foreground object, decomposing the scene into a static background and a dynamic object. Neural Scene Graphs [119] decomposes the dynamic scene into multiple independent rigidly moving objects via a learned scene graph. Each object and the background are encoded by different neural networks and latent class codes, requiring annotated tracking data for each object and object class.

Instead of the implicit approach of conditioning a NeRF with a time input, the dynamic or deforming neural radiance field can also be modelled explicitly. This is achieved by using a deformation field that can map from the deformed space to a canonical space where the NeRF is embedded. These methods disentangle the dynamics from the geometry and appearance, modelling a separate deformation function. This function acts as a space warping or scene flow which shares information across time into a canonical scene. These approaches tend to have better controllability over their scenes. However, explicit deformation approaches find hard to track complex motions, being only able to represent smaller movements, and cannot handle topological changes well. D-NeRF [126] uses an unregularised ray-bending MLP to model deformations of a single synthetic object. Nerfies [121] instead is applied to casual video footage from a hand-held moving camera of subjects with a complex background to create a few-viewpoint selfie. Their approach conditions a deformation NeRF on an auto-decoded latent code per input view to model the dynamics. They warp the scene into a single canonical frame using this deformation field, where the warped rays are regularised to make motion be piece-wise rigid. This allows them to model small non-rigid deformations well. From the canonical frame they build a template NeRF conditioned on a latent appearance code. HyperNeRF [122] extends this method using a canonical hyperspace instead of a single frame. The deformation network is complemented with an auxiliary ambient slicing surface network that selects a canonical subspace for every input view. The template NeRF is then conditioned not only on the warped canonical rays, but also on the coordinates of the ambient space. This mixture of implicit and explicit representation allows them to handle topological changes such as opening and/or closing of the mouth. Non-rigid NeRF (NR-NeRF) [161] models a monocular dynamic scene using a per-scene canonical volume. This volume is created by warping the deformed volume using a rigidity network and a ray-bending network, to model the static background and the dynamic foreground respectively. Nevertheless, all these methods require per-scene optimisation processes, and they suffer from similar computational limitations as other NeRF models. ENerf [85] uses multi-view depth prediction and learned depth-guided sampling along rays to predict radiance fields near the scene surface. They follow an approach similar to the image case MVSNeRF [15], combining 2D image features and 3D feature volumes to aid reconstruction. They argue their network is so efficient it can do close-to-real-time synthesis in complex scenarios, removing the need to explicitly model dynamics of the scene.

2.4 Summary

In this chapter we have conducted an extensive review of the existing literature surrounding the use of generative Deep Learning (DL) approaches for image inpainting and NVS. First, we have examined how the introduction of learning-based methods has significantly shifted the state-of-the-art in the field of inpainting. Following approaches that take into account both the texture and structure of images lead to generally improved and more consistent image reconstruction. In the specific case of stereo inpainting, little research has been done on the application of DL techniques. The only DL approaches that address this problem to date [16, 94, 93] have focused on artificial or small unrealistic inpainting regions, or do not enforce multi-camera consistency. Thus, our first technical chapter will explore multi-view consistent image reconstruction methodologies to establish a self-supervised pipeline that can take advantage of structural and textural stereo context. We will study the application of such a system across complex geometrically-meaningful inpainting masks that represent object occlusions.

Subsequently, the literature review discussed various techniques and scene representations employed to solve the task of NVS. Recent years have witnessed remarkable growth in this field, largely driven by pioneering photo-realistic approaches such as NeRF [104]. Unfortunately, in its original form, NeRF poses substantial computational limitations which prevents it from being useful in many important applications. On its own, NeRF is very costly to run, requires dense input images, and has to be optimised per scene. In this direction, some approaches [170, 185, 15, 20] have tried to overcome the challenges of this scene-specific implicit representation by drawing from MVS and IBR techniques. However, these solutions are predominantly adept at reconstructing scene content visible from the reference view, with limited capacity to generate image content in regions occluded across all input views. This shortcoming makes them impractical for wide-baseline and very-sparse scene reconstruction tasks where large occlusions are present. In our research, we will examine how to extend these approaches to the context of SVS, by leveraging the power of generative inpainting methods like GANs. GANs help enrich the output variability of generative models, which, in the domain of NVS, has the potential to ensure realistic extrapolation in unobserved regions.

Finally, this literature review has delved into the area of dynamic scene reconstruction and the ongoing efforts to extend NeRF to the temporal dimension. We have seen that using implicit conditioning of NeRFs has the potential to track larger and more complex motions with topological changes. Despite many works tackling this dynamic problem [82, 81], most struggle with the same limitations as the static counterparts: namely the intensive computational resource needs and having a long per-scene optimisation process. Consequently, our final technical chapter will investigate the application of zero-shot techniques to the neural field representation of videos, with the goal of mitigating these limitations.

Chapter 3

SaiNet: Stereo Aware Inpainting Behind Objects with Generative Networks

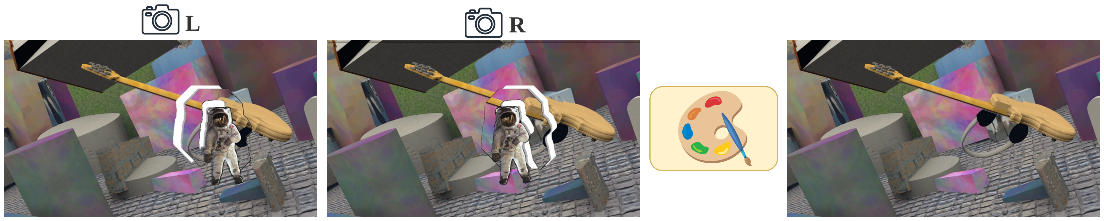


Figure 3.1: **SaiNet:** Inpainting behind objects using geometrically meaningful masks. Visualisation of the inpainting system that extracts a context layer around an object. The surrounding area of a foreground object (astronaut) is used as a guide to inpaint the area behind the object.

3.1 Problem Definition

This thesis research pertains the generation of new camera views given a multi-camera setup. In this scenario, the transition from one camera perspective to another introduces image disocclusions, which are voids of visual data that occur as a result of this viewpoint change. Image inpainting is a technique which finds wide-ranging applications in computer vision and image processing. In the context of our research, it offers a valuable tool for generating realistic and perceptually plausible image fillers for these disocclusions.

Our research is primarily concerned with the practical applications of inpainting techniques to enhance view synthesis in media production. Our approach aims to leverage the benefits of having multiple cameras without necessitating the computational demands associated with more complex and expensive methods that reconstruct an entire 3D scene. Furthermore, we want an approach that can generalise well to unseen scenes and generate creative content without human input. To achieve this, we apply CNNs that can single-handedly propagate structures and textures reasonably.

The initial focus of our research centres on the simpler and under-explored problem of stereo-inpainting. As we have seen in the Literature Review (Section 2.1), traditional monocular techniques tried to achieve inpainting by propagating local image structures and textures [8, 23, 24] or copying patches from the known areas of the image [146, 3]. This worked well for small or narrow regions, but it was prone to generating visual inconsistencies in more significant gaps. Early stereo techniques attempted to generate consistent image output by mechanically warping the available data from the other views [169], or completing the disparity images [105, 106], and then proceeding similarly to the monocular inpainting approaches. However, in recent years, DL techniques have taken advantage of large-scale training data to create more semantically significant inpainting outputs. Some works focused on learning embeddings of the images [123, 60], while others developed different types of convolutional layers to be able to handle more realistic irregular holes [87, 186].

Nevertheless, the only DL techniques that address the stereo inpainting problem [16, 94, 93] to date have focused on artificial or unrealistic inpainting regions, or do not enforce multi-camera consistency. In contrast, our approach focuses on inpainting one target image on geometrically meaningful masks while making use of the information available from the other viewpoint. Our network is based on a UNet [134] architecture with partial convolutions [87], which optimises the use of irregular masks at random locations. Furthermore, we introduce a structure-guided technique to enhance inpainting by leveraging colour edge information [109].

More importantly, we propose a novel stereo-inpainting training mechanism, which transforms the task of "peeking" behind objects into a self-supervised problem. We employ a bank of meaningful and geometrically-consistent 3D object masks, consisting on pairs of context/synthesis regions representing object occlusions and their surrounding visible image context. Previous work may use random image masks, which usually represent the physical damage a picture can suffer, or square unrealistic masks to which the network overfits. Our context/synthesis masks contain more semantic information and are not necessarily bounded within the image, making the inpainting task more realistic than other approaches. Ground-truth training examples are generated from random virtual ("fake") 3D objects placed at random locations in the foreground of the scene, allowing us to have a fully self-supervised stereo training approach. This data augmentation process addresses both the significance of masked regions and the stereo data scarcity problem. Moreover, the resulting model is computationally efficient and able to generalise to previously unknown scenes and occluding objects.

In summary, the contributions of this work are:

- A novel stereo-aware structure-guided inpainting model suitable for efficient novel-view synthesis and free viewpoint applications.
- First inpainting work to take full advantage of stereo-context with geometrically-consistent object masks.
- A novel stereo consistency loss enforcing inpainting results to be consistent with disoccluded information present in other views.

3.2 Method

Our proposed model consists of a deep neural network that follows a UNet-like architecture [134] with skip connections and partial convolutions [87]. Partial convolutions are masked convolutional layers that make sure the inpainted output is only conditioned on

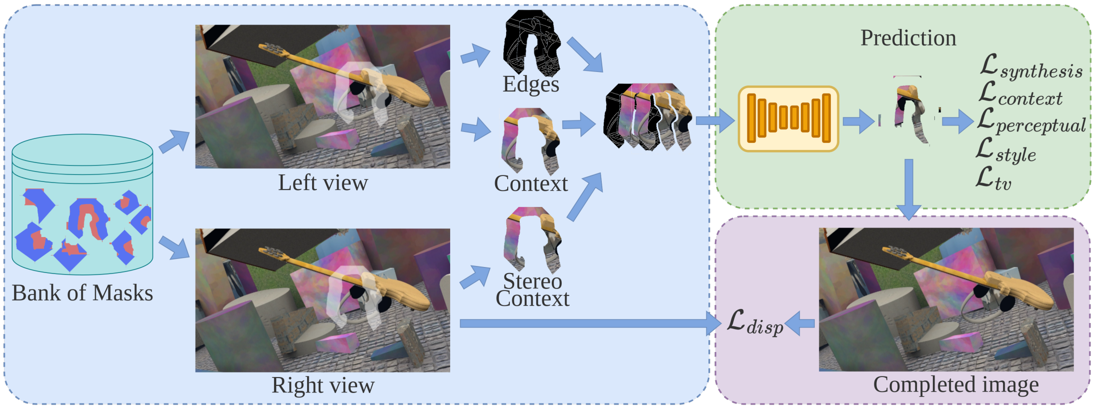


Figure 3.2: **Model overview:** Given a bank of context (blue) and synthesis (red) masks, image context is extracted from both left and right views. These context images are fed into the network along with edge maps to guide the inpainting of the synthesis area. Reconstruction losses as well as a disparity-based loss are applied to the results.

valid pixels. The network takes as input the context and synthesis areas of an object, extracted from pairs of rectified stereo images. The synthesis area is the region behind the object (hole) that the network will inpaint, and the context area is the background surrounding the object which the network will use to inform the reconstruction of the hole. The generation of these areas is explained in Section 3.2.1. In addition, the colour edges are fed into the network for structural guidance (see Section 3.2.3). Finally, to enrich the inpainting and make the network aware of the stereo view, a stereo-context image is added to the input. This context image is generated from the image stereo pair as explained in Section 3.2.2. To further enforce consistency across views, we use a disparity-based stereo loss (Section 3.2.4). An overall visualisation of our proposed model can be seen in Figure 3.2.

3.2.1 Object occlusion regions

An essential part of image inpainting is specifying the type of missing regions that the model needs to handle. Most previous approaches to inpainting have focused on randomly shaped inpainting masks of limited complexity. This is reasonable when dealing with image degradations such as scratches or removing regions containing nuisance objects in 2D. However, for stereo inpainting where we wish to maintain crisp object boundaries, this approach no longer makes sense. We need inpainting masks that represent real image occlusions. Therefore we propose a self-supervised approach where stereoscopic scenes are augmented with geometrically valid inpainting masks, based on a virtual 3D occluding object. This object is hallucinated in a stereo-consistent way over both images, which allows us to collect *"behind the object"* ground-truth data. As such, the network learns to fill in geometrically-meaningful holes with background information, which can then be applied to actual object occlusions in novel view synthesis applications.

We generate these geometrically-valid masks from the MSCOCO dataset [86], a natural scene dataset with a wide diversity in object types and backgrounds. This dataset is unrelated to the ones we use for training, and it only serves to extract pairs of context/synthesis inpainting masks. Once we have a pool of these inpainting masks, we sample and place them on our training datasets, creating a simulated occlusion, and allowing us to extract ground-truth data.

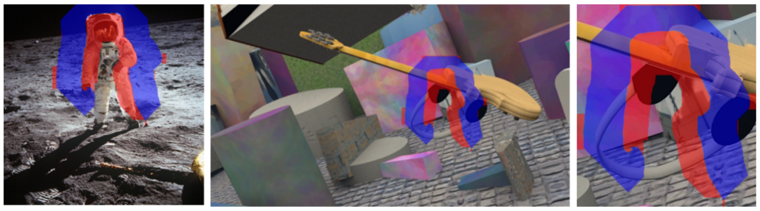


Figure 3.3: **Data generation process.** A collection of context/synthesis regions is created by extracting them from object boundaries in images on the COCO dataset (left). They are then randomly sampled, warped, and pasted onto different images, forming the training dataset of ground truth context/synthesis regions (middle, right).

To create this bank of inpainting masks we follow Shih *et al.* [144] approach, summarised in Algorithm 1. First, a pre-trained monocular depth estimation model [83] is applied on the COCO dataset to obtain pseudo ground truth depth maps. The depth maps are normalised and sharpened using a bilateral median filter, with a 7 × 7 window,

$\sigma_{spatial} = 4.0$ and $\sigma_{intensity} = 0.5$. From these sharp depth maps, depth discontinuities are detected by thresholding the disparity difference between neighbouring pixels, then cleaned up and merged using connected component analysis into a collection of "linked depth edges". The images are disconnected along these depth edges, which represent object boundaries. Context and synthesis regions are generated using a flood-like algorithm to propagate the regions towards the background image (context), and the foreground object area (synthesis) (See Fig. 3.3 for a visualisation of these areas).

These masks are varied and irregular, preventing our model overfitting to one type of mask. Furthermore, as opposed to most methods, our model does not use the whole image as context for the inpainting process, but just the region closer to the object boundaries. Although this reduces the available information that the network can learn from, it allows the network to narrow its attention to the most relevant and meaningful area. However, this poses a more challenging problem, as the context-to-synthesis area ratio is smaller, and the masked regions are not necessarily bounded by context on all sides.

Algorithm 1: Generation of geometrically-valid masks

```
Input: $\mathcal{N} = \{\mathbf{I} : \mathbf{I} \text{ is a natural image}\}$
Output: $\mathcal{M} = \{(\mathbf{C}^{obj}, \mathbf{S}^{obj}) \mid \forall obj \in \mathbf{I}, \forall \mathbf{I} \in \mathcal{N}\}$
for $\mathbf{I}$ in $\mathcal{N}$ do
    $\mathbf{D} = MegaDepth(\mathbf{I})$;
    Pre-process $\mathbf{D}$;
    Find set of discontinuities using $\mathbf{D}$, $d_{\mathbf{I}} \equiv \{d_{\mathbf{I}}^{obj} \mid obj \text{ is an object in image } \mathbf{I}\}$;
    for $d_{\mathbf{I}}^{obj}$ in $d_{\mathbf{I}}$ do
        Propagate background around $d_{\mathbf{I}}^{obj}$ to generate context mask $\mathbf{C}^{obj}$;
        Propagate foreground around $d_{\mathbf{I}}^{obj}$ to generate synthesis mask $\mathbf{S}^{obj}$;
    end
end
```

3.2.2 Stereo awareness

Our approach aims to make inpainting consistent across views in two different ways. One is by enriching the network with extra available information. The other is by enforcing a disparity loss on the output of the network (as explained in Section 3.2.4). The main advantage of having two (or more) cameras is the additional information we can extract to make the task of inpainting “unknown” areas easier. For example, some colours or textures may be completely occluded by the object in one view, but still be partially visible from the other view. In this case, the additional input can provide strong cues for the network to inpaint the occluded region. We make our system stereo-aware by providing this extra information as input to the network by warping the context mask of each object based on its estimated depth value into the additionally available view (See Algorithm 2). We use PSMNet [13] to estimate this depth and select the closest depth value to make sure the object is situated at the front of all other objects in the scene. In other words, we extract contextual information around the boundary of the occluding object, in both views. Then we feed this extra context into the network to learn to use it in filling in the synthesis area. As a result, we aid the inpainting process by enriching the texture and colour information available.

3.2.3 Structural guidance

Several methods [109, 132, 152] have shown that structure-guided inpainting performs better at reconstructing high frequency information accurately. Since image structure is well-represented in its edge mask, superior results can be obtained by conditioning an image inpainting network on edges in the missing regions. For this reason, we feed the edge maps generated using Canny edge detector [11] along with the colour information, as a conditioning information to our network, following a similar process to Nazeri *et al.* [109]. At test time, we estimate the edges using a pre-trained EdgeConnect [109] model. While a comprehensive end-to-end model could be trained to predict edges, introducing the potential for the model to generate more accurate structures, this approach may also compromise inpainting accuracy as the model tackles the complexities of a more intricate problem.

Algorithm 2: Stereo-aware training set generator

```
Input: $\mathcal{D} = \{(\mathbf{I}_L, \mathbf{I}_R)$ : stereo pair of images$\}$
$\mathcal{M} = \{(\mathbf{C}^i, \mathbf{S}^i) \mid \forall$ object $i\}$
Output: Training_set = $\{(\mathbf{CC}_L, \mathbf{CS}_L, \mathbf{E}_L, \mathbf{CC}_R) \mid \forall(\mathbf{I}_L, \mathbf{I}_R) \in \mathcal{D}\}$
for $(\mathbf{I}_L, \mathbf{I}_R)$ in $\mathcal{D}$ do
    1. Select random context and synthesis mask pairs $(\mathbf{C}^i, \mathbf{S}^i)$ from the mask bank $\mathcal{M}$;
    2. Select a random position $x, y$ to situate the object at $\mathbf{I}_L$;
    3. Crop image $\mathbf{I}_L$ at $x, y$ with mask $\mathbf{C}^i$ to generate colour context region $\mathbf{CC}_L$;
    4. Crop image $\mathbf{I}_L$ at $x, y$ with mask $\mathbf{S}^i$ to generate colour synthesis region $\mathbf{CS}_L$;
    5. Generate edge map $\mathbf{E}_L = \text{Canny}(\mathbf{CC}_L + \mathbf{CS}_L)$;
    6. Estimate depth map $\mathbf{D}_L = \text{PSMNet}(\mathbf{I}_L, \mathbf{I}_R)$;
    7. Crop image $\mathbf{D}_L$ at $x, y$ with mask $\mathbf{S}^i$ to generate depth synthesis region $\mathbf{DS}_L$;
    8. depth = $\min(\mathbf{DS}_L)$;
    9. Reproject $\mathbf{C}^i$ using depth value onto $\mathbf{I}_R$ and crop to generate stereo colour context region $\mathbf{CC}_R$;
end
```

3.2.4 Stereo consistency loss

The second way in which we enforce stereo consistency on our inpainted results is by introducing a disparity loss. We propose a local consistency loss which measures the consistency between the inpainted area in one view, and the ground truth in the other view. In this way, we encourage the system to use the stereo context; inpainting not just any perceptually acceptable background, but specifically the one consistent with any partial observations. The loss is inspired by the work of Chen *et al.* [16] and is illustrated in Fig. 3.2. Note that this loss makes the Lambertian assumption, as it is not able to represent view dependent effects like reflectance. We deem this limitation acceptable, given that we are working with stereo datasets with a limited baseline change and therefore very minimal non-Lambertian effects.

We compare a patch $P\left(i\right)$ around every pixel i in the inpainted area $\mathbf{S} \odot \mathbf{I}$ against a patch centred on the corresponding pixel on the other view. $\mathbf{S}$ is the binary mask indicating the synthesis region, $\mathbf{I}$ is the inpainted image, and $\odot$ denotes the Hadamard product.

$$\mathcal{L}_{\text{disp}} = \frac{1}{|\mathbf{S}|} \sum_{i \in \mathbf{S} \odot \mathbf{I}} \overleftarrow{\text{cost}}\left(i\right), \tag{3.1}$$

$$\overleftarrow{\text{cost}}\left(i\right) = 1 - \Phi\left(P\left(i\right), P\left(\overleftarrow{W}\left(i\right)\right)\right) \tag{3.2}$$

where $\overleftarrow{W}$ is the warping function corresponding to a change from source to target view, using the disparity estimated by PSMNet [13]. We use a Normalised Cross-Correlation (NCC) as our stereo matching cost (Φ) which works well with back-propagation.

$$\Phi(X, Y) = \frac{\|X \odot Y\|_{1,1}}{\|X\|_F \, \|Y\|_F} \tag{3.3}$$

here $\|\cdot\|_{1,1}$ and $\|\cdot\|_F$ are the 1-entrywise and Frobenious matrix norms respectively.

3.2.5 Inpainting losses

In addition to the disparity loss, other per-pixel similarity losses and losses based on deep features are used to enforce perceptually realistic results. First, two per-pixel reconstruction losses are defined over the synthesis and context regions, these losses help to guide the inpainting of the missing areas, as well as making sure that context and synthesis areas are recovered consistently and with smooth boundaries.

$$\mathcal{L}_{\text{synthesis}} = \frac{1}{N_{\mathbf{I}_{\text{gt}}}} \left\|\mathbf{S} \odot \left(\mathbf{I} - \mathbf{I}_{\text{gt}}\right)\right\|_1, \tag{3.4}$$

$$\mathcal{L}_{\text{context}} = \frac{1}{N_{\mathbf{I}_{\text{gt}}}} \left\|\mathbf{C} \odot \left(\mathbf{I} - \mathbf{I}_{\text{gt}}\right)\right\|_1 \tag{3.5}$$

where $\mathbf{S}$ and $\mathbf{C}$ are the binary masks indicating synthesis and context regions respectively, $N_{\mathbf{I}_{\text{gt}}}$ is the total number of pixels, $\mathbf{I}$ is the inpainted result, and $\mathbf{I}_{\text{gt}}$ is the ground truth image.

We also include two deep feature losses from Gatys *et al.* [38], based on VGG-16 [147] embeddings, that measure high-level perceptual and semantic differences. Firstly,

$$\mathcal{L}_{\text{perceptual}} = \sum_{p=0}^{P-1} \frac{\left\|\Psi_p\left(\mathbf{I}\right) - \Psi_p\left(\mathbf{I}_{\text{gt}}\right)\right\|_1}{N_{\Psi_p}} \tag{3.6}$$

where $\Psi_p(\cdot)$ is the output of the p'th layer from VGG-16 [147], and N_{Ψ_p} is the total number of elements in the layer. This perceptual loss encourages the network to create images with similar content and similar feature representations. Secondly, the style loss is defined as,

$$\mathcal{L}_{\text{style}} = \sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p \left[\left(\Psi_p \left(\mathbf{I} \right) \right)^{\intercal} \Psi_p \left(\mathbf{I} \right) - \left(\Psi_p \left(\mathbf{I}_{\text{gt}} \right) \right)^{\intercal} \Psi_p \left(\mathbf{I}_{\text{gt}} \right) \right] \right\|_1 \tag{3.7}$$

where $K_p = \frac{1}{C_p H_p W_p}$ is a normalisation factor, and C_p, H_p, W_p are the number of channels, height, and width of the output $\Psi_p(\cdot)$. The style loss is similar to the perceptual loss which compares images at the feature level, but performing an autocorrelation (Gram matrix) on each feature map. This allows the loss to capture information about which features tend to activate together, ensuring the style of the output images resemble the input in colour, textures, common patterns, etc.

Finally, a total variation loss is used as a smooth regularization.

$$\mathcal{L}_{\text{tv}} = \sum_{(i,j) \in \mathbf{S}} \frac{\|\mathbf{I}(i, j+1) - \mathbf{I}(i,j)\|_1}{N_{\mathbf{I}_{\text{gt}}}} + \sum_{(i,j) \in \mathbf{S}} \frac{\|\mathbf{I}(i+1, j) - \mathbf{I}(i,j)\|_1}{N_{\mathbf{I}_{\text{gt}}}} \tag{3.8}$$

where the $\mathbf{S}$ denotes the pixels in the synthesis region.

Similar to Liu *et al.* [87], we use the following weights to combine all these losses to yield the final training objective: $\lambda_{\text{synthesis}} = 6$, $\lambda_{\text{context}} = 1$, $\lambda_{\text{perceptual}} = 0.05$, $\lambda_{\text{style}} = 120$, $\lambda_{\text{tv}} = 0.1$, $\lambda_{\text{disparity}} = 0.1$. The same parameters are used for all evaluations.

3.3 Implementation

Our model follows a UNet [134] architecture where the convolutional layers are partial convolutions [87]. These partial convolutions are masked and renormalised convolutional layers that are only conditioned on valid pixels, to avoid invalid initialisation values in a hole region to affect the inpainted outcome. The upsampling layers at the decoding stage use nearest neighbour, and we keep the skip connections to propagate information across layers. Further details of the layers can be found in Table 3.1. At train time, we generate edges using Canny edge detector [11], while at test time we predict edges using the EdgeConnect model [109] pre-trained on Places2 [192]. For disparity estimation we use a pre-trained PSMNet [13] model over SceneFlow [98] dataset. An overview of the test-time pipeline can be seen in Figure 3.4.

Table 3.1: **Model architecture.** PConvX denotes the X Partial Convolution layer. BN is Batch Normalisation. The context and synthesis regions are combined to create the partial masks for the PConvX layers.

Module	Filter Size	#Channels	Dilation	Stride	Norm	NonLinearity
PConv1	7 × 7	64	1	2	-	ReLU
PConv2	5 × 5	128	1	2	BN	ReLU
PConv3	5 × 5	256	1	2	BN	ReLU
PConv4	3 × 3	512	1	2	BN	ReLU
PConv5	3 × 3	512	1	2	BN	ReLU
PConv6	3 × 3	512	1	2	BN	ReLU
PConv7	3 × 3	512	1	2	BN	ReLU
PConv8	3 × 3	512	1	2	BN	ReLU
NearestUpsample	-	512	-	2	-	-
Concatenate (w/ PConv7)	-	512+512	-	-	-	-
PConv9	3 × 3	512	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	512	-	2	-	-
Concatenate (w/ PConv6)	-	512+512	-	-	-	-
PConv10	3 × 3	512	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	512	-	2	-	-
Concatenate (w/ PConv5)	-	512+512	-	-	-	-
PConv11	3 × 3	512	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	512	-	2	-	-
Concatenate (w/ PConv4)	-	512+512	-	-	-	-
PConv12	3 × 3	512	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	512	-	2	-	-
Concatenate (w/ PConv3)	-	512+256	-	-	-	-
PConv13	3 × 3	256	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	256	-	2	-	-
Concatenate (w/ PConv2)	-	256+128	-	-	-	-
PConv14	3 × 3	128	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	128	-	2	-	-
Concatenate (w/ PConv1)	-	128+64	-	-	-	-
PConv15	3 × 3	64	1	1	BN	LeakyReLU(0.2)
NearestUpsample	-	64	-	2	-	-
Concatenate (w/ Input)	-	64+9	-	-	-	-
PConv16	3 × 3	3	1	1	-	-

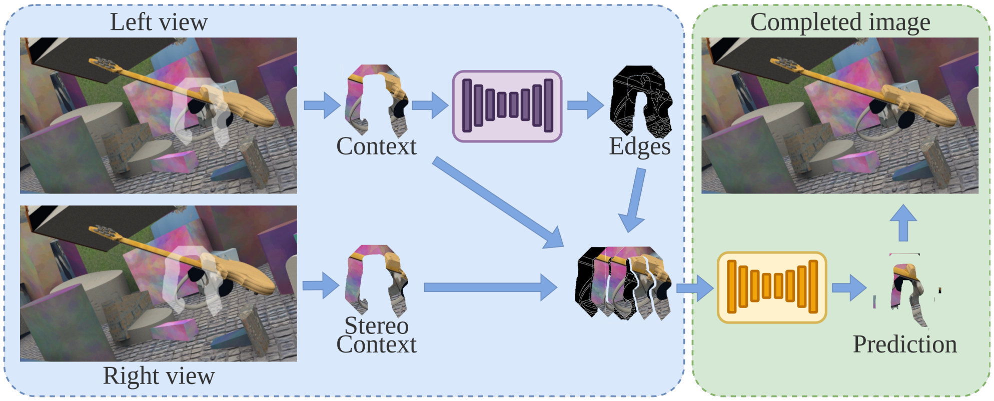


Figure 3.4: **Test pipeline:** At test time the edge maps are predicted using an edge detection model [109], while at training time the model is provided with ground truth edges generated with Canny [11].

3.4 Evaluation

3.4.1 Datasets

Good quality, natural stereo datasets are very hard to come by. This is a problem for training deep neural networks, which usually require a high number of images to extract meaningful statistical information. Our approach to data collection intrinsically performs data augmentation, as the random sampling of context-synthesis areas makes it possible to use different samplings of the same image without overfitting.

For training we have used three different datasets: *FlyingThings3D* and *Driving* from the SceneFlow [98] dataset, and *Middlebury* [139]. *FlyingThings3D* consists of 21,818 frames from 2,247 scenes, containing everyday objects flying around in a randomised way. This is ideal for training CNNs due to the large amount of data and variety of objects. *Driving* is a more naturalistic-looking dynamic street scene resembling the *KITTI* dataset [39]. It contains 4400 images from one scene. On the other hand, the *Middlebury* dataset consists of only 33 pairs of stereo images of natural scenes. Even though this dataset is not big enough to train a DL model, we are able to perform transfer learning and generate pleasant results over real world data (See Fig. 3.5). Both SceneFlow artificial datasets are rendered with a baseline translation and no rotation between cameras, while the natural Middlebury dataset has been rectified to put both views on the same camera plane.

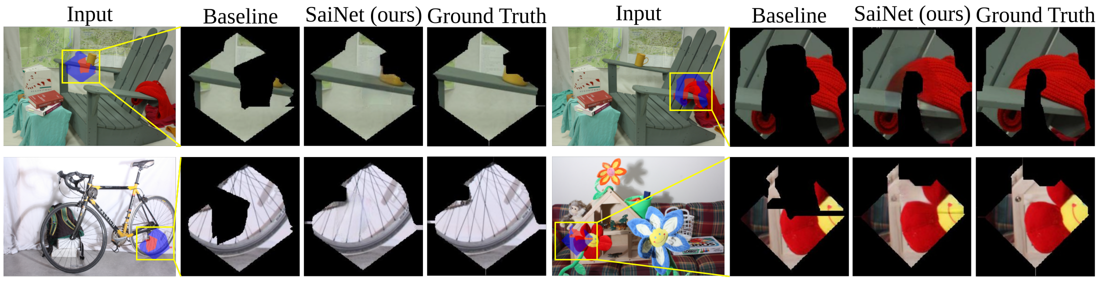


Figure 3.5: **Real dataset evaluation** over Middlebury [139] using Canny edge detector. The zoomed-in crop of yellow area is visualised as “Ground Truth”.

These datasets contain ground truth disparity maps, but for our model we have included a disparity estimation step using PSMNet [13] so we do not rely on existing ground truth data. This makes it fairer to compare to other models that use a similar approach, as well as being more relevant to our application to media production, where we may have several views from the same scene, but no depth information.

To generate the pool of geometrically-meaningful masks we follow the process as in Shih *et al.* [144], using the COCO [86] dataset to generate the context and synthesis regions. The COCO dataset contains around 120k images, and has a large diversity of object types and scenes. We sample at most 3 pairs of context-synthesis regions per image.

3.4.2 Experiment setup

The network is trained using a batch size of 8 and 256 × 256 images. The model is optimised using Adam optimiser [75] and a learning rate of $1e-3$. A model has been trained for each different dataset. As *FlyingThings3D* is 3 to 5 times bigger than the other datasets, a transfer learning approach has been followed where the model is trained on *FlyingThings3D* first and then fine-tuned over *Driving*, and *Middlebury*.

For fair comparison to the results of Chen *et al.* [16] and Ma *et al.* [94], we have trained our *Driving* model using 128 × 128 square context masks and 64 × 64 centred synthesis masks. Our baseline model is the 3D photography colour inpainting network by Shih *et* *al.* [144], which has been trained in the same fashion as our model, and conditioned over depth edges instead of colour as per their original pipeline.

For training, we generate edge maps using Canny [11] edge detector following the EdgeConnect [109] approach. At test time, we apply pre-trained EdgeConnect models to generate the synthesis area edges, using the pre-trained model over Places2 [192] for our *FlyingThings3D*, and *Middlebury*, and a pre-trained model over Paris StreetView [28] for our *Driving*.

3.4.3 Evaluation of accuracy

There is no perfect numerical metric to evaluate image inpainting outputs given the variety of possible valid results. In order to assess the performance of our model, we employ a range of widely recognised metrics that evaluate various aspects of an image. To measure image quality we make use of the Peak Signal-to-Noise Ratio (PSNR) [59] and the Structural Similarity Index (SSIM) [171] index. PSNR serves as an indicator of the overall consistency of pixels, while SSIM gauges the coherency of local structures. We define PSNR as

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_C^2}{\text{MSE}\left(\hat{C}^b(\mathbf{r}), C(\mathbf{r})\right)} \right) \tag{3.9}$$

$$\text{MSE}\left(\hat{C}^b(\mathbf{r}), C(\mathbf{r})\right) = \frac{1}{N} \sum_{\mathbf{r}} [\hat{C}^b(\mathbf{r}) - C(\mathbf{r})]^2 \tag{3.10}$$

where MAX_C is the maximum possible input value, and $\text{MSE}\left(\hat{C}^b(\mathbf{r}), C(\mathbf{r})\right)$ represents the per-pixel Maximum Squared Error between the predicted colour $\hat{C}^b(\mathbf{r})$ at ray $\mathbf{r}$, and the original colour $C(\mathbf{r})$, in a batch of N rays.

On the other hand, SSIM is given by

$$\text{SSIM}(\hat{C}^b, C) = \frac{(2\mu_{\hat{C}^b}\mu_C + k_1)(2\sigma_{\hat{C}^b}\sigma_C + k_2)}{(\mu^2_{\hat{C}^b} + \mu^2_C + k_1)(\sigma^2_{\hat{C}^b} + \sigma^2_C + k_2)} \tag{3.11}$$

where $k_1 = 0.01^2$ and $k_2 = 0.03^2$ are variables to stabilise the operation. We use a window size of 5 for the Gaussian kernel to smooth the images.

It is worth noting that these metrics assume independence among pixels, which can result in favourable scores for visually inaccurate outcomes. Consequently, we also incorporate the application of a Learned Perceptual Image Patch Similarity (LPIPS) [190] metric, which endeavours to capture human perception by leveraging deep features. We use the default settings for the implementation based on AlexNet [77].

The stereo consistency is quantified using the disparity error metric from [94], which counts the erroneous pixels of the PSMNet [13] estimated disparity map of the inpainted image, compared against the ground truth[1]. Given the inpainted image $\mathbf{I}$, for every pixel i we consider its estimated disparity d^i_{est} to be erroneous iff the absolute error against the equivalent pixel in the ground truth disparity image d^i_{gt} is greater than p_1 and its relative error greater than p_2 (we use $p_1 = 3$ and $p_2 = 0.05$). This is described in equation 3.12, where N is the total number of pixels, and [ ] is the Iverson bracket.

$$\text{DispE} = \frac{1}{N} \sum_{i \in \mathbf{I}} \left[\left(\left| d^i_{\text{est}} - d^i_{\text{gt}} \right| > p_1 \right) \ \& \ \left(\frac{\left| d^i_{\text{est}} - d^i_{\text{gt}} \right|}{d^i_{\text{gt}}} > p_2 \right) \right] \tag{3.12}$$

3.4.4 Results

Table 3.2: **Quantitative results.** Image quality & stereo consistency of different models. **Bold** is best. $\star$ values are from their paper.

Dataset	Model	PSNR↑	SSIM↑	LPIPS↓	DispE (%)↓
FlyingThings3D	Shih *et al.* [144]	28.32	0.8589	0.0707	7.96
	Ours	**30.50**	**0.8643**	**0.0556**	**7.67**
Driving	Shih *et al.* [144]	30.46	0.969	0.1141	9.94
	Chen *et al.* [16]*	22.38	0.959	-	7.79
	Ma *et al.* [94]*	23.20	0.964	-	**4.72**
	Ours	**34.94**	**0.977**	**0.0628**	8.01

We perform a quantitative comparison of SaiNet against other state-of-the-art methods [144, 16, 94] in Table 3.2. We can observe our model performs better across all metrics compared with the baseline model of Shih *et al.* [144]. Our model also performs competitively against other stereo inpainting models [16, 94], showing a superior image

[1] The definition of [94] has a typo where the absolute error $\left| d^i_{\text{est}} - d^i_{\text{gt}} \right|$ is replaced by d^i_{est}.

inpainting quality with an improvement on PSNR values of 50%, and some improvement to SSIM. Due to the nature of our mask generation process, our stereo context information is quite narrow, limiting the visible area that our network can learn from. Despite this, our model accomplishes similar results to the stereo consistency of Chen *et al.* [16]. The image quality of the inpainting is superior on the Driving dataset, which was trained using square centred masks to match the experimental setup of [16, 94]. Meanwhile, the object-like occlusion masks used on the FlyingThings3D dataset, which are not fully bounded, are much more challenging.

A qualitative example is shown in Figure 3.6. Despite having access to depth edges, Shi *et al.* struggles to produce sharp object boundaries in the inpainted region. Meanwhile SaiNet is able to use stereo context to inpaint sharp boundaries using colour edge information. This is evidenced by Shih *et al.* success recovering the green bar in the first example, but failing on the colour edge of the second example. However, as shown in the 4th example, our technique still struggles to inpaint especially intricate structures which are not visible through stereo context. Nevertheless it produces sharper and more visually pleasing results.

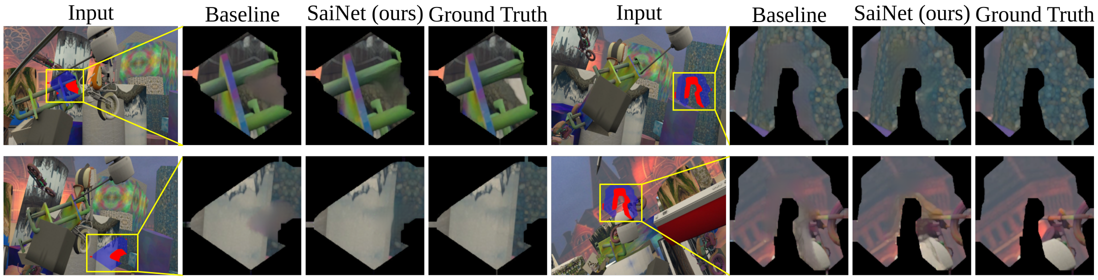


Figure 3.6: **Qualitative inpainting results** for FlyingThings3D. Baseline is Shi *et al.* [144]. The zoomed-in crop of the yellow area is visualised in the "Ground Truth" column.

We present further inpainting examples that document the improvements from the baseline, and our model's general performance and limitations. In Figures 3.8, 3.9, 3.10, 3.11, 3.12 we can see the results compared against the baseline of Shih *et al.* [144] over the FlyingThings3D dataset [98]. We show the full image with the synthesis area highlighted in red for reference, while the context crop is what we actually feed into our network. On the other hand, in Figure 3.13 we present how well the inpainted results blend in the full image, as our context regions are quite small, and we care about the visual consistency of the final result over the whole image. In a similar fashion, we exhibit qualitative results over the *Driving* dataset [98] in Figures 3.14, 3.15, and 3.16, as well as full image blending in Figure 3.17.

As the ratio of synthesis to context area decreases, the quality of the inpainted regions increases. In other words, the model performs better at inpainting smaller areas, when it has a lot of context. Figure 3.7 shows the performance of the model on a 256 × 256 image with different size synthesis areas. Our approach deals with object synthesis regions that are not necessarily bounded by context images on all sides, this makes the inpainting harder, as the context effect on the inpainted output weakens with distance.

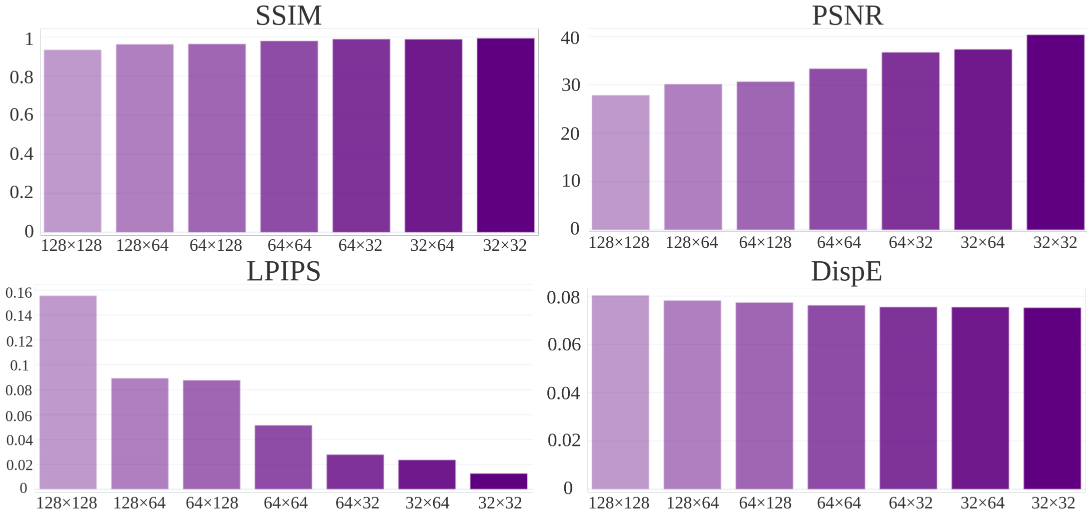


Figure 3.7: **Performance vs hole size.** Metrics calculated over different synthesis region sizes on 256 × 256 context images.

3.4.5 Ablation Study

In the interest of proving the contribution of every stage to the accuracy of the model, we have studied its performance removing the key contributions. Results presented in Table 3.3 show that every part of the model performs better than the baseline, with the combination of all modules having the best performance across all metrics. It is interesting to note that the use of a disparity loss provides the largest individual benefit in terms of stereo consistency.

Table 3.3: **Ablation study.** Compare the accuracy of different stages of the model over all regions. 'Baseline' is the monocular inpainting model, 'Stereo' is the model + stereo context, 'Disp' is the model + disparity loss, and 'Full' is the model with both stereo context and disparity loss. **Bold** is best result. Blue are results in synthesis regions only.

Model	PSNR↑	SSIM↑	LPIPS↓	DispE (%)↓
Baseline	28.32 (22.26)	0.8589 (0.5619)	0.0707 (0.0676)	7.96
Ours (Stereo)	29.41 (23.61)	0.8604 (0.5597)	0.0625 (0.0582)	7.68
Ours (Disp)	29.79 (24.00)	0.8619 (0.5684)	0.0570 (0.0569)	7.71
Ours (Full)	**30.50** (**24.70**)	**0.8643** (**0.5771**)	**0.0556** (**0.0539**)	**7.67**

3.5 Conclusion

In this chapter we have presented SaiNet, a novel learned stereo-aware inpainting model that addresses the challenge of recovering missing image information while enforcing essential stereo consistency in its output. This approach is useful for applications in media production and image editing, as it allows us to see behind objects when moving a camera point of view. To achieve this, SaiNet is trained in a self-supervised fashion over geometrically meaningful masks representing object occlusions. Stereo-consistency is enforced by leveraging vital stereo-context and a disparity based loss. It is worth noting that our problem's complexity extends beyond the scope of existing models. Our context-to-mask ratio is smaller than similar approaches, and our masks are not fully bounded by image content. Despite this, SaiNet shows substantial performance improvement in comparison to existing state-of-the-art alternatives, showcasing a notable increase of up to 50% in PSNR. We demonstrated its performance by evaluating over several diverse datasets. While our model presents a significant advancement in consistent image inpainting, there exist limitations that need further exploration. In particular, refining the application of the stereo context and loss to the model could help attain more consistency across views. It would also be interesting to explore more comprehensively the impact of the masks shape, size, and location within the image.

More crucially to our intended application, the main limitation of this approach lies in having to balance stereo consistency against realism and accuracy of the results. Other more powerful inpainting techniques, such as recent adversarial networks, may be more equipped to generate more visually realistic content, but they still necessitate conditioning on multiple camera inputs to achieve consistency. Given that our application pertains to media production from few camera views, there's a potential to explore systems that offer greater flexibility and can replicate more complex camera setups beyond the stereo scenario. Our stereo inpainting method could be applied to layered scene representations where camera movements are small and parallel. In contrast, real scenarios often involve more diverse camera arrangements with significant translations and rotations. Our current system uses a consistency loss that models the stereo displacement, and thus is not able to generalise to other arrangements. This might require a shift toward representations that can model scenes directly in 3D, which would be intrinsically stereo-consistent and may be more naturally equipped to solve our more general problem.

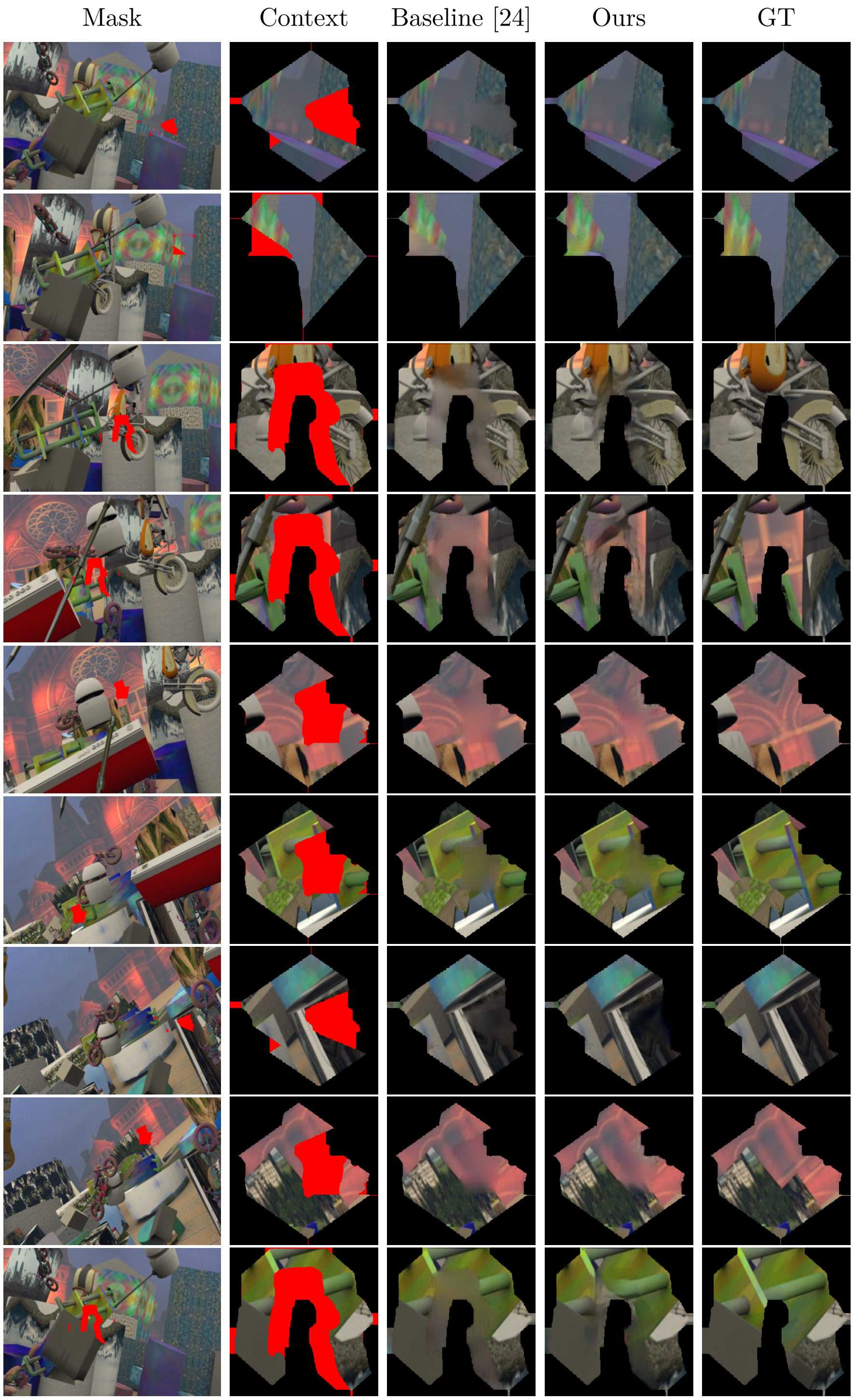


Figure 3.8: **Visual comparison. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Baseline:** Shih *et al.* [144]. **Ours:** our inpainted results. **GT:** ground truth. **Dataset:** *FlyingThings3D* [98].

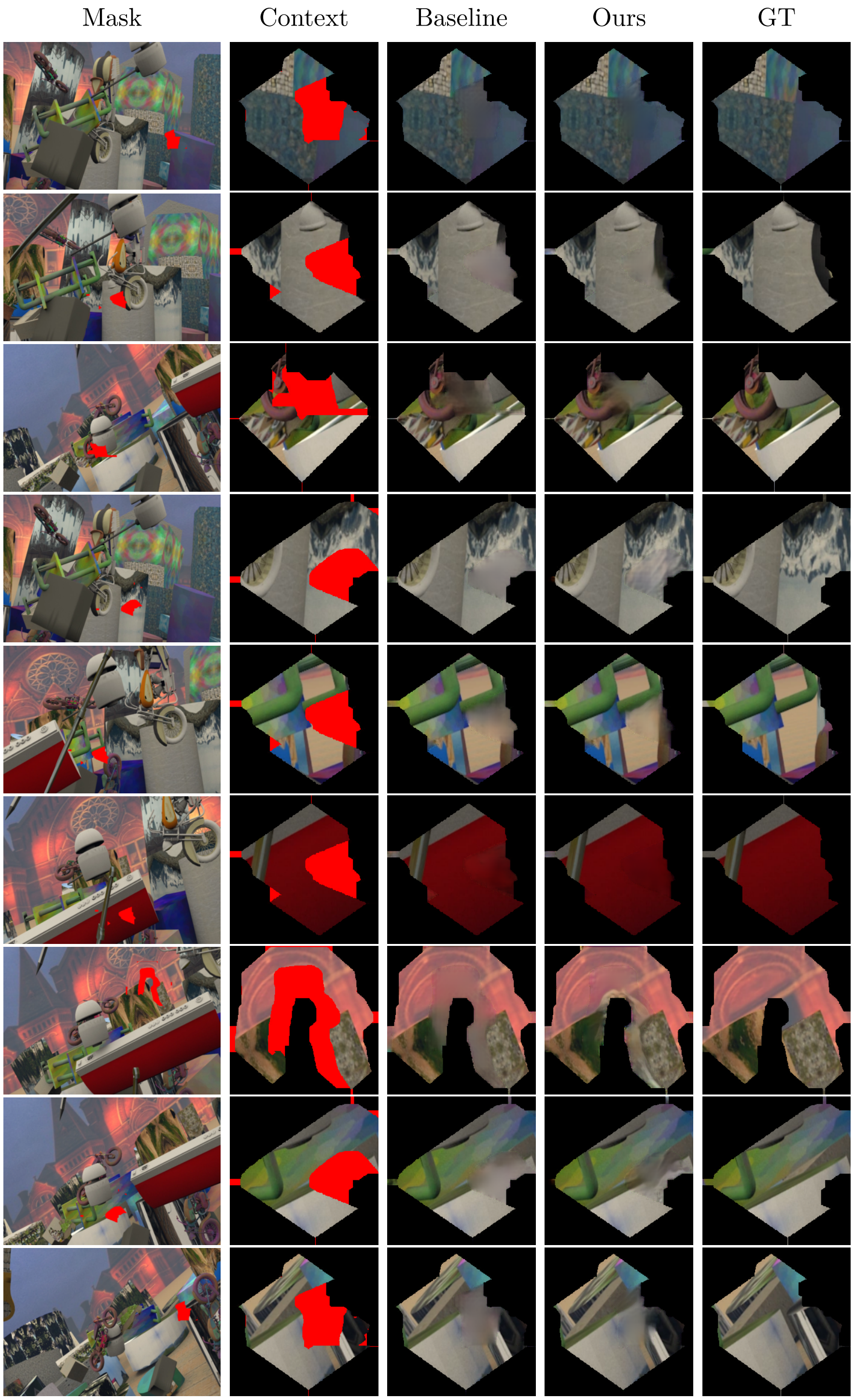


Figure 3.9: **Visual comparison. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Baseline:** Shih *et al.* [144]. **Ours:** our inpainted results. **GT:** ground truth. **Dataset:** *FlyingThings3D* [98].

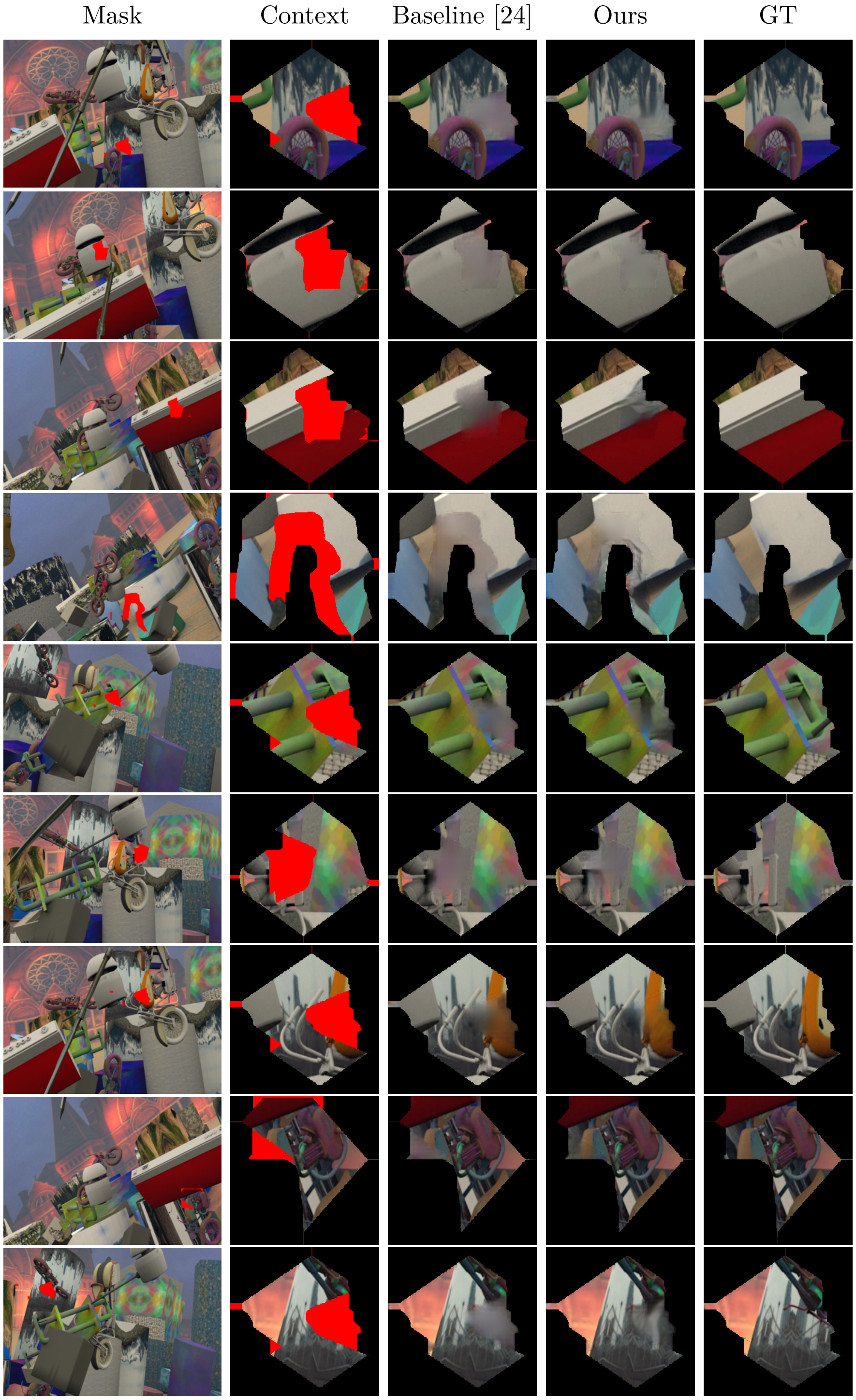


Figure 3.10: **Visual comparison. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Baseline:** Shih *et al.* [144]. **Ours:** our inpainted results. **GT:** ground truth. **Dataset:** *FlyingThings3D* [98].

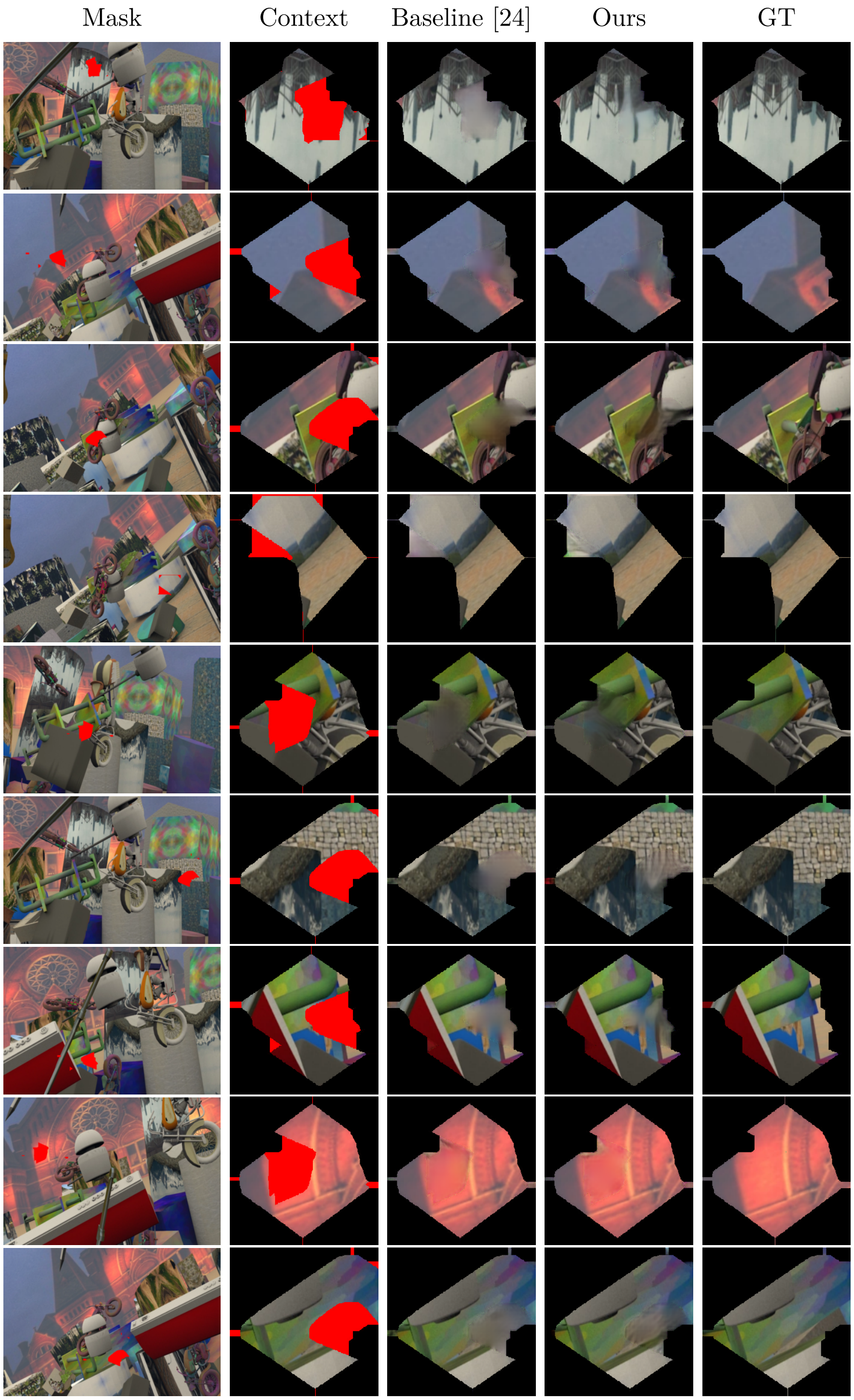


Figure 3.11: **Visual comparison. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Baseline:** Shih *et al.* [144]. **Ours:** our inpainted results. **GT:** ground truth. **Dataset:** *FlyingThings3D* [98].

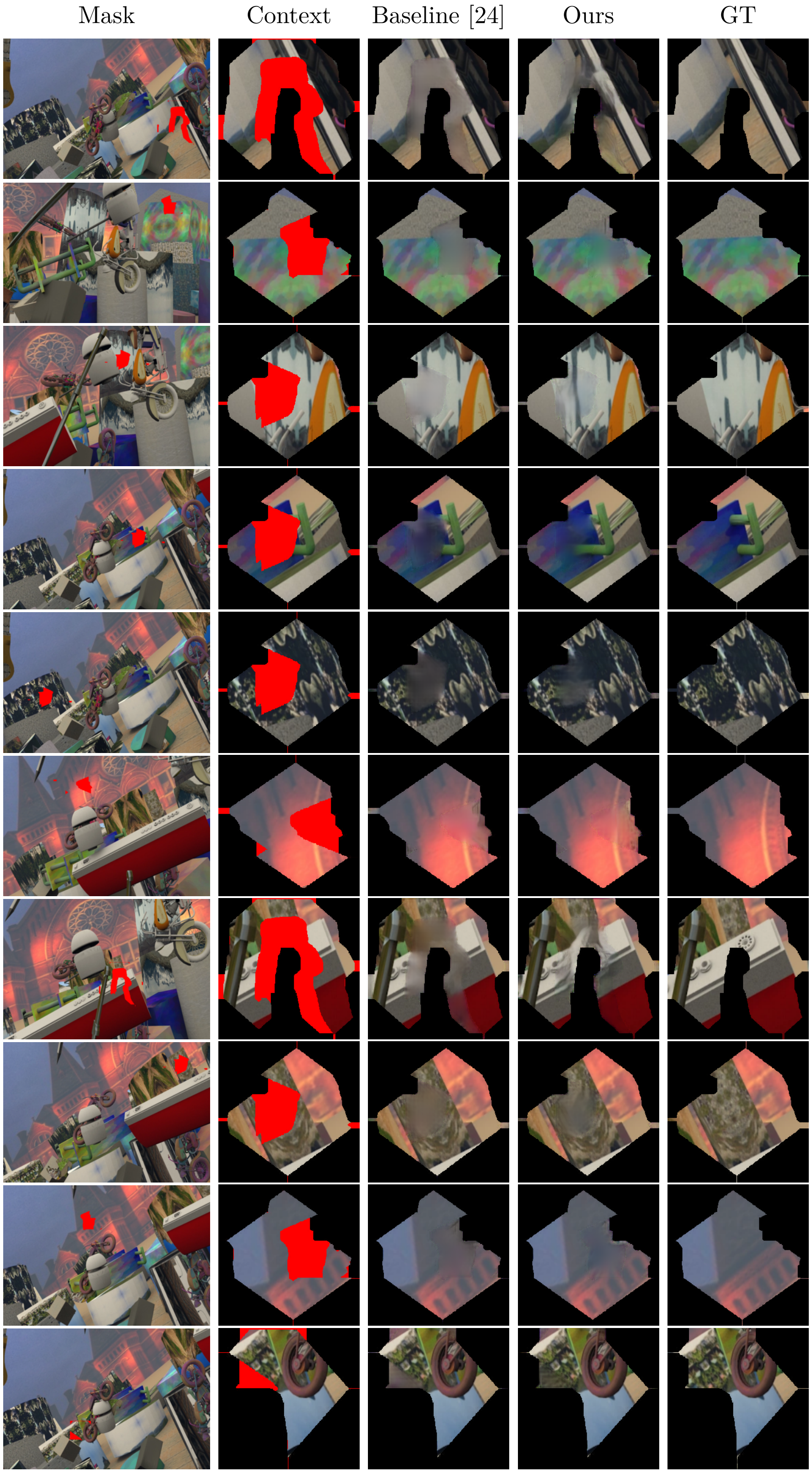


Figure 3.12: **Visual comparison. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Baseline:** Shih *et al.* [144]. **Ours:** our inpainted results. **GT:** ground truth. **Dataset:** *FlyingThings3D* [98].

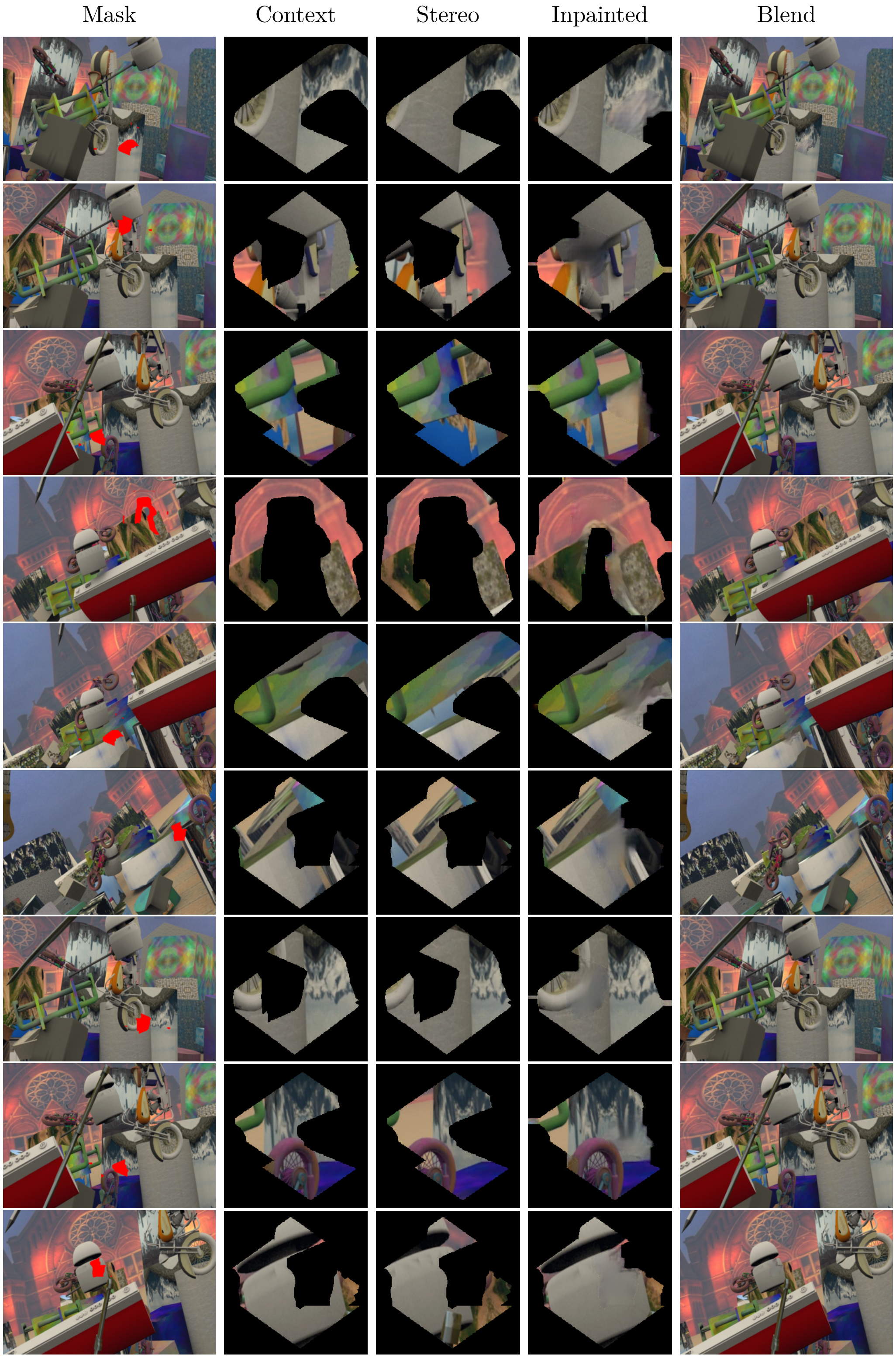


Figure 3.13: **Model performance. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Stereo:** RGB crop fed into the network. **Inpainted:** our inpainted results. **Blend:** Target image with our inpainted region. **Dataset:** *FlyingThings3D* [98].

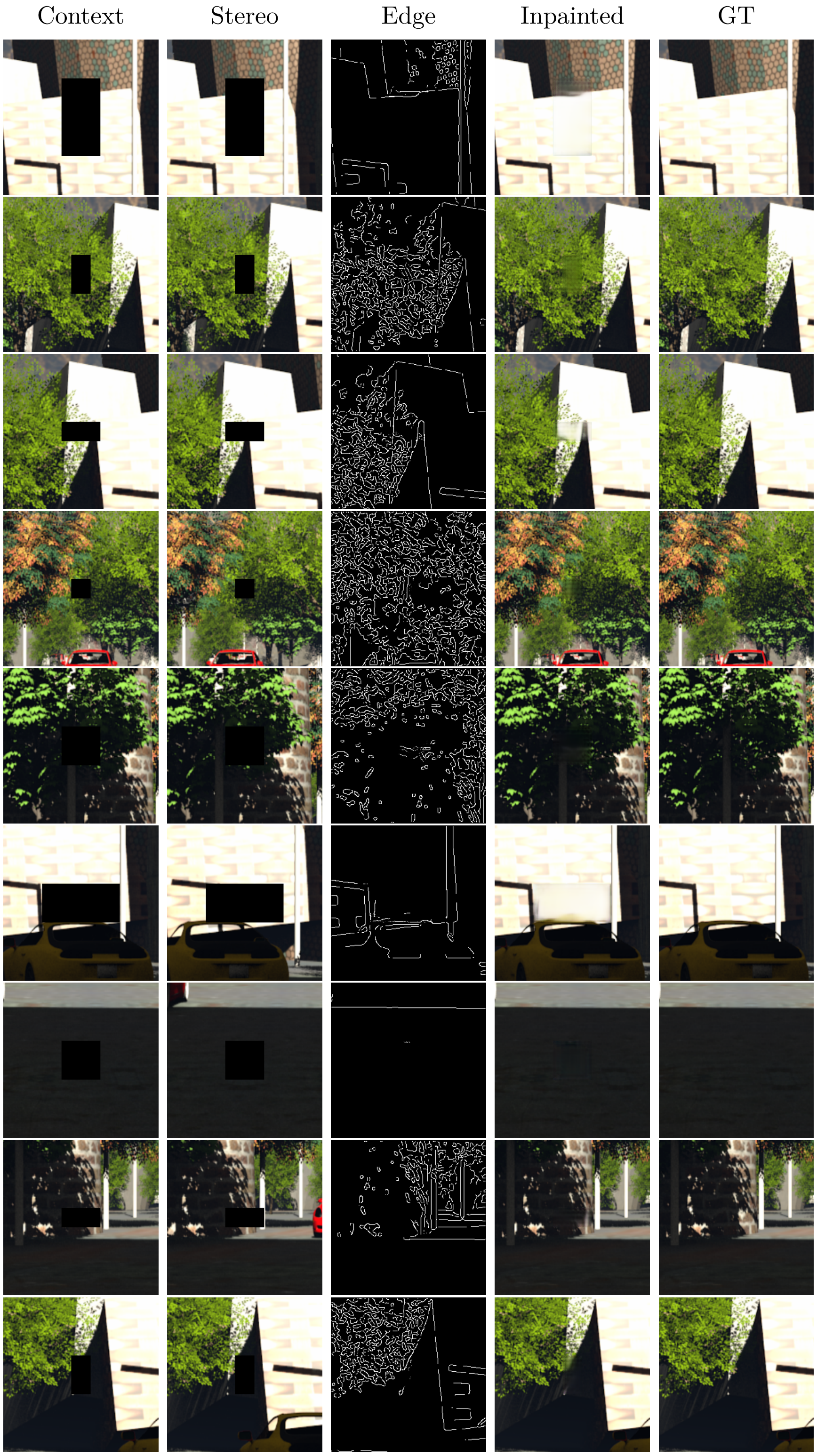


Figure 3.14: **Qualitative results. Context:** RGB crop fed into the network. **Stereo:** RGB crop fed into the network. **Edge:** results from EdgeConnect [109]. **Inpainted:** our inpainted results. **GT:** ground truth. **Dataset:** *Driving* [98].

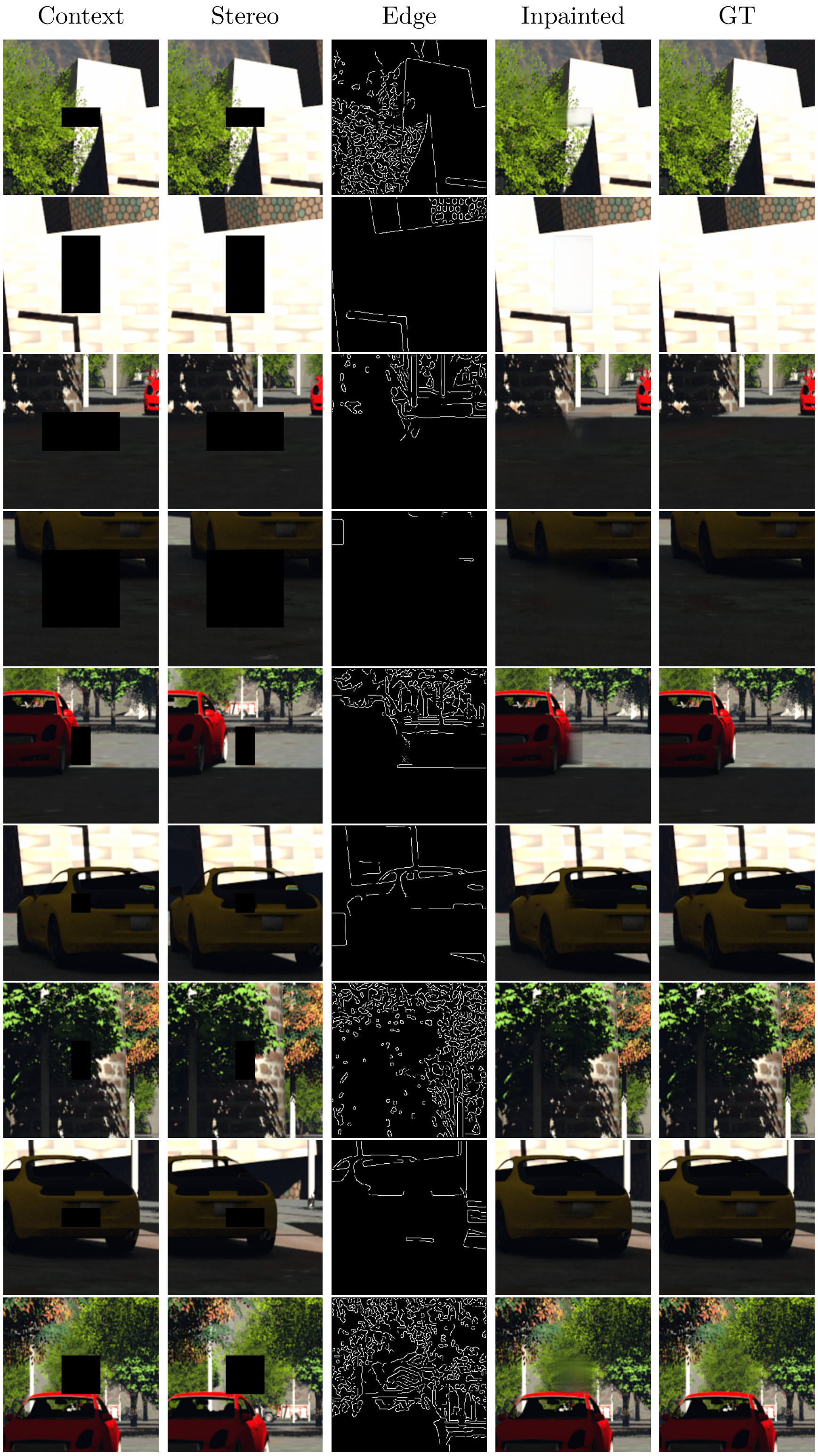


Figure 3.15: **Qualitative results. Context:** RGB crop fed into the network. **Stereo:** RGB crop fed into the network. **Edge:** results from EdgeConnect [109]. **Inpainted:** our inpainted results. **GT:** ground truth. **Dataset:** *Driving* [98].

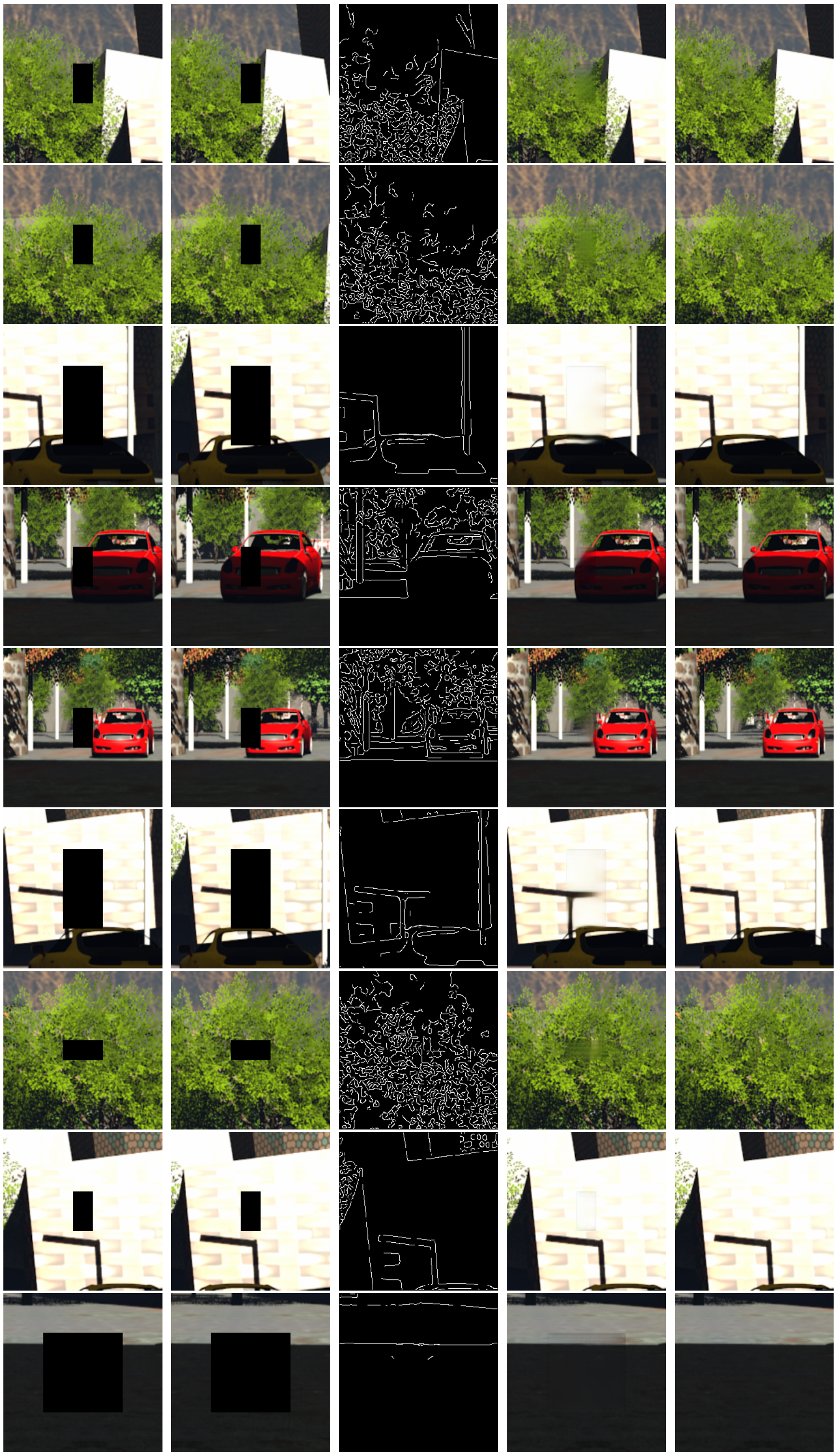


Figure 3.16: **Qualitative results. Context:** RGB crop fed into the network. **Stereo:** RGB crop fed into the network. **Edge:** results from EdgeConnect [109]. **Inpainted:** our inpainted results. **GT:** ground truth. **Dataset:** *Driving* [98].

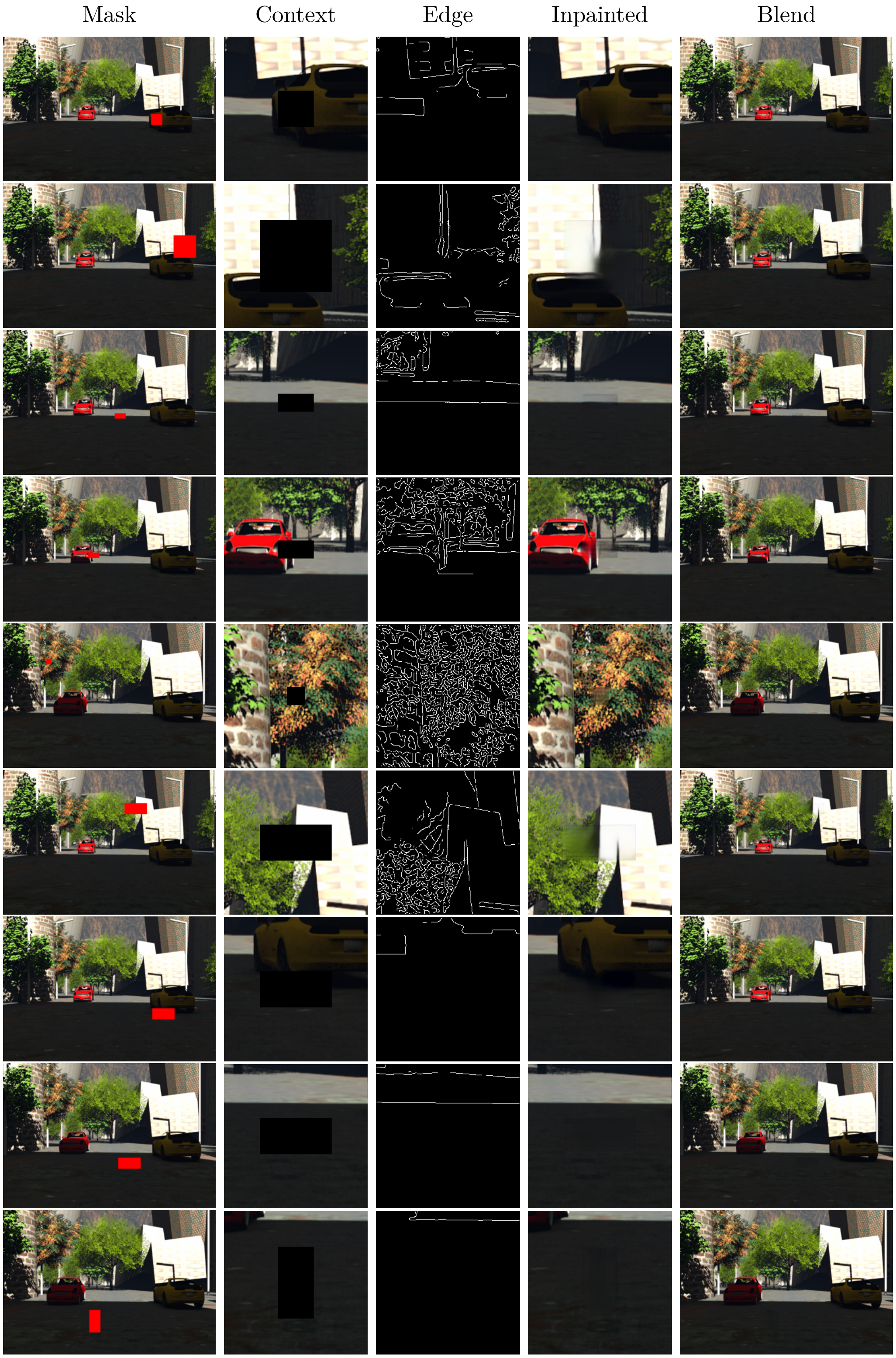
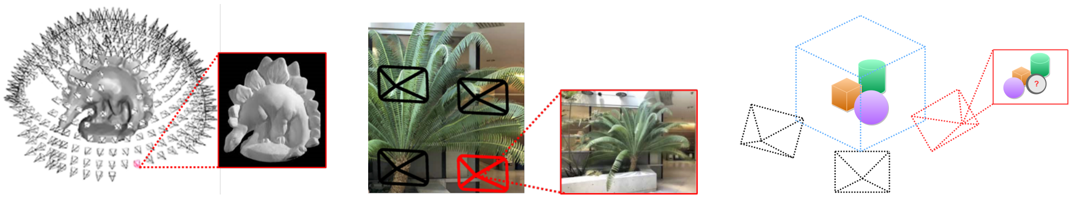


Figure 3.17: **Model performance. Mask:** Target image with masked synthesis area in red. **Context:** RGB crop fed into the network. **Edge:** Edge map fed into the network. **Inpainted:** our inpainted results. **Blend:** Target image with our inpainted region. **Dataset:** *Driving* [98].

Chapter 4

SVS: Adversarial Refinement for Sparse Novel View Synthesis

4.1 Problem Definition



(a) Dense View Synthesis (b) Few View Synthesis (c) Sparse View Synthesis

Figure 4.1: Different view synthesis operating modes, with varying numbers of views and varying baselines between views. In each case the black cameras are reference views and the red camera is the target view. Note that with few views and a wide baseline, occluded regions appear in the rendered scene which are visible in only 1 or none of the reference views.

This thesis focuses on the use of deep Novel View Synthesis (NVS) models applied to the production of media, in particular we are concerned with the capture of live events and low budget productions. In these cases, we are confronted with the practical constraints imposed by the available physical recording locations. Oftentimes, these locations deviate significantly from the ideal professional studios, offering suboptimal camera placements. These cameras may need to be located at awkward angles and camera planes, sitting far apart from each other, being only able to capture partial views of the scene. Consequently, this scenario introduces a unique view synthesis challenge characterised by a limited number of reference views and a substantial baseline between the target and reference views. We denote this as the Sparse View Synthesis (SVS) problem (Fig. 4.1). This results in a number of challenges which are not present in the standard Novel View Synthesis (NVS) problem setup. Most notably the fact that large portions of the target view may be occluded or otherwise not visible within the reference views.

In the previous chapter, we presented a model that performed stereo-consistent inpainting, where the missing part of the image was recovered consistently across views. This could be used in NVS systems where the underlying scene representation is based on Multi-Plane Images (MPIs) or Layered Depth Images (LDIs), to fill in the image holes left behind objects when the point of view is changed. The main limitation of these layered representations lies in the prescription of the underlying scene representation, and the discretisation of the scene which hinders the reproduction of sloped surfaces and reduces the quality of the renderings. Subsequent to the development of our previous work, there was a rapid growth in the use of implicit scene representations, which are far more expressive and flexible. In particular, the use of Neural Radiance Fields (NeRFs) [104] has become very popular due to its photo-realistic results. However, approaches based on NeRFs tend to be very expensive, requiring a multitude of input views and a very long per-scene optimisation process to obtain high-quality radiance fields. While this can be useful for tasks such a 3D object reconstruction for graphics design, it is far from practical and accessible for other applications such as live event capture.

The work in this chapter aims to make neural scene reconstruction more accessible and applicable to real world scene capture. In particular, we propose a method which does not require scene-specific model training, while still providing realistic results from a small sparse set of input views. The key challenge is effectively recognising and handling occluded areas, which were not observed from the small number of training views, while keeping rendering efficient. This necessitates a greater focus on generalisation and extrapolation and pure synthesis, as opposed to the data aggregation of traditional radiance field models.

To aid with the generalisation of radiance fields we try to replicate MVS techniques using DL approaches to reconstruct geometry priors. Similar models [20, 15] also attempt scene generalisation using MVS principles. However, these approaches have focused on narrow baseline extrapolation, where occlusions are limited, and thus do not solve the SVS problem.

To be able to deal with large occlusions and artefacts sensibly, we unify adversarial training [45] with radiance field models. Adversarial training was designed to help enrich the output variability of generative models, while dealing with artefacts in a realistic way. In the domain of NeRFs, this has the potential to ensure realistic extrapolation in unobserved regions. We have made our code publicly available[1].

In summary, the contributions of this chapter are:

- A radiance field approach capable of rendering novel views of a completely-unseen complex scene.
- An efficient MVS encoding-volume approach to scene-agnostic scene representation from a sparse set of views.
- A system that leverages adversarial networks to hallucinate large occlusion areas and regularise radiance fields.

4.2 Method

We propose a pipeline based on a Plane Sweep Volume (PSV) [36] neural encoding following MVSNeRF [15]. From this volume we sample random patches using radiance fields [104] which are supervised by an adversarial loss. As opposed to [104], we do not require a dense set of input images. We aim to learn a general model that can

[1] `https://github.com/violetamenendez/svs-sparse-novel-view`

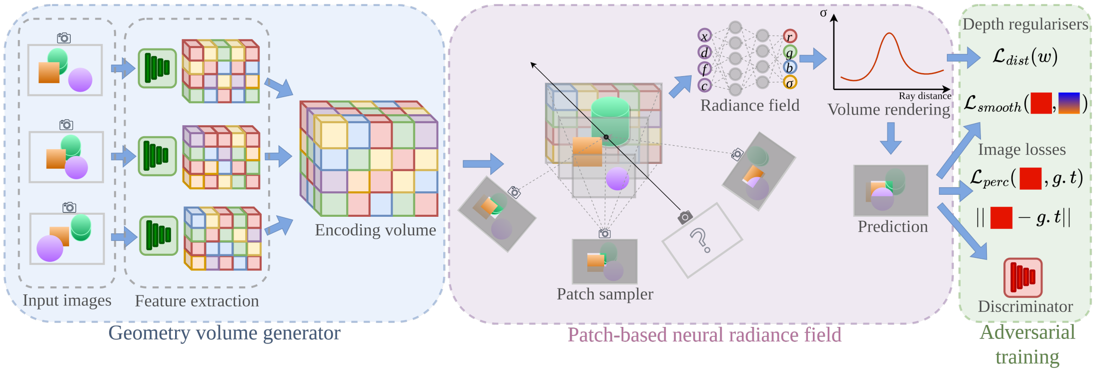


Figure 4.2: **Model overview.** The training pipeline consists of three main stages. Firstly, a feature extraction process in which a geometry volume is built to extract correlations across input views. Secondly, a patch-based neural radiance field stage, where an MLP predicts colour and density along rays. Finally, volume rendering is applied to reconstruct the predicted image patch, to which an adversarial loss is applied.

be applied to new unseen scenes without fine-tuning. Our model also aims to handle significant occlusions due to large baseline changes from sparse input viewpoints. In particular, we train a generalisable adversarial framework for radiance fields. An overall visualisation of our proposed model can be seen in Figure 4.2.

Given some sparse input images our model reconstructs an embedded neural volume which allows the model to reason about the implicit geometry of a scene. We use ray marching to sample from this volume and render a new point of view. We leverage adversarial training to help provide plausible rendering for large dis-occlusions and artefacts that arise from the large baseline changes. The following sections will detail each of these elements of our approach in turn.

4.2.1 Geometry volume generator

As the initial encoder for our generator, we use a 3D CNN encoding volume [182, 15] which integrates 2D CNN features of the input images. This allows the network to extract correlations between images, which can then be used to reason about geometry.

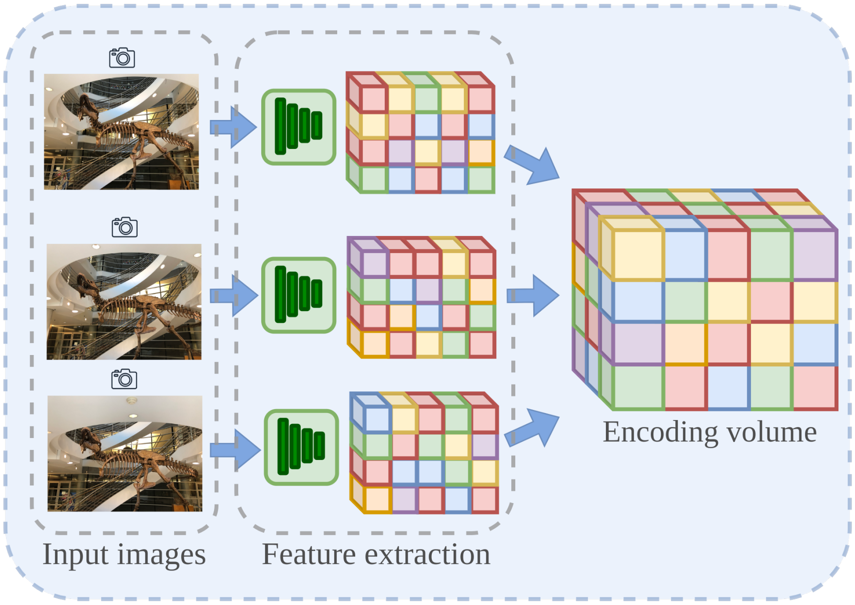


Figure 4.3: **Geometry volume generator.** To extract correlations across input images we extract 2D CNN features of these inputs and then combine them into an encoding volume using a variance-based cost and 3D CNNs.

The focus on image correlations as a mechanism for geometry extraction helps the network generalise to previously unseen scenes. The encoding volume is created at the reference view by warping multiple sweeping planes of source view features (see Figure 4.3). This is in contrast to techniques like Deep Stereo [36], which perform plane sweeps using the raw colour pixels to produce their correlation volume.

To construct this volume, we first extract the deep features of the N input images $\mathcal{N} = \{\mathbf{I}_i \mid \mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{N}$ using a deep 2D convolutional network $E(\mathbf{I}_i|\mathbf{w}_E)$ with weights $\mathbf{w}_E$. This network consists of downsampling convolutional layers, batch-normalization and ReLU activation layers (see Section 4.3 for further details on the implementation). For efficiency and generality, the feature encoding network is shared across all views [182].

Next we must align each feature map to the reference view at multiple depths to encode the plane sweep volume. To achieve this, a homography $\mathcal{H}_i\left(d\right)$ is computed for each view at each depth. Given the camera parameters $\{\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i\}$ (intrinsics, rotation and translation) for camera i the homography is defined as

$$\mathcal{H}_i\left(d\right) = \mathbf{K}_i \cdot \mathbf{R}_i \cdot \left(\mathbb{I} + \frac{\left(\mathbf{t}_{\text{ref}} - \mathbf{t}_i\right) \cdot \mathbf{n}_{\text{ref}}^T}{d}\right) \cdot \mathbf{R}_{\text{ref}}^T \cdot \mathbf{K}_{\text{ref}}^T \tag{4.1}$$

where $\mathbb{I}$ is the 3×3 identity matrix, $\mathbf{n}_{ref}$ the principle axis of the reference camera, and d is the depth which the images are being warped to. This operation is differentiable, which allows for end-to-end training of the feature encoding network weights $\mathbf{w}_E$ based on the downstream reconstruction losses.

Warping the feature maps according to this homography gives us the feature sweep volumes for each input image $\mathbf{I}_i \in \mathcal{N}$,

$$\mathbf{S}(\mathbf{I}_i) = \{W(\mathcal{H}_i\left(d\right), E(\mathbf{I}_i|\mathbf{w}_E)) \mid \forall d = 1, ..., D\}. \tag{4.2}$$

Then, a cost volume $\mathbf{C}(\mathcal{N})$ is created by aggregating all the warped feature sweep volumes, which encode appearance variations across views. To do this, a variance based cost metric is used, as it makes it possible to use an arbitrary number of input views,

$$\mathbf{C}(\mathcal{N}) = \text{Var}\left(\{\mathbf{S}(\mathbf{I}_i)|\forall \mathbf{I}_i \in \mathcal{N}\}\right) = \frac{\sum_{i=1}^{N}\left(\mathbf{S}(\mathbf{I}_i) - \overline{\mathbf{S}(\mathbf{I}_i)}\right)^2}{N}. \tag{4.3}$$

This cost volume is then processed using a 3D CNN UNet-like network [134]. This includes downsampling and upsampling layers with skip connections, to propagate scene appearance information. The output of this network is the neural embedding volume,

$$\mathbf{E} = V\left(\mathbf{C}(\mathcal{N})|\mathbf{w}_V\right) \tag{4.4}$$

This embedding volume represents the feature correlations from the point of view of the reference frame's Plane Sweep Volume. The structure of this volume is consistent across any arrangement of input viewpoints, and even any number of input views. This allows the system to generalise to new scene arrangements at inference time.

4.2.2 Volume rendering

We next use a neural radiance MLP with parameters $\mathbf{w}_\Theta$ to decode the *geometry embedding volume* $\mathbf{E}$ (Eq. 4.4) into volume density and view-dependent radiance (colour). Given a 3D point x, and a viewing direction d, we optimise a network F_Θ to regress the

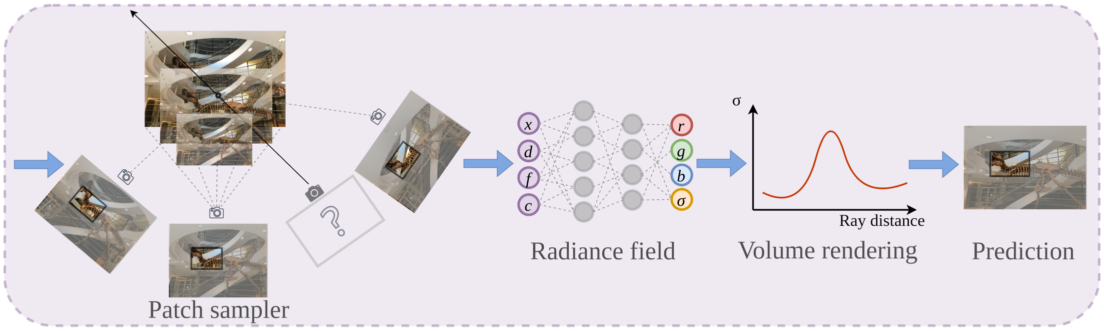


Figure 4.4: **Volume rendering pipeline**. We cast rays through image patches and sample along these rays. Then an MLP is used to predict colour and density values at those ray samples, which are then combined using volume rendering to return the final patch prediction.

density σ and colour r conditioned on the 3D feature $f = \mathbf{E}(x)$ corresponding to point x. To allow the correlations and structures in $\mathbf{E}$ to be mapped back to the original scene albedo, we use the pixel colour of the original image inputs as additional conditioning information. To do this, we concatenate the image pixel colours corresponding to point x into a colour vector $c = [\mathbf{I}_i(u_i, v_i)] \ \forall \mathbf{I}_i \in \mathcal{N}$, where (u_i, v_i) are the pixel coordinates corresponding to the reprojection of x onto the input view $\mathbf{I}_i$.

$$F_\Theta \colon \ (x, d, f, c|\mathbf{w}_\Theta) \mapsto (\sigma_{x,d}, r_{x,d}) \tag{4.5}$$

We use differentiable ray marching to regress the colour of reference image pixels. This is done by projecting ("marching") a ray through a pixel p in the reference image $\mathbf{I}_{\text{ref}}$. We can use the neural radiance network to obtain the radiance r_γ and density σ_γ at regular intervals $\gamma \in [1..\infty]$ along this ray via

$$(\sigma_\gamma, r_\gamma) = F_\Theta \left(\mathbf{t}_{\text{ref}} + \gamma \hat{d}, \hat{d}, f, c|\mathbf{w}_\Theta \right) \tag{4.6}$$

where $\hat{d} = \mathbf{R}_{\text{ref}}^T \cdot \mathbf{K}_{\text{ref}}^T p$. We can use these regular samples from F_Θ to obtain the predicted colour of the pixel $R(p)$ via the volume rendering Equation [69]:

$$R(p) = \sum_{\gamma} \tau_{\gamma} \left(1 - \exp\left(-\sigma_{\gamma}\right)\right) r_{\gamma} \tag{4.7}$$

$$\tau_{\gamma} = \exp\left(-\sum_{j=1}^{\gamma-1} \sigma_j\right) \tag{4.8}$$

where τ_{γ} is the transmittance at sample γ, which represents the probability that the ray travels up to γ without hitting another surface.

It is intuitive that the proposed approach will be able to predict density based on the consistency of feature representations between views. We can even see how analysing exactly which views correlate well for a given point can provide hints about occlusions, and guidance for albedo lookup. However, there is no simple mechanism to distinguish a region which has low correlation due to being empty, and one with low correlation due to being occluded in all views. The prevalence of these fully occluded regions grows drastically as the number of input views is reduced, and leads traditional radiance field models to produce reconstructions full of unrealistic holes.

4.2.3 Adversarial training

To combat this, we couple the above Generator network with a Discriminator network and undertake adversarial training. This makes it possible to enforce realism in unobserved regions. Usually in NeRF approaches, ray marching is performed through random pixels of the images during training. However, effective adversarial training requires spatial structure in the generated output, therefore we use a patch-based neural generator function based on Equation 4.7. The use of a patch-based generator serves two purposes. Firstly, it exponentially increases the number of possible training samples, ensuring that the discriminator is not able to memorise the training dataset. Secondly it greatly improves training efficiency as it can be expensive to repeatedly render entire images via the NeRF.

Following Schwarz *et al.* [141], we generate a variable patch that scales with training time. This allows for a variable receptive field. The patch $\hat{\mathbf{P}}_p$ of size $\delta \times \delta$ and centred on pixel p is defined as

$$\hat{\mathbf{P}}\left(p,s\right)=\left\{R\left(si+p_x,sj+p_y\right)\mid i,j\in\left\{-\frac{\delta}{2},...,\frac{\delta}{2}\right\}\right\} \tag{4.9}$$

where s is the scale that controls the active field of the patch. The scale exponentially decays during the training process, allowing our convolutional discriminator to learn independently of the image resolution.

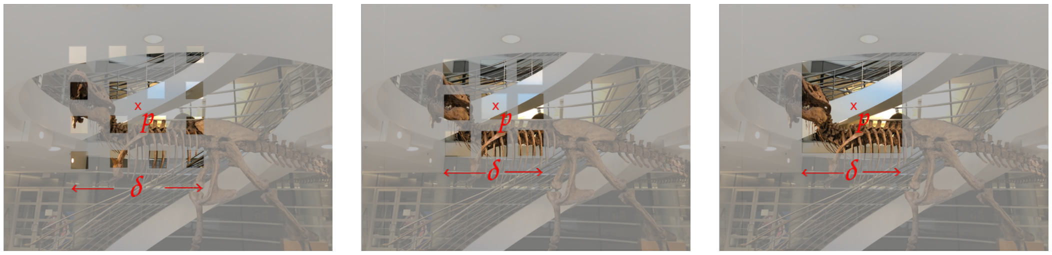


Figure 4.5: **Variable patches.** An illustration of the varying patch scale over the course of training, which allows the discriminator to learn independently of the image resolution.

Given the ground truth patch $\mathbf{P}_p$ for the target view $\mathbf{I}_{\text{gt}}$,

$$\mathbf{P}\left(p,s\right)=\left\{\mathbf{I}_{\text{gt}}\left(si+p_x,sj+p_y\right)\mid i,j\in\left\{-\frac{\delta}{2},...,\frac{\delta}{2}\right\}\right\} \tag{4.10}$$

we compute the L_1 loss between a randomly selected real and target patch

$$\mathcal{L}_{\text{rec}}=||\hat{\mathbf{P}}_p-\mathbf{P}_p|| \text{ where } p\sim\mathcal{U}([0,0],[W,H]). \tag{4.11}$$

This L_1 loss is effective at enforcing low-frequency correctness in the output. However, it can lead to overly-blurred results and difficulty recovering high-frequency structures. We therefore follow an LSGAN [95] approach and augment this with a convolutional PatchGAN [63] discriminator network D_Φ with parameters $\mathbf{w}_\Phi$.

The discriminator network takes patches as input, and is trained to classify input patches from the ground truth and from the generator as real or fake respectively. As such, the discriminator loss is defined as

$$\mathcal{L}_D=||1-D_\Phi\left(\mathbf{P}_p\right)||_2^2+||D_\Phi\left(\hat{\mathbf{P}}_p\right)||_2^2 \text{ where } p\sim\mathcal{U}([0,0],[W,H]). \tag{4.12}$$

Finally, we can use the PatchGAN discriminator's loss to also create an adversarial loss which constrains the generator network

$$\mathcal{L}_G = \lambda || 1 - D_\Phi \left(\hat{\mathbf{P}}_p \right) ||_2^2 \text{ where } p \sim \mathcal{U}([0,0],[W,H]). \tag{4.13}$$

where λ is a weighting factor. This adversarial loss encourages the generator to produce more realistic outputs which are able to fool the discriminator. Importantly, any holes in the reconstruction due to occlusions will provide obvious clues for the discriminator. Therefore the generator can only succeed in its task if the holes are filled with hallucinated photo-realistic content.

We should re-iterate that this entire pipeline is fully differentiable. This includes the discriminator, the patch-based volumetric rendering, the radiance field estimation, the feature correlation computation, the homographic plane-sweep warping and the input image feature lookup. As such, our adversarial loss $\mathcal{L}_G$ is able to constrain all the learnable parameters $\mathbf{w}_E, \mathbf{w}_V, \mathbf{w}_\Theta$ apart from those in the discriminator $\mathbf{w}_\Phi$ which are optimised based on $\mathcal{L}_D$. These two optimisations are performed using separate Adam optimisers [75] which are alternated.

4.2.4 Depth regularisation

Because SVS is an ill-defined problem, we found that the predicted depth images were extremely noisy, if not completely nonsensical. To be able to reconstruct a well-defined scene geometry, the predicted depth needs to be coherent with the image. We approach this issue by adding two depth regularisers.

Edge-aware depth smoothness

Firstly, we introduce a depth smoothness loss to encourage the network to generate continuous surfaces, similar to monocular depth estimation approaches [41]. As depth discontinuities usually happen at colour edges [53], the depth smoothness is weighted with the colour image gradients ∂I.

$$\mathcal{L}_{\text{smooth}}(d) = \frac{1}{N} \sum_{i,j} \left[|\partial_x d_{ij}| \, \mathrm{e}^{-||\partial_x I_{ij}||} + |\partial_y d_{ij}| \, \mathrm{e}^{-||\partial_y I_{ij}||} \right] \tag{4.14}$$

where $d_{i,j}$ is the predicted depth at pixel (i, j), and $I_{i,j}$ the respective colour value.

Distortion loss

In addition to smooth surfaces, we also want to get rid of other potential artefacts like "floaters" (small disconnected regions of occupied space that look fine from the input views, but would not be coherent if seen from another view), and "background collapse" (far surfaces modeled as semi-transparent clouds of dense content in the foreground). NeRF-based models [104] try to achieve this by adding Gaussian noise to the output σ values during optimisation. But this does not eliminate all geometry artefacts, and reduces the reconstruction quality. Instead, we follow Barron *et al.* [5] and include a distortion loss in our regularisation.

$$\mathcal{L}_{\text{dist}}\left(\mathbf{w}\right) = \sum_{i,j} \mathrm{w}_i \mathrm{w}_j + \frac{1}{3}\sum_{i} \mathrm{w}_i^2 \tag{4.15}$$

$$\mathrm{w}_i = \tau_i \left(1 - \exp\left(-\sigma_i\right)\right) \tag{4.16}$$

where w_i are the alpha compositing weights at ray sample i, derived from Equation 4.7. This regulariser minimises the weighted distances between all pairs of ray points, and the weighted size of each individual point. This helps the distribution function of the density along the rays approximates a delta function. Finally, we combine all the generator losses and regularisers with an LPIPS [190] perceptual loss. Our total loss is as follows:

$$\mathcal{L}_{\text{total}} = \frac{1}{2}\mathcal{L}_D + \mathcal{L}_G + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{perc}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}} \tag{4.17}$$

where we chose $\lambda_{\text{rec}} = 20, \lambda_{\text{smooth}} = 0.4, \lambda_{\text{dist}} = 0.001$ for our experiments.

4.3 Implementation details

We use COLMAP [140] to generate camera intrinsics and extrinsics at each input image. We extract deep image features from the selected input views using a 2D CNN network (Table 4.1), which are used to construct our cost volume. The neural embedding volume is defined by the 3D CNN architecture on the second section of Table 4.1.

For the NeRF MLPs, we follow a similar setup to the original case [104]. We sample 128 points along each ray, with a ray batch of 1024. The MLP networks return the

Table 4.1: **Detailed architecture**: f is the 3D volume features, c the reference colours, **k** the kernel size, **s** the stride, **d** the kernel dilation, and **chns** shows the number of input and output channels for each layer. We denote CBR2D/CBR3D/CTB3D/LR to be ConvBnReLU2D, ConvBnReLU3D, ConvTransposeBn3D and LinearRelu respectively. PE refers to the positional encoding as used in [104].

	Layer	**k**	**s**	**d**	**chns**	input
2D CNN	CBR2D$_0$	3	1	1	3/8	I
	CBR2D$_1$	3	1	1	8/8	CBR2D$_0$
	CBR2D$_2$	5	2	2	8/16	CBR2D$_1$
	CBR2D$_3$	3	1	1	16/16	CBR2D$_2$
	CBR2D$_4$	3	1	1	16/16	CBR2D$_3$
	CBR2D$_5$	5	2	2	16/32	CBR2D$_4$
	CBR2D$_6$	3	1	1	32/32	CBR2D$_5$
	F = CBR2D$_7$	3	1	1	32/32	CBR2D$_6$
3D CNN	CBR3D$_0$	3	1	1	$32 + 3 * 3/8$	F, I
	CBR3D$_1$	3	2	1	8/16	CBR3D$_0$
	CBR3D$_2$	3	1	1	16/16	CBR3D$_1$
	CBR3D$_3$	3	2	1	16/32	CBR3D$_2$
	CBR3D$_4$	3	1	1	32/32	CBR3D$_3$
	CBR3D$_5$	3	2	1	32/64	CBR3D$_4$
	CBR3D$_6$	3	1	1	64/64	CBR3D$_5$
	CTB3D$_0$	3	2	1	64/32	CBR3D$_6$
	CTB3D$_1$	3	2	1	32/16	CBR3D$_4$ + CTB3D$_0$
	CTB3D$_2$	3	2	1	16/8	CBR3D$_2$ + CTB3D$_1$
	f	-	-	-	-	CBR3D$_0$ + CTB3D$_2$
MLP	PE$_0$	-	-	-	3/63	x
	LR$_0$	-	-	-	8+3*3/256	f, c
	LR$_1$	-	-	-	63/256	PE
	LR$_{i+1}$	-	-	-	256/256	LR$_i$+LR$_0$
	σ	-	-	-	256/1	LR$_6$
	PE$_1$	-	-	-	3/27	d
	LR$_7$	-	-	-	27+256/256	PE$_1$,LR$_6$
	r	-	-	-	256/3	LR$_7$

estimated colour r and density σ. We use an Adam optimiser [75] with a learning rate of $5e-4$. We use Positional Encoding (PE) [104] for the 3D location and viewing direction before feeding them into the networks. For more detailed information about the architecture, refer to the Table 4.1.

4.4 Evaluation

4.4.1 Datasets

For training we are using two different commonly used datasets, DTU [65] and Forward-Facing (LLFF) data [103]. The DTU dataset consists of a variety of scenes and objects taken in a lab setup. We follow the same training approach in related papers [185, 15], and split the dataset into 88 training scenes and 16 testing scenes, using an image resolution of 512×640. The Forward-Facing dataset consists of handheld phone captures taken in a 2D grid. We split the dataset into 35 training sets and 8 for testing in the same scenes used for NeRF. Because we focus on the Sparse View Synthesis problem, models are trained on 3 input views per scene.

4.4.2 Baseline models

We compare our method against the current state-of-the-art neural rendering methods for few-input NVS. All methods are trained over LLFF and DTU using three input images. We evaluate IBRNet [170], MVSNeRF [15] and our method over long baseline movements. We were not able to train GeoNeRF [68] as the code had not been released at the time of the development of this work, and the results in their paper are for a much easier problem.

4.4.3 Results

From Table 4.2 we can see that our proposed approach performs similarly to RegNeRF in terms of accuracy. This is despite the fact that RegNeRF is trained in a scene specific regime, while our approach is trained on unrelated scenes, then applied to a completely unknown scene at test time.

Table 4.2: **Quantitative evaluation.** We evaluate our model over the DTU and Forward Facing datasets. **Bold** is best result, *italic* is second best.

Model	Experiment	DTU			Forward facing		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RegNeRF* [114]	Optimised	18.89	0.745	0.190	19.08	0.587	0.336
IBRNet [170]	Unseen	12.71	0.4772	0.5678	16.40	0.5230	0.4986
MVSNeRF [15]		*18.92*	*0.6831*	*0.2580*	**16.98**	**0.5839**	*0.3853*
Ours		**19.03**	**0.6929**	**0.2066**	*16.55*	*0.5534*	**0.3441**

When comparing our technique against the other scene-agnostic state-of-the-art approaches (IBRNet and MVSNeRF) under the SVS evaluation protocol, we note that the simplistic PSNR and SSIM accuracy measures are relatively similar. However, drastic improvements are seen in the LPIPS metric over previous work ranging from a 15% to 60% improvement in the perceptual accuracy of the reconstructed scene. It is interesting to note that our approach performs especially well on the more challenging DTU dataset. Qualitative examples for both datasets are shown in Figures 4.6 and 4.7. We can observe that our model manages to recover sharper edges and object boundaries where the state-of-the-art baseline experiences fuzziness, smearing, and duplication artefacts. Unfortunately, our approach still struggles to reconstruct very complex backgrounds and suffers from certain blurriness in some areas.

4.4.4 Ablation study

In Table 4.3 we undertake an ablation study to measure the performance enhancement of each contribution. The depth smoothing loss makes a small but noticeable difference across all metrics. It is interesting to note that the distortion loss leads to a marginal decrease in the PSNR and SSIM metrics on the Forward Facing dataset. However, it still provides a more significant improvement in terms of LPIPS. This is expected, as

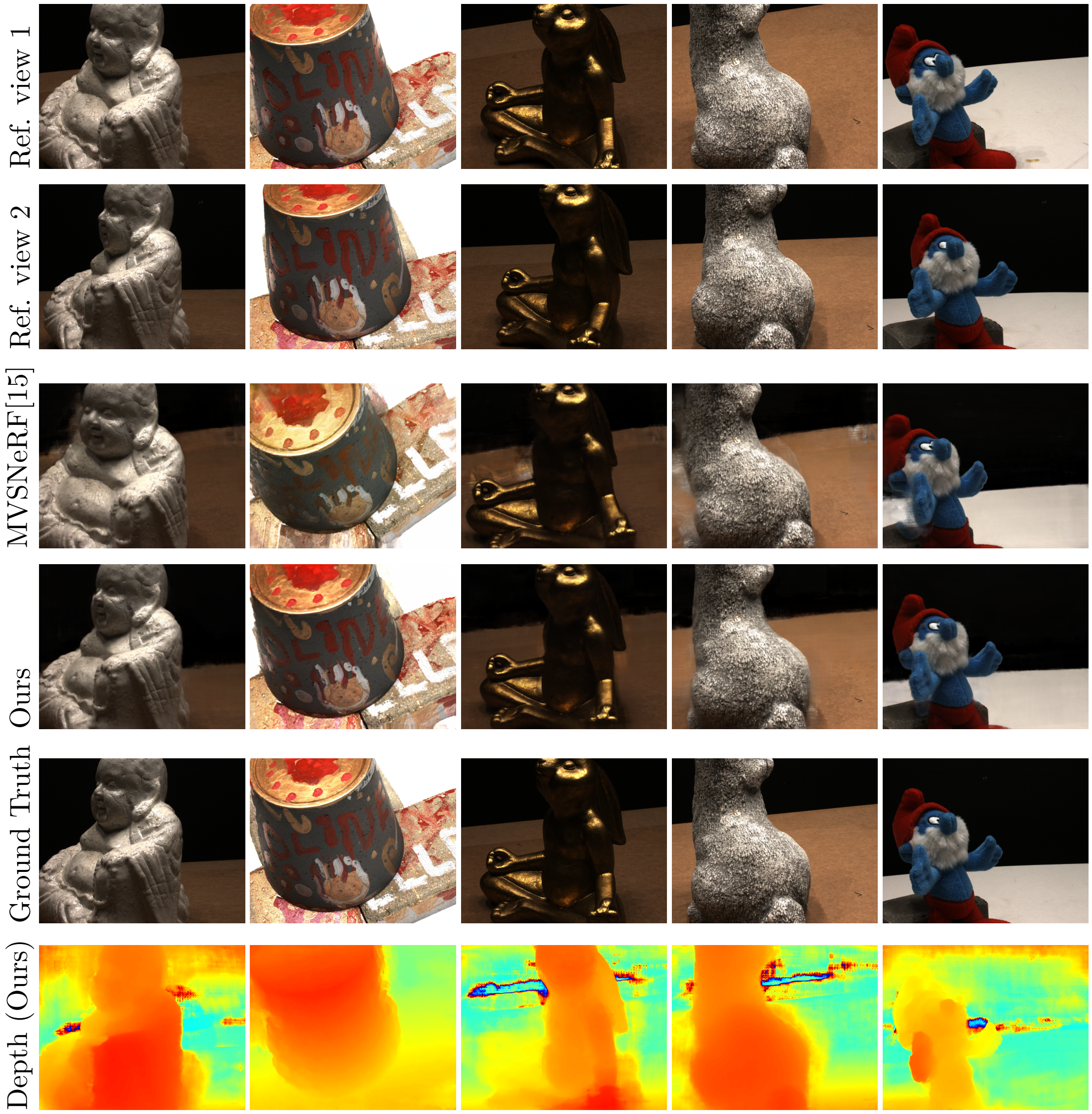


Figure 4.6: Example predictions from the DTU dataset. Each column shows a different scene predicted under the Sparse View Synthesis evaluation protocol. Note that state of the art [15] predictions exhibit fuzziness or “smearing” around object boundaries where occlusions occur. In contrast our approach provides sharper edges as it realistically fills unobserved regions.

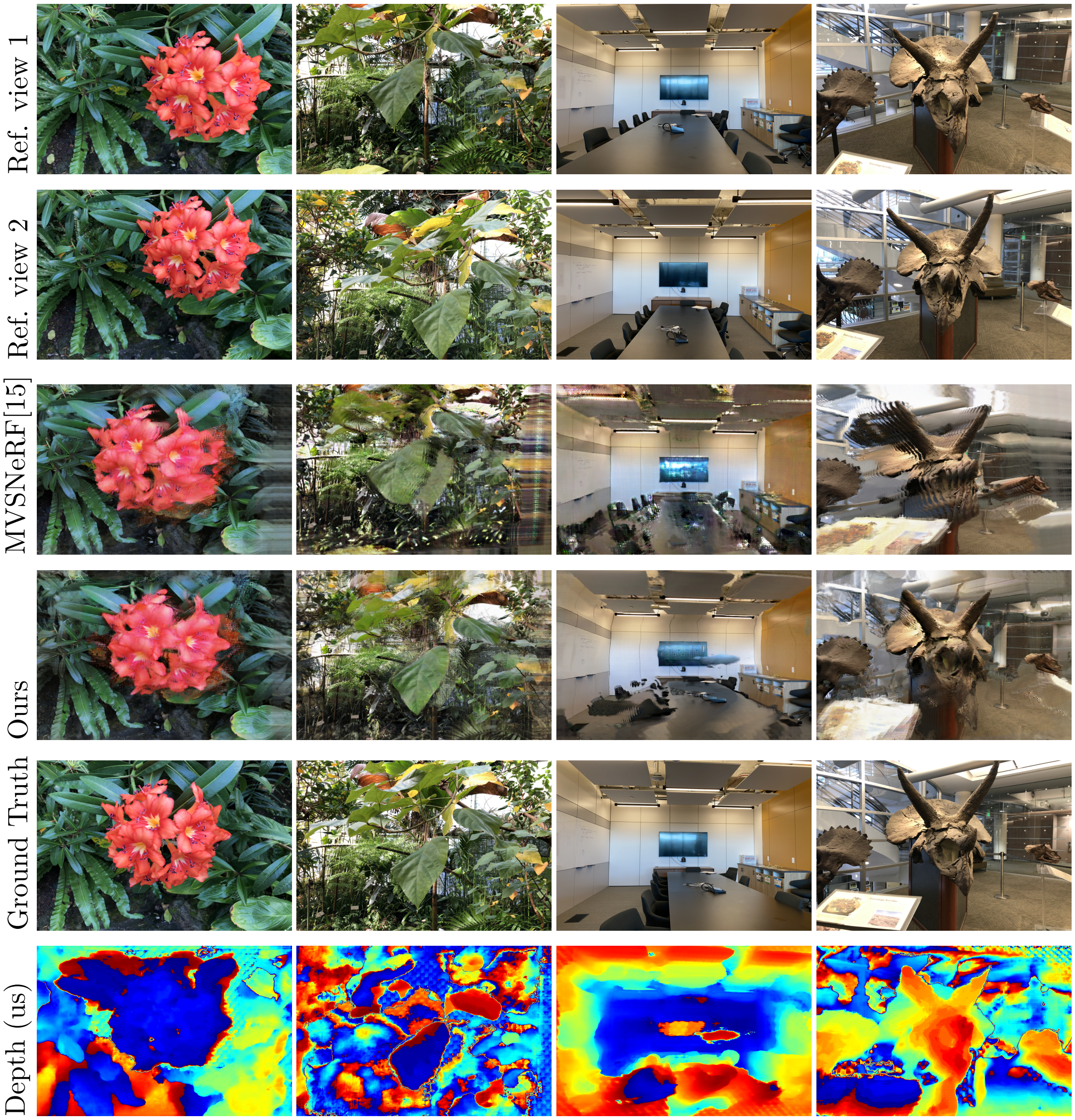


Figure 4.7: Example predictions from the LLFF dataset. Again we can observe "smearing" artefacts in the state-of-the-art predictions, especially along the image borders. Additionally, we can observe "halo" or "duplicated boundary" artefacts in some of the state-of-the-art scenes such as the rightmost two columns. These perceptually jarring artefacts are easily detected by our adversarial discriminator, and so they do not occur in the predictions for our approach.

the distortion loss slightly limits the flexibility of the volumetric rendering by preventing "smearing" the scene across depths. However, this in turn removes floating blob artefacts and blurred scene depth when viewed from distant viewpoints. These artefacts have a significant impact in the overall perceptual quality of the rendered image, and are vital for possible human-centric applications.

The final adversarial loss leads to significant improvements across all metrics, with the largest gains once again being with the LPIPS score. This demonstrates that the integration of adversarial learning is vital for producing plausible renders for SVS.

Table 4.3: **Ablation study.** We study the effect of each addition to the model on the Forward facing dataset. **Bold** is best result, *italic* is second best.

Depth Smooth	Distortion	Adversarial	DTU			Forward facing		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
×	×	×	18.65	0.6780	0.3959	16.11	0.5318	0.4791
✓	×	×	18.70	0.6799	0.3882	*16.14*	0.5361	0.4709
✓	✓	×	*18.71*	*0.6802*	*0.3530*	16.10	*0.5419*	*0.4611*
✓	✓	✓	**19.03**	**0.6929**	**0.2066**	**16.55**	**0.5534**	**0.3441**

4.5 Conclusion

Within the frame of generating virtual camera views for live event capture or low budget productions, we are faced by the physical constraints posed by the recording locations. These locations are often not ideal for the production of audio-visual media, and cameras are placed in non-optimal positions. This poses a view synthesis problem where the number of reference views is limited, and the baseline between target and reference view is significant. We call this the Sparse View Synthesis (SVS) problem. This results in a number of challenges which are not present in the standard Novel View Synthesis (NVS) problem setup. Most notably the fact that large portions of the target view may be occluded or otherwise not visible within the reference views.

While implicit representation models like NeRF have become very popular to tackle

NVS problems, these approaches face computational limitations, and require a large number of dense input views. In the case of SVS, there is no mechanism for a standard radiance field model to appropriately fill the large disocclusion gaps. Therefore, in this chapter we proposed an algorithm which unified generative adversarial learning techniques with traditional radiance field modelling. This encouraged the system to inpaint unobserved regions with plausible scene completions. This led to perceptual quality improvements of up to 60% compared to existing radiance field models.

Although the proposed approach is a major step towards achieving more extreme and sparse renderings, there is still some way to go to achieve full extreme SVS. While GANs produce good results at generating new content, they suffer from the classic training instability, which makes the model harder to train. In addition, the difficulty of the problem means the complexity of the solution increases. As we handle extreme baseline movements, this creates an ill-posed problem where sometimes the network does not differentiate between empty or occluded space. Thus, in areas viewed by only one of the source views, the reconstruction can lack fidelity. In future work it may be possible to resolve this by re-weighting the generative losses in different regions based on visibility.

Most importantly for the application of audio-visual content production, future techniques must be generalised to temporal video. Dynamic scenarios may consist of static or moving cameras, and may contain rigid and non-rigid dynamic objects. Adapting our current approach by naively adding a temporal dimension to our NeRF input would not be able to correctly capture complex dynamics and maintain temporal consistency across video frames. Thus, we require a model that can represent scenes from sparse inputs with large uncertainty, as well as being capable of modelling complex dynamics.

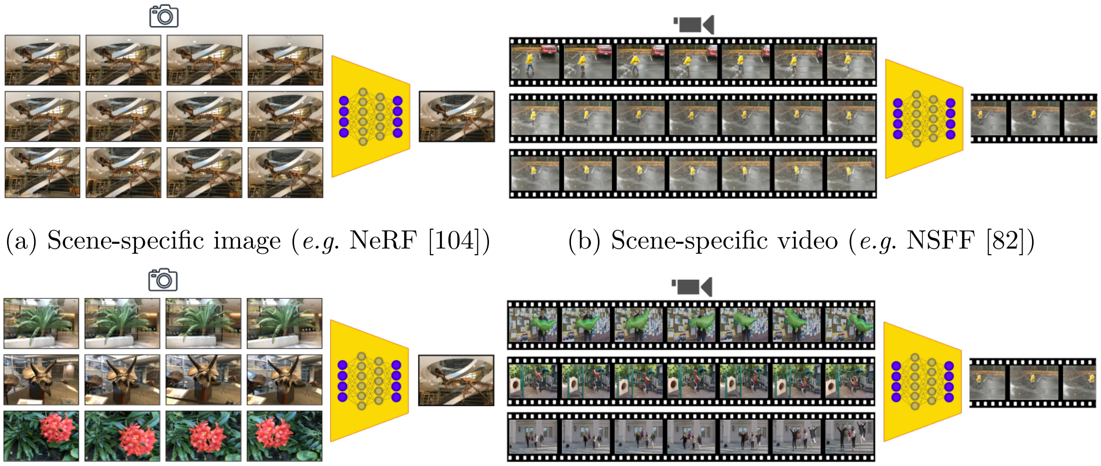
Chapter 5

ZeST-NeRF: Using Temporal Aggregation for Zero-Shot Temporal NeRFs

5.1 Problem Definition

As the motivation of this thesis lies within television production and live event capture, our final aim is to be able to render novel views from input video. This problem is particularly challenging, due to the need to model the temporal component of videos. In this chapter we present the next logical extension of our SVS image model to video rendering. Instead of using multiple camera views, we follow a similar approach to recover missing content of a scene by leveraging temporal information across keyframes. This allows us to render new points of view from a given monocular video.

As we have previously seen, many new approaches [114, 5] tackle the problem of image NVS by using NeRF [104] as an impicit representation of the scene. These models overfit a Multi-Layer Perceptron (MLP) to a given static scene (Fig. 5.1a). This has been extended to dynamic scene methods that aim to model the temporal dimension alongside the spatial ones [82, 81] (Fig. 5.1b). These approaches achieve impressive photo-realistic results but have similar limitations to NeRF, being very expensive, requiring a lot of



(a) Scene-specific image (*e.g.* NeRF [104])

(b) Scene-specific video (*e.g.* NSFF [82])

(c) "Zero-shot" image (*e.g.* MVSNeRF [15])

(d) "Zero-shot" video (ZeST-NeRF)

Figure 5.1: **"Zero-shot" temporal NeRF** contrasted with existing techniques

input views, and having a very long per-scene optimisation process. Some approaches, like PixelNeRF [185], MVSNeRF [15], and our previous work on SVS [43], sought to generalise NeRFs to unseen scenes from a limited number of input views (Fig. 5.1c). This is already a highly ill-defined problem, which suffers from uncertainty and ambiguity in the reconstructed data. Naively adding a layer of temporal complexity can significantly exacerbate this, preventing the network to model complex dynamics and to maintain temporal consistency. In this work, the temporal data becomes an advantage, as information is aggregated across frames to resolve ambiguities in unknown scenes.

In particular, we propose ZeST-NeRF (Fig. 5.1d), a model that uses temporal information to reconstruct a scene geometry and motion estimate. This can then be used to generalise to unseen scenes without requiring expensive retraining. Given a set of keyframes, we use simultaneously a static geometry encoding volume to inform a static NeRF (which reconstructs the background) and a dynamic encoding volume to inform a dynamic NeRF (which reconstructs the motion between frames). We then combine these to generate a new video from a new point of view. Because our approach learns to estimate structure and motion in a scene-agnostic way, it can be applied "zero-shot" to new scenes without laborious training. Note that in the context of NeRFs we use 'zero-shot' to refer to the ability to perform inference on scenes (and scene configurations) that differ from those seen during training. This is in slight contrast to the usage of the phrase in supervised learning which refers specifically to ‘categories’ of data which were unseen during training. We have made our code publicly available[1].

In summary, the contributions of this chapter are:

- First radiance field approach capable of “zero-shot” novel viewpoint rendering in completely unseen videos of complex scenes.
- An efficient multi-encoding-volume approach to scene-agnostic video representation.
- A visibility-based loss re-weighting for high-fidelity disocclusion reconstruction.
- A new evaluation protocol and newly developed baselines for the problem of “zero-shot” novel-viewpoint-rendering in the video.

5.2 Method

5.2.1 Geometry and motion encoding volumes

Our architecture (See Fig. 5.2) exploits the power of multi-view 3D CNN encoding volumes [15, 43] to generalise to new scenes. These volumes are constructed similarly to our previous *Geometry Volume Encoder* from Section 4.2.1. The 3D volumes integrate 2D CNN features of the input images by warping multiple sweeping planes of source view features. However, instead of using images from different camera views to construct our volumes, we use selected frames across the video. This allows us to generalise correlations between images, which can then be used to reason about geometry and motion, helping the network generalise to previously unseen scenes.

We use two different volumes which learn different correlations between frames by construction. The *geometry volume* and the *motion volume* described below are constructed with the same network architecture but have different inputs and do not share weights.

[1] `https://github.com/violetamenendez/zest-nerf`

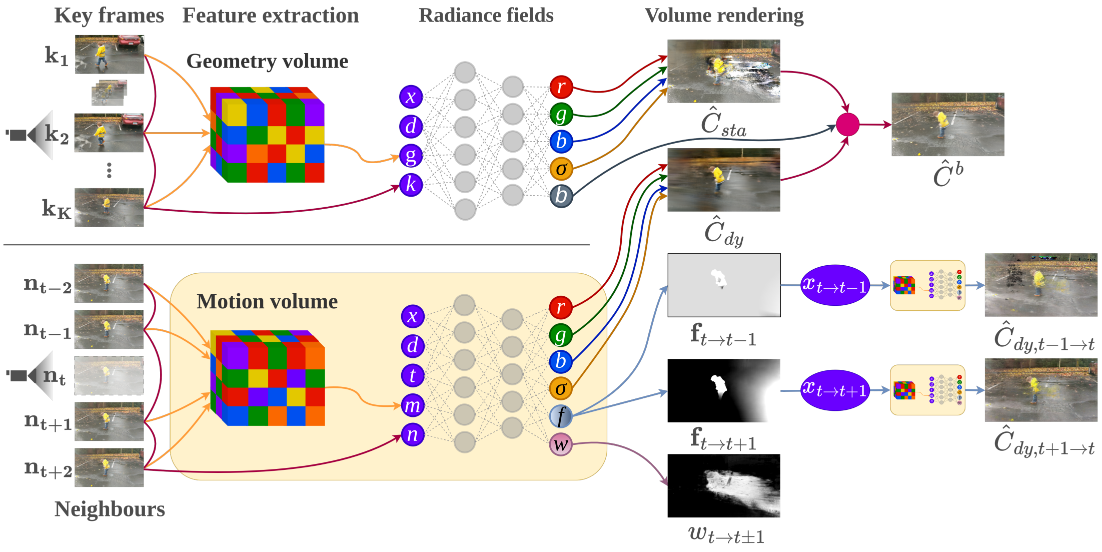


Figure 5.2: **Model overview:** ZeST-NeRF uses a geometry and a motion encoding volume to inform the static and the dynamic NeRFs, respectively. The static network captures the static elements of the scene, like the parts of the background that remain constant throughout the video. While the dynamic network represents the dynamic objects and motions of the scene, which can include static parts that are partially occluded by transient objects.

Geometry volume Given the image sequence $\mathcal{V} = \{\mathbf{I}_i \mid \mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{N}$, we use K keyframes that are representative of the whole video $\mathcal{K} = \{\mathbf{I}_i \mid \mathbf{I}_i \in \mathcal{V}, i = \lambda \cdot j, \lambda \in \mathbb{N}\}_{j=1}^{K}$. In our experiments, we use keyframes equally spread across the sequence. However, our approach could easily make use of intelligent keyframe selection techniques. These frames will inform the volume about the general geometry of the scene, particularly including static background elements. We then extract the deep features of each key frame using a deep 2D convolutional network $E(\mathbf{I}_i|\mathbf{w}_\Psi)$ with weights $\mathbf{w}_\Psi$. This network consists of downsampling convolutional layers, batch-normalisation and ReLU activation layers.

We build the plane sweep volume by aligning each feature map to the reference view at multiple depths. To achieve this, a homography $\mathcal{H}_i\left(d\right)$ is computed for each view at each depth. Given the camera parameters $\{\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i\}$ (intrinsics, rotation and translation) for camera i the homography is defined as

$$\mathcal{H}_i\left(d\right) = \mathbf{K}_i \cdot \mathbf{R}_i \cdot \left(\mathbb{I} + \frac{\left(\mathbf{t}_{\text{ref}} - \mathbf{t}_i\right) \cdot \mathbf{n}_{\text{ref}}^T}{d}\right) \cdot \mathbf{R}_{\text{ref}}^T \cdot \mathbf{K}_{\text{ref}}^T \tag{5.1}$$

where $\mathbb{I}$ is the 3×3 identity matrix, $\mathbf{n}_{\text{ref}}$ the principle axis of the reference camera, and d is the depth to which the images are being warped. This operation is differentiable, which allows for end-to-end training of the feature encoding network weights $\mathbf{w}_\Psi$ based on the downstream reconstruction losses.

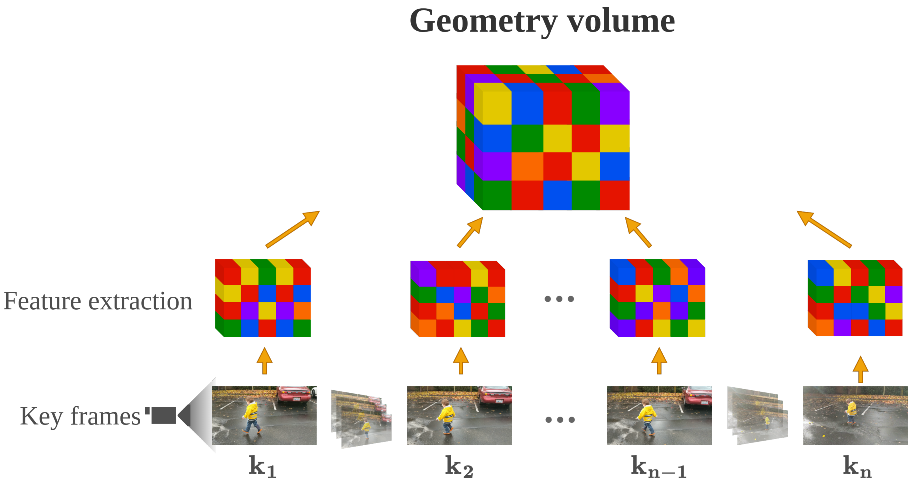


Figure 5.3: **Static encoding volume.** To extract static correlations across the video we select n keyframes and extract their 2D CNN features, which are then combined into a 3D CNN encoding volume through a variance-based cost.

The feature sweep volumes $S(\mathbf{I}_i)$ for each keyframe $\mathbf{I}_i \in \mathcal{K}$ are created by applying the warping function W determined by the homography $\mathcal{H}$ to the keyframe's feature maps at every depth,

$$S(\mathbf{I}_i) = \{W(\mathcal{H}_i\left(d\right), E(\mathbf{I}_i|\mathbf{w}_\Psi)) \mid \forall d \in \{1, ..., D\}\} \tag{5.2}$$

Then, a variance-based cost volume $\mathbf{C}(\mathcal{K})$ [19, 182] is generated by aggregating all the warped feature sweep volumes,

$$\mathbf{C}(\mathcal{K}) = \text{Var}\left(\{S(\mathbf{I}_i)|\forall \mathbf{I}_i \in \mathcal{K}\}\right) \tag{5.3}$$

This cost volume encodes appearance variations across views. A variance based metric makes it possible to compute this using an arbitrary number of input views.

Finally, the cost volume is processed using a 3D CNN UNet-like network [134]. The output of this network is the neural geometry embedding volume (Fig. 5.3),

$$\mathbf{G} = V(\mathbf{C}(\mathcal{K})|\mathbf{w}_{\Omega}) \tag{5.4}$$

This embedding volume encodes the feature correlations across key views in the video, which have been propagated using downsampling and upsampling layers with skip connections. This structure allows the volume to represent the static appearance elements from the video. In this way, the system can generalise to new scene arrangements and video lengths at inference time.

Motion volume This volume is similar in concept to the *geometry volume*, but it extracts dynamic correlations across short-term neighbouring frames instead of modelling static structure. By choosing only frames near our target time, the network is informed of the correlations caused by moving objects (Fig. 5.4).

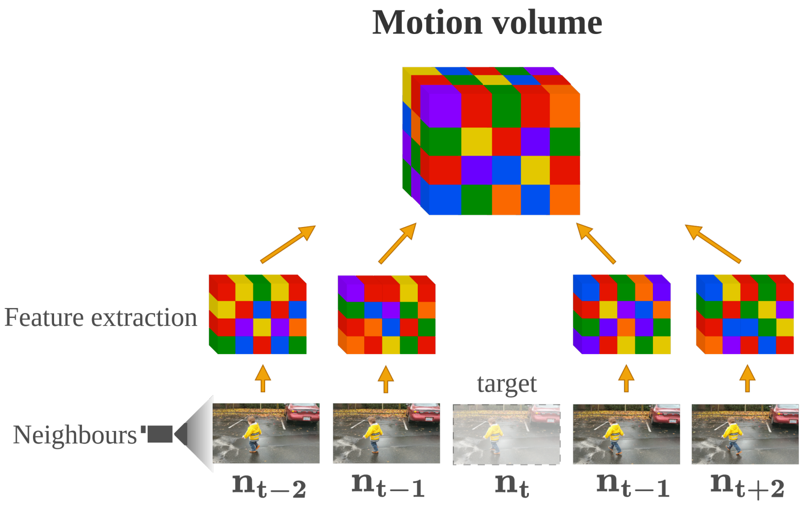


Figure 5.4: **Dynamic encoding volume.** To extract dynamic correlations we select temporal neighbouring frames to our target frame, we then extract 2D CNN features from them and combine these into a 3D CNN motion encoding volume using a variance-based cost.

We choose M neighbours around our target frame at time t to create this motion encoding volume. In practice, we use $M = 4$ for neighbours $\mathcal{N}_t = \{\mathbf{I}_i \mid \mathbf{I}_i \in \mathcal{V}, i \in \{t \pm 1, t \pm 2\}\}$.

We build the motion volume following the same approach as for the static geometry encoding volume. Using the neighbouring frames $\mathbf{I}_i \in \mathcal{N}_t$ and the homography in Equation 5.1 to build the feature sweep volumes $S(\mathbf{I}_i)$. Then aggregating the feature volumes using Equation 5.3 in the cost volume $\mathbf{C}(\mathcal{N}_t)$. And subsequently, passing the cost volume through a 3D CNN network with the same architecture as for the *geometry volume*, but with different training weights. We then obtain the motion volume,

$$\mathbf{M} = V(\mathbf{C}(\mathcal{N}_t)|\mathbf{w}_\Xi) \tag{5.5}$$

This volume thus represents the short-term behaviours of dynamic scene elements. Note that this includes the parts of the scene's background that are occluded by dynamic content, whose changing appearance across time makes the geometry network unable to reconstruct (see example in Figure 5.2).

5.2.2 Static neural radiance fields

We optimise an MLP [104] F_Θ with parameters $\mathbf{w}_\Theta$ to decode the *geometry volume* $\mathbf{G}$ (Eq. 5.4) embedding into a density and view-dependent radiance (colour). Given a 3D point x and a viewing direction d, the network F_Θ regresses the density σ and colour c at that point, conditioned on the corresponding 3D feature $g = \mathbf{G}(x)$. To allow the correlations and structures in $\mathbf{G}$ to be mapped back to the original scene albedo, we use the pixel colour of the original key frame inputs $\mathcal{K}$ as additional conditioning information [15]. These original colours are extracted from the input keyframes by reprojecting the 3D point x onto each keyframe and concatenating the pixel colour values. The colour vector is then $k = [\mathbf{I}_i(u_i, v_i)]\ \forall \mathbf{I}_i \in \mathcal{K}$, where (u_i, v_i) is the pixel location when reprojecting point x onto keyframe $\mathbf{I}_i$. We also predict blending weights b [82], which assign the linear weights to blend the colour and density estimated by the static and dynamic radiance fields (see Sec. 5.2.4). These weights are unsupervised and give higher importance to the static regions of the scene that this static representation can best model.

$$F_\Theta\colon\ (x, d, g, k|\mathbf{w}_\Theta) \mapsto (\sigma_x, c_{x,d}, b_{x,d}) \tag{5.6}$$

5.2.3 Dynamic neural radiance fields

To model the dynamics of our scene, we cannot just naively add a time dimension to our radiance field input. Doing so results in very noisy and inconsistent results due to the problem's dimensionality [82, 81]. Instead, we estimate scene flow fields [82] to aggregate information between frames. Our dynamic representation predicts the forward and backward 3D scene flow at a given location $\mathcal{F}_t = (\mathbf{f}_{t\rightarrow t+1}, \mathbf{f}_{t\rightarrow t-1})$. This denotes 3D offset vectors that point to the position of that point at times $t+1$ and $t-1$.

Unfortunately, density and scene flow are ambiguous in disocclusion regions caused by 3D motion. There is no simple mechanism to distinguish a region with a low correlation due to being empty and one with a low correlation due to being occluded in all frames. The prevalence of these fully occluded regions grows drastically as the number of input frames is reduced, leading traditional radiance field models to produce reconstructions full of unrealistic holes. To avoid this, we also predict disocclusion weights $\mathcal{W}_t = (w_{t\rightarrow t+1}, w_{t\rightarrow t-1})$. These unsupervised weights can be seen as the confidence of the estimated results and can guide the application strength of the reconstruction losses to the areas we are certain are observable across the video. The dynamic representation is then given by the following function F_Φ where x is the query point in 3D space, with viewing direction d at time $t \in \{1, ..., N\}$. $m = \mathbf{M}(x)$ is the corresponding feature from our *motion encoding volume* from Equation 5.5, to which we concatenate the original colour values from neighbouring frames $n = [\mathbf{I}_i(u_i, v_i)]\ \forall \mathbf{I}_i \in \mathcal{N}_t$,

$$F_\Phi\colon\ (x, d, t, m, n|\mathbf{w}_\Phi) \mapsto (\sigma_{x,t}, c_{x,d,t}, \mathcal{F}_t, \mathcal{W}_t)\,. \tag{5.7}$$

5.2.4 Volume rendering

We use differentiable ray marching to render the colour of image pixels. This is done by projecting ("marching") a ray $\mathbf{r}_t$ through a pixel in the target image $\mathbf{I}_t$ at time t. We query the respective neural radiance networks at regular intervals $\gamma \in [1..\infty]$ along this ray to obtain the radiance (colour) $c_\gamma = c\,(\gamma)$ and density $\sigma_\gamma = \sigma\,(\gamma)$ at each ray sample.

$$(\sigma_\gamma, c_\gamma, b_\gamma) = F_\Theta\left(\mathbf{r}_{t,\gamma}, \mathbf{d}, g, k|\mathbf{w}_\Theta\right) \tag{5.8}$$

$$(\sigma_{t,\gamma}, c_{t,\gamma}, \mathcal{F}_{t,\gamma}, \mathcal{W}_{t,\gamma}) = F_\Phi\left(\mathbf{r}_{t,\gamma}, \mathbf{d}, t, m, n|\mathbf{w}_\Phi\right) \tag{5.9}$$

where $\mathbf{r}_{t,\gamma} = \mathbf{r}_t(\gamma) = \mathbf{o}_t + \gamma\mathbf{d}$ with $\mathbf{o}_t$ being the ray origin and $\mathbf{d}$ the direction vector. Then, we use the predicted blending weights b_γ to "blend" the colour and density samples from F_Θ and F_Φ. Obtaining the estimated colour $\hat{C}^b(\mathbf{r}_t)$ of the pixel via the volume rendering equation [69]:

$$\hat{C}^b_{\text{sta}}(\mathbf{r}_t) = \sum_\gamma \tau^b_\gamma \left(1 - \exp\left(-\sigma_\gamma\right)\right) c_\gamma, \qquad \hat{C}^b_{\text{dy}}(\mathbf{r}_t) = \sum_\gamma \tau^b_\gamma \left(1 - \exp\left(-\sigma_{t,\gamma}\right)\right) c_{t,\gamma} \tag{5.10}$$

$$\hat{C}^b(\mathbf{r}_t) = \left(1 - b_\gamma\right) \hat{C}^b_{\text{sta}}(\mathbf{r}_t) + b_\gamma \hat{C}^b_{\text{dy}}(\mathbf{r}_t) \tag{5.11}$$

where τ^b_γ is the blended transmittance at sample γ, which represents the probability that the ray travels up to γ without hitting another particle.

$$\tau^b_\gamma = \exp\left(-\sum_{j=1}^{\gamma-1} \left(\sigma_j\left(1 - b_j\right) + \sigma_{t,j} b_j\right)\right) \tag{5.12}$$

5.2.5 Losses and regularisations

Reconstruction loss To enforce low-frequency correctness in the output, we compute the L_2 loss between the blended colour estimate and the true colour,

$$\mathcal{L}_{\text{rec}} = ||\hat{C}^b(\mathbf{r}_t) - C(\mathbf{r}_t)||^2_2 \tag{5.13}$$

Temporal photometric consistency To encourage the reconstruction of the scene at time t to be consistent with the scene at neighbouring times $k \in \mathcal{N}(t)$. We apply a photometric loss [82] to our dynamic network, which considers the motion due to 3D scene flow. This is done by warping the scene from neighbouring frame k to time t and comparing it to the ground truth scene at time t. Because this loss suffers from ambiguity in disocclusion areas, it is scaled by the estimated confidence weights mentioned in Sec. 5.2.3.

$$\mathcal{L}_{\text{pho}} = \sum_{k \in \mathcal{N}(t)} \hat{W}_{k \to t}(\mathbf{r}_t) ||\hat{C}_{\text{dy}}(\mathbf{r}_t + \mathbf{f}_{t \to k}) - C(\mathbf{r}_t)||^2_2 \tag{5.14}$$

where $\mathbf{f}_{t\to k} \in \mathcal{F}_t$ is the scene flow field from time t to k. We apply volume rendering to get the colour $\hat{C}_{\text{dy}}(\mathbf{r}_t + \mathbf{f}_{t\to k})$ from the dynamic network F_Φ at time k. We also apply volume rendering to get the disocclusion weights $\hat{W}_{k\to t}(\mathbf{r}_t)$, which are the accumulated confidence estimates $w_{t\to k,\gamma}$ along each ray.

$$\hat{C}_{\text{dy}}(\mathbf{r}_t + \mathbf{f}_{t\to k}) = \sum_{\gamma} \tau_{k,\gamma} \left(1 - \exp\left(-\sigma_{k,\gamma}\right)\right) c_{k,\gamma} \tag{5.15}$$

$$\hat{W}_{k\to t}(\mathbf{r}_t) = \sum_{\gamma} \tau_{k,\gamma} \left(1 - \exp\left(-\sigma_{k,\gamma}\right)\right) w_{t\to k,\gamma} \tag{5.16}$$

$$\tau_{k,\gamma} = \exp\left(-\sum_{j=1}^{\gamma-1} \left(\sigma_{k,j}\right)\right) \tag{5.17}$$

Disocclusion weight regularisation The system can fall into a degenerate local minimum, where the disocclusion weights $w_{t\to k}$ are zero at every 3D point x_t, and only the static NeRF is active. We avoid this with an L_1 regularisation that pushes them closer to one.

$$\mathcal{L}_w = \sum_{x_t} \sum_{k\in\mathcal{N}(t)} ||w_{t\to k}(x_t) - 1|| \tag{5.18}$$

Blending weights entropy loss This loss encourages blending weight to be either 0 or 1, which can help to reduce the ghosting caused by learned semi-transparent blending weights.

$$\mathcal{L}_b = || - b \cdot \log\left(b\right)|| \tag{5.19}$$

Cycle loss We enforce a cyclic regularisation to ensure that the predicted forward flow field at time t, $\mathbf{f}_{t\to k}$, is consistent with the backward flow field at time k, $\mathbf{f}_{k\to t}$, where k is a neighbouring time $k \in \{t \pm 1\}$. Fundamentally, if $x_{t\to k} = x_t + \mathbf{f}_{t\to k}$ then we should see that $x_{t\to k} + \mathbf{f}_{k\to t} = x_t$. That is, $\mathbf{f}_{t\to k} = -\mathbf{f}_{k\to t}$. Flow fields are ambiguous at disocclusion areas, so we regulate this loss with the predicted disocclusion weights [82].

$$\mathcal{L}_{\text{cyc}} = \sum_{x_t} \sum_{k\in\{t\pm1\}} w_{t\to k}||\mathbf{f}_{t\to k}(x_t) + \mathbf{f}_{k\to t}(x_{t\to k})|| \tag{5.20}$$

Scene flow regularisation We assume that motion is small in most 3D space [164], so we minimise the absolute value of the flow fields.

$$\mathcal{L}_{\text{min}} = \sum_{x_t} \sum_{k \in \{t \pm 1\}} ||\mathbf{f}_{t \to k}(x_t)|| \tag{5.21}$$

Scene flow spatial smoothness We assume that the scene deforms in a piece-wise smooth way [110], meaning the scene flow field is spatially smooth.

$$\mathcal{L}_{\text{sp}} = \sum_{x_t} \sum_{y_t \in \mathcal{N}(x_t)} \sum_{k \in \{t \pm 1\}} \mathrm{w}^{\text{dist}} \left(x_t, y_t\right) ||\mathbf{f}_{t \to k}(x_t) - \mathbf{f}_{t \to k}(y_t)|| \tag{5.22}$$

where $\mathcal{N}\left(x_t\right)$ are the neighbouring points of x_t, and $\mathrm{w}^{\text{dist}}\left(x_t, y_t\right) = \exp(-2||x_t - y_t||)$ are weightings based on Euclidean distance.

Scene flow temporal smoothness Finally, we add a scene flow temporal smoothness loss. This loss encourages 3D point trajectories to have minimal kinetic energy [168] i.e. constant velocity and piece-wise linear motion.

$$\mathcal{L}_{\text{temp}} = \sum_{x_t} ||\mathbf{f}_{t \to t+1}(x_t) + \mathbf{f}_{t \to t-1}(x_t)||_2^2 \tag{5.23}$$

5.2.6 Weakly-supervised pre-training

Since the problem of reconstructing complex dynamic scenes is extremely ill-posed, the losses can converge to local minima when randomly initialised [82]. We first complete a data-mining stage to initialise the problem optimally. We use monocular optical flow and depth estimation networks to generate pseudo ground truth data to guide our network. However, as both models are not completely accurate, we use them for initialisation only and decay their contribution to zero during training.

Geometric consistency First, we compute a reprojection error between the scene flow field estimated by ZeST-NeRF and that derived from pre-trained optical flow models [159]. Although losses that rely on backprojection do not consider transparencies, this loss is only used for initialisation, and it can be compensated by other losses during training given that NeRFs are capable of representing transparencies. Given a ray $\mathbf{r}_t$ at time t and estimated 3D scene-flow $\mathbf{f}_{t\to k}$, we can calculate the expected location of the 3D point $\hat{X}_t(\mathbf{r}_t)$ and the expected 3D scene flow $\hat{F}_{t\to k}(\mathbf{r}_t)$ of that point, by using volume rendering,

$$\hat{X}_t(\mathbf{r}_t) = \sum_{\gamma} \tau_{t,\gamma} \left(1 - \exp\left(-\sigma_{t,\gamma}\right)\right) x_{t,\gamma} \tag{5.24}$$

$$\hat{F}_{t\to k}(\mathbf{r}_t) = \sum_{\gamma} \tau_{t,\gamma} \left(1 - \exp\left(-\sigma_{t,\gamma}\right)\right) \mathbf{f}_{t\to k,\gamma} \tag{5.25}$$

Then we project that displaced 3D point $\hat{X}_t(\mathbf{r}_t) + \hat{F}_{t\to k}(\mathbf{r}_t)$ into the camera view of the neighbouring frame (k) as

$$\hat{p}_{t\to k} = \pi\left(\mathbf{K}\left(\mathbf{R}_k\left(\hat{X}_t(\mathbf{r}_t) + \hat{F}_{t\to k}(\mathbf{r}_t)\right) + \mathbf{t}_k\right)\right). \tag{5.26}$$

We compare our estimation to the displaced pixel $p_{t\to k} = p_t + \mathbf{u}_{t\to k}$, where $\mathbf{u}_{t\to k}$ is the 2D optical flow derived using a pre-trained optical flow model [159]. We can then minimise the reprojection error with the following loss [61],

$$\mathcal{L}_{\text{geo}} = \sum_{k\in\{t\pm 1\}} ||\hat{p}_{t\to k} - p_{t\to k}|| \tag{5.27}$$

Single-view depth prior In traditional Neural Radiance Fields approaches, depth maps are extremely noisy, if not nonsensical. We use a pre-trained monocular depth estimation network [127] to encourage the expected termination depth along each ray $\hat{D}(\mathbf{r}_t)$ to be close to the derived depth $D(\mathbf{r}_t)$. We apply an L_1 loss using a robust scale-shift invariant metric [82],

$$\mathcal{L}_{\text{depth}} = ||\hat{D}(\mathbf{r}_t) - D(\mathbf{r}_t)||. \tag{5.28}$$

5.3 Implementation details

We use COLMAP [140] to generate camera intrinsics and extrinsics at each frame while masking features from regions associated with dynamic objects [82] using off-the-shelf instance segmentation [50]. We use the same architecture to build both the *geometry* and *motion* volumes, only differing in the number of input channels ($K = 8$ key-frames and $N = 4$ neighbours, respectively). The volumes are optimised separately and thus do not share weights. We extract 32 feature channels from the input images using a 2D CNN consisting of downsampling convolutional layers with Batch-Normalisation (BN) and ReLU activation function (first section of table 5.1). Warping these features we create a Plane Sweep Volume per input image, selecting $D = 128$ depth planes. These are aggregated through a variance-based cost volume and processed using a 3D UNet-like network, consisting of downsampling and upsampling convolutional layers with BN and ReLU, and skip connections (second section of table 5.1).

For the NeRF MLPs, we follow a similar setup to the original case [104]. We sample 128 points along each ray, with a ray batch of 1024. We also have two separate networks for the static and dynamic parts, which do not share weights. We append the normalised time indices as in NSFF [82] to our dynamic network inputs. The MLP networks return the estimated colour c and density σ, as well as blending weights b in the case of the Static MLP, and 3D scene flow f and occlusion weights w in the case of the Dynamic MLP. We use an Adam optimiser [75] with a learning rate of $5e-4$. We use Positional Encoding (PE) [104] for the 3D location and viewing direction before feeding them into the networks. For more detailed information about the architecture, refer to the table 5.2.

5.4 Evaluation

5.4.1 Datasets

We train on the Dynamic Scenes dataset [184]. This dataset consists of 8 short-time video clips recorded with 12 synchronised cameras, and sampled from each camera at different times to simulate a moving monocular camera. The scenes are collected in the wild and feature complex motions such as jumping, running, or dancing. As this dataset is quite small, we perform a leave-one-out cross-validation study to prove our performance. We train one full model per subset of scenes, created by holding out one single scene for validation. This means that we can test on every scene of the Dynamic Scenes dataset with a model that has been trained on completely unrelated scenes.

Table 5.1: **Encoding volumes architecture**: g/m denote the geometry and motion 3D features respectively. **k** is the kernel size, **s** is the stride, **d** is the kernel dilation, and **chns** shows the number of input and output channels for each layer. We denote CBR2D/CBR3D/CTB3D to be ConvBnReLU2D, ConvBnReLU3D, and ConvTransposeBn3D layer structure respectively.

	Layer	**k**	**s**	**d**	**chns**	input
2D CNN	CBR2D_0	3	1	1	3/8	I
	CBR2D_1	3	1	1	8/8	CBR2D_0
	CBR2D_2	5	2	2	8/16	CBR2D_1
	CBR2D_3	3	1	1	16/16	CBR2D_2
	CBR2D_4	3	1	1	16/16	CBR2D_3
	CBR2D_5	5	2	2	16/32	CBR2D_4
	CBR2D_6	3	1	1	32/32	CBR2D_5
	$F = \text{CBR2D}_7$	3	1	1	32/32	CBR2D_6
3D CNN	CBR3D_0	3	1	1	$32 + (K/N) * 3/8$	F, I
	CBR3D_1	3	2	1	8/16	CBR3D_0
	CBR3D_2	3	1	1	16/16	CBR3D_1
	CBR3D_3	3	2	1	16/32	CBR3D_2
	CBR3D_4	3	1	1	32/32	CBR3D_3
	CBR3D_5	3	2	1	32/64	CBR3D_4
	CBR3D_6	3	1	1	64/64	CBR3D_5
	CTB3D_0	3	2	1	64/32	CBR3D_6
	CTB3D_1	3	2	1	32/16	$\text{CBR3D}_4 + \text{CTB3D}_0$
	CTB3D_2	3	2	1	16/8	$\text{CBR3D}_2 + \text{CTB3D}_1$
	g/m	-	-	-	-	$\text{CBR3D}_0 + \text{CTB3D}_2$

Table 5.2: **MLPs architecture**: g/m denote the geometry and motion 3D features respectively. k and n are the original colours of the K key-frames and N neighbouring frames, that are concatenated to the inputs. **chns** shows the number of input and output channels for each layer. We denote LR to be LinearReLU layer structure. PE refers to the positional encoding as used in [104].

	Layer	**chns**	input
Static MLP	PE_0	3/63	x
	LR_0	8+K*3/256	g, k
	LR_1	63/256	PE
	LR_{i+1}	256/256	LR_i+LR_0
	σ	256/1	LR_6
	b	256/1	LR_6
	PE_1	3/27	d
	LR_7	27+256/256	PE_1,LR_6
	c	256/3	LR_7
Temporal MLP	PE_0	4/63	x, t
	LR_0	8+N*3/256	m, n
	LR_1	63/256	PE
	LR_{i+1}	256/256	LR_i+LR_0
	σ	256/1	LR_6
	f	256/6	LR_6
	w	256/2	LR_6
	PE_1	3/27	d
	LR_7	27+256/256	PE_1,LR_6
	c	256/3	LR_7

5.4.2 Baseline models

We compare our results to a space-time view synthesis approach NSFF [82] and two "zero-shot" static view synthesis approaches SVS [43] and MVSNeRF [15] applied naively across frames. We attempted to reproduce the results in Li *et al.* [81] but found it impossible with the information given in the paper due to the lack of published code.

5.4.3 Results

Table 5.3: **Quantitative evaluation.** Results on cross-validation training and further fine-tuning. **Bold** is best result, *italic* is second best.

Model	Scene-agnostic			Scene-specific		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NSFF [82]	13.15	0.3178	0.7334	**28.19**	**0.9280**	**0.0450**
SVS [43]	17.91	0.4958	0.3698	21.28	0.7103	0.2001
MVSNeRF [15]	*19.37*	*0.6198*	*0.2885*	*26.94*	*0.8678*	0.1208
ZeST-NeRF (Ours)	**21.59**	**0.6239**	**0.2048**	*26.94*	0.8575	*0.0995*

When comparing ZeST-NeRF to other state-of-the-art models in Table 5.3, we can see that our approach outperforms all models when applied in a scene-agnostic manner. ZeST-NeRF improves the results of other scene-agnostic methods by at least 15% across all metrics. In addition, a ZeST-NeRF model pre-trained on different sequences can be rapidly fine-tuned on a specific scene to produce results competitive with exhaustively optimised scene-specific approaches quickly. Qualitative results are shown in Figure 5.5. We can observe that our model produces significantly more accurate and pleasant results than the other methods. Approaches like MVSNeRF [15] or SVS [43] are able to reconstruct the correct background, but struggle to recover the dynamic objects in the scene. As the numbers from the ablation study suggest (Table 5.4), this is probably due to the necessity of having both a geometry and a dynamic network to recover static and dynamic features. Despite the superior results of our model, our results still suffer from some duplication artefacts and background blurriness. We can observe that this blurriness lessens when we fine-tune to each scene (see Figure 5.6). This may be caused by the static and dynamic networks trying to balance and preventing each other to obtain the best result. On the other hand, the NSFF [82] model is not really equipped to solve this scene-agnostic problem.

Table 5.4: **Ablation study.** Effect of each addition to the model. **Bold** is best result.

Geometry	Dynamic	PSNR↑	SSIM↑	LPIPS↓
×	×	12.67	0.4142	0.8277
✓	×	17.87	0.4533	0.4781
×	✓	20.19	0.5814	0.2447
✓	✓	**21.59**	**0.6239**	**0.2048**

5.5 Conclusion

The problem of generating virtual camera views from raw video is central to the application of audio-visual content production. In this chapter, we have introduced ZeST-NeRF, a novel and dynamic-scene representation technique that addresses the challenge of scene-agnostic “zero-shot” Novel View Synthesis for videos. By combining MVS and scene flow field estimation approaches, our proposed method enables the rendering of novel views in previously unseen scenes. The application of ZeST-NeRF proves particularly beneficial in media production scenarios, where numerous videos are available, and the need to generate new camera views without resorting to expensive scene-specific training is paramount.

Attempts to apply state-of-the-art techniques to tackle this problem by naively adding a temporal dimension have resulted in suboptimal outcomes, characterised by blurry results and a high prevalence of artefacts. In some cases, training these methods proved infeasible altogether. To overcome these limitations, we have proposed an algorithm that harnesses the generalisation power of MVS techniques. Our approach extends these techniques into the spatial and temporal dimensions, predicting scene flow field and occlusion weights. The inclusion of occlusion weights in our technique plays a crucial

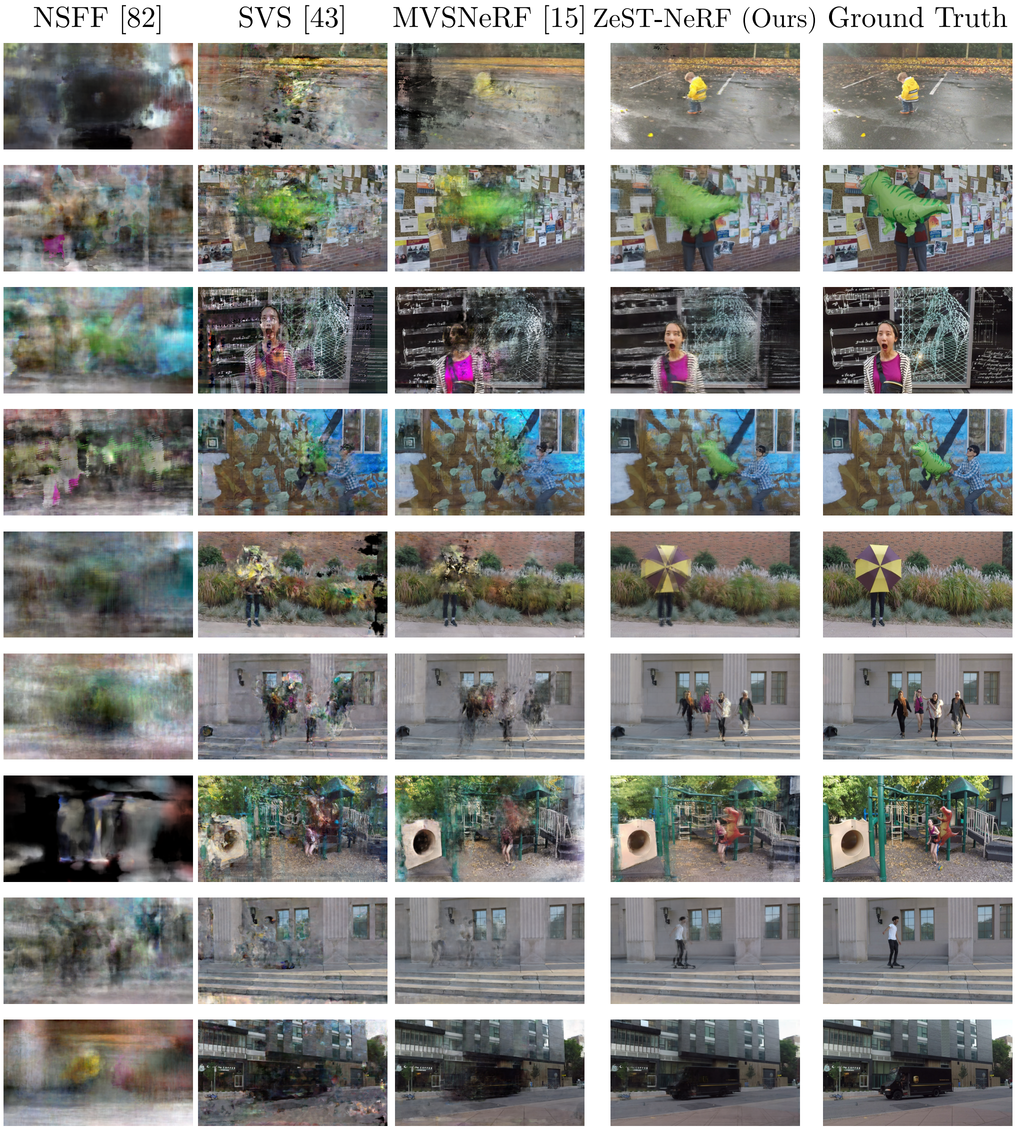


Figure 5.5: **“Zero-shot” qualitative results** on the Dynamic Scenes dataset [184]

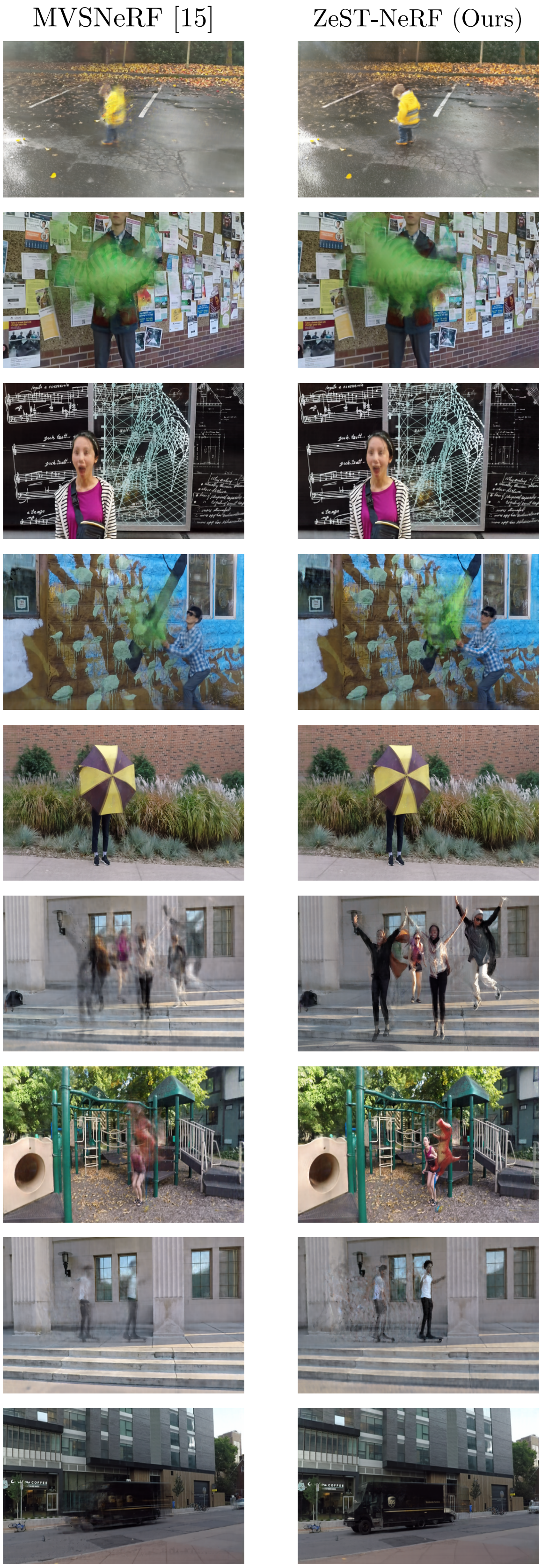


Figure 5.6: **Fine-tuned qualitative results** on the Dynamic Scenes dataset [184]

role in enhancing the fidelity of reconstructions, particularly in areas that are partially occluded. Simultaneously, the predicted flow field enables effective modelling of the temporal dimension within videos, providing a significant advantage over prior methods that focused solely on spatial reconstructions.

Despite the inherent complexity and ill-posed nature of the “zero-shot” NVS problem, it is important to acknowledge that our approach is not without its limitations. While ZeST-NeRF represents a significant step-change in performance compared to previous research, there are still challenges to address. Duplication artefacts and occasional blurriness may persist in certain scenes. Nevertheless, the advancements made by ZeST-NeRF offer promising possibilities for future research and development in the field of dynamic-scene representation, with applications in media production and in various other domains, including Virtual Reality (VR), Augmented Reality (AR), and 3D content creation.

Chapter 6

Conclusions and Future Work

In this thesis we tackled the challenging problem of Novel View Synthesis (NVS) in sparse camera setups. This was motivated by television production, with an aim to facilitate the coverage of resource-constrained events. The challenges encountered in these scenarios included a constraint on the number of available cameras and specialised equipment, feasible camera locations, and computational resources. Reconstructing a novel view of a scene was already a challenging and ill-defined problem which suffered from high ambiguity and uncertainty. These challenges were exacerbated by the physical limitations, as visible content of the scene was reduced dramatically, and the occlusions became more extreme.

Traditional computer vision methods relied on dense input data or specialised equipment to reconstruct 3D geometries of scenes. These approaches struggled to capture scene semantics and reconstruct high ambiguity areas. Our aim was to harness the extrapolating power of Deep Learning (DL) algorithms to develop a framework that could reconstruct these extremely sparse scenes. Furthermore, we were interested in approaches that were able to handle different camera arrangements and changes of scene.

We approached this problem as one of image generation, using inpainting and NVS techniques. With this in mind, we laid out a series of objectives in Section 1.2, namely,

1. Explore the use of generative inpainting techniques to the application of novel

view synthesis.

2. Develop a generative deep learning framework that exploits multi-view camera information and can generate novel content in unobserved regions avoiding visual artefacts.

3. Provide view-synthesis models with a generalisation mechanism to be applicable to diverse and sparse camera setups in varied scenes.

4. Explore the temporal consistency and generalisation of videos for the view synthesis problem.

6.1 Conclusions and limitations

In this chapter we summarise our technical contributions, their alignment with our objectives, their limitations, and potential avenues for future improvements.

Chapter 3 – SaiNet: Stereo aware inpainting behind objects with generative networks

Our first technical contribution in Chapter 3 addressed Objective 1. We introduced a deep learning architecture and training scheme that leveraged stereo camera information to reconstruct missing parts of an image. This inpainting model made use of two strategies to enforce stereo consistency: the introduction of an image stereo context, and the use of a stereo consistency loss. We trained our model in a self-supervised fashion using geometrically meaningful masks representing object occlusions. To create ground truth data we first generated a pool of binary masks extracted from a set of unrelated images, using edge detection and background propagation. We then pasted those masks on our training images according to a simulated disparity to obtain virtual object occlusions. Our architecture used Partial Convolutions [87] and edge-guided inpainting [109] to recover the image content behind the disocclusions.

We evaluated the performance of our framework over diverse datasets, including synthetic stereo datasets containing a multitude of objects, realistic-looking driving datasets, and a small real-world dataset. Our technique exhibited substantial performance improvements over state-of-the-art models by up to 50% PSNR, as well as achieving a competitive overall multi-view consistency of the reconstruction. Although our model fulfilled Objective 1 and delivered superior quantitative results compared to other approaches, there remains ample room for further enhancements in our methodology.

It would be interesting to further study the impact of mask size and placement on the inpainting process. Additionally, investigating the calibration of various hyper-parameters and reconstruction loss functions could enhance the effectiveness of the stereo-consistency loss. Furthermore, an end-to-end training approach that encompasses edge reconstruction may lead to more optimal results. However, it is worth noting that our current approach is capable of handling only minor changes in baseline and camera configurations that lie within the same plane, as is the case in a stereo camera setup.

Chapter 4 – SVS: Adversarial refinement for sparse novel view synthesis

In Chapter 4 we proposed the problem of SVS, a scenario characterised by the presence of a limited number of cameras widely spaced apart on different planes. With this second technical contribution, we tackled Objectives 2 and 3. The nature of this problem was inherently complex and ambiguous due to the sparsity of input data, resulting in large occluded areas across all views.

In this framework we employed Neural Radiance Fields (NeRFs), a 3D rendering approach known for producing high-quality photorealistic results. However, NeRF required dense inputs under ideal conditions, and necessitated per-scene optimisation involving several hours of processing. In contrast, our system was engineered to handle substantial camera motion using just three input views. Moreover, it demonstrated the capacity to generalise to new scenes that are unseen at training. Our pipeline consisted of a neural encoding volume which encoded scene geometry, along with a radiance field that sampled from this encoding volume. Adversarial training was used to diversify the sampling of occlusions and hallucinate content to fill in the gaps.

Our comprehensive evaluation showcased the superiority of our model over comparable approaches, especially in accommodating larger baseline changes. While other methods frequently grappled with issues such as 3D floating artefacts, blurring, and structural duplication, our model reduced these significantly. We conducted tests on two well-known datasets, DTU [65] and LLFF [103].

Despite accomplishing Objectives 2 and 3, our model still suffered from certain limitations. Notably, while GANs excel at generating new content, they suffer from the classic training instability, making model training more challenging. In addition, the difficulty of the problem dealing with extreme baseline disparities meant the complexity of the solution increased. This resulted in an ill-posed problem where the network sometimes struggled to differentiate between empty and occluded spaces, leading to fidelity issues in regions solely visible from one or none of the source views. Nonetheless, the pursuit of achieving full-fledged extreme SVS remains a work in progress, although our proposed approach represents a significant stride toward enabling more extreme and sparse rendering scenarios. Future exploration could entail more diverse camera arrangements, particularly those with substantial differences in distance from the scene being recorded, potentially involving a combination of camera frames ranging from close-ups to medium and full shots.

Chapter 5 – ZeST-NeRF: Using temporal aggregation for Zero-Shot Temporal NeRFs

In our third and final technical contribution, we addressed Objective 4 by expanding the scope of our system to the temporal dimension. While several temporal approaches based on NeRF [104] had been proposed, they faced similar limitations as their static counterparts. In aligment with our Objective 3, we could not rely on systems that needed to be carefully trained for hours per scene. We instead presented ZeST-NeRF, a novel NeRF-based temporal approach designed to enable zero-shot applicability to previously unseen scenes. Our methodology leveraged zero-shot multi-view synthesis techniques in conjunction with scene flow estimation, thereby enabling generalised modelling across both spatial and temporal dimensions of monocular videos.

The core of our architectural innovation consisted on the incorporation of a motion encoding volume based on a frame-aggregated PSV [36] to complement our existing geometry encoding volume. In this context, temporal data proved advantageous, as it facilitated information aggregation across multiple frames to disambiguate complex scenarios encountered in unfamiliar scenes. Furthermore, to enhance the precision of our reconstructions and assist the network in distinguishing between unoccupied and occluded spaces, we employed disocclusion weights as a regularisation mechanism based on visibility cues.

To empirically evaluate our approach, we conducted experiments using the Dynamic Scenes dataset [184]. Our results demonstrated that ZeST-NeRF succeeded in scenarios where other models fall short. Notably, our model presented a pioneering attempt at achieving zero-shot NeRF rendering, and it exhibited superior performance in capturing dynamic information compared to the naive application of conventional zero-shot static methods. Moreover, our model consistently produced reconstructions of substantially higher accuracy and visual appeal than existing models. However, despite achieving our predefined objectives, our model exhibited certain limitations. Artefacts like duplication persisted in certain scenes, particularly in regions characterised by temporal motion. In future, this may be tackled by applying an inpainting generative regularisation like adversarial training. Nonetheless, our approach constituted a significant advancement in performance compared to prior research efforts.

6.2 Future Work

Looking ahead, it would be interesting to explore the extension of ZeST-NeRF for reconstructing dynamic scenes using multi-view videos captured from multiple cameras. While some work has been undertaken in developing multi-view video NeRFs [81], current models typically rely on an extensive array of camera views. Exploring the integration between these techniques and ZeST-NeRF for achieving sparse few-shot generalisation remains a promising avenue for future research.

Given the challenges associated with the instability of adversarial techniques, an alternative approach could be to leverage more robust inpainting methods, such as diffusion models. These advanced methods hold promise for addressing uncertainty in highly occluded regions or areas characterised by substantial motion. In fact, recent work has already begun to explore the use of diffusion models as a means to regularise NeRFs [175].

Our research focus primarily centres on zero-shot algorithms capable of generalising to unseen scenes. This emphasis stems from the overarching objective of establishing efficient systems that do not necessitate retraining for each unique scene. Such systems prove invaluable when dealing with the rendering of numerous and diverse scenes or real-time rendering scenarios. However, zero-shot scene-agnostic techniques are not the sole option for achieving this goal, identifying more optimal algorithms and highly efficient representations may suffice to achieve fast “on-the-fly” rendering of new scenes. Some approaches in this direction have sought to optimise NeRF, like ENeRF [85] or instant Neural Graphic Primitives [107]. Very recently, the utilisation of 3D anisotropic Gaussians and splatting has gained popularity for 3D rendering [73]. This innovative method represents 3D scenes using clouds of 3D Gaussians, circumventing the use of neural networks and minimising computational overhead in empty spaces, unlike traditional radiance fields. The outcome is a model capable of producing high-resolution photorealistic outputs while drastically reducing training time and enabling real-time rendering. Nevertheless, it is important to note that this approach has its limitations, including artefacts in poorly observed regions, a reliance on dense input, and substantial GPU memory usage. As a result, it currently falls short of practicality for our objectives of creating cost-effective and accessible media production tools. Moreover, this technique is exclusively applied to static scenes, prompting further exploration into adapting it for representing non-rigid dynamics. Despite the room for improvement and the identified limitations, this direction holds significant promise to tackle the problem of NVS.

As we look to the future, the use of DL and generative methods for dynamic scene reconstruction promises to reshape the landscape of computer vision and graphics. These advancements are poised to become the cornerstone of future media production systems, offering efficient, adaptive, and cost-effective solutions for production techniques and real-time rendering. This transformative journey holds the potential to revolutionise the way we interact with 3D scenes and create photorealistic content, ultimately enabling free-viewpoint consumption of live and edited programming.

Bibliography

[1] Jack Allnutt, Rosie Campbell, Michael Evans, Ronan Forman, James Gibson, Stephen Jolly, Lianne Kerlin, Susan Lechelt, Graeme Phillipson, and Em Shotton. Ai in production: Video analysis and machine learning for expanded live events coverage. In *International Broadcasting Convention (IBC)*, 2018.

[2] Jack Allnutt, Rosie Campbell, Michael Evans, Ronan Forman, James Gibson, Stephen Jolly, Lianne Kerlin, Susan Lechelt, Graeme Phillipson, and Em Shotton. Ai in production: Video analysis and machine learning for expanded live events coverage. *Motion Imaging Journal*, pages 36–45, March 2020.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM transactions on graphics (TOG)*, August 2009.

[4] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[5] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Mojtaba Bemana, K. Myszkowski, H. Seidel, and Tobias Ritschel. X-Fields: Implicit Neural View-, Light- and Time-Image Interpolation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, October 2020.

[7] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, July 2000.

[8] Marcelo Bertalmio, Luminita Vese, Guillermo T, and Stanley Osher. Simultaneous Structure and Texture Image Inpainting. *IEEE Transactions on Image Processing*, August 2003.

[9] Aljaž Božič, Michael Zollhöfer, Christian Theobalt, and Matthias Nießner. DeepDeform: Learning Non-Rigid RGB-D Reconstruction With Semi-Supervised Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GaN training for high fidelity natural image synthesis. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[11] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 1986.

[12] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[13] Jia-Ren Chang and Yong-Sheng Chen. Pyramid Stereo Matching Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM transactions on graphics (TOG)*, June 2013.

[15] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from

Multi-View Stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[16] Shen Chen, Wei Ma, and Yue Qin. CNN-Based Stereoscopic Image Inpainting. In *International Conference on Image and Graphics (ICIG)*, November 2019.

[17] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1993.

[18] Tianlong Chen, Peihao Wang, Zhiwen Fan, and Zhangyang Wang. Aug-NeRF: Training Stronger Neural Radiance Fields With Triple-Level Physically-Grounded Augmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep Stereo Using Adaptive Thin Volume Representation With Uncertainty Awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo Radiance Fields (SRF): Learning View Synthesis for Sparse Views of Novel Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[21] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[22] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM transactions on graphics (TOG)*, July 2015.

[23] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based

inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2003.

[24] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, October 2004.

[25] Paul Debevec, Yizhou Yu, and George Borshukov. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In *Rendering Techniques '98. EGSR 1998. Eurographics.* 1998.

[26] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-Supervised NeRF: Fewer Views and Faster Training for Free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[27] Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[28] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. What Makes Paris Look Like Paris? In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2012.

[29] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM transactions on graphics (TOG)*, November 2017.

[30] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4D: Real-time performance capture of challenging scenes. *ACM transactions on graphics (TOG)*, July 2016.

[31] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural Radiance Flow for 4D View Synthesis and Video Processing. In

Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), September 2021.

[32] Alexei Efros and Thomas Leung. Texture Synthesis by Non-parametric Sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1999.

[33] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, June 2018.

[34] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Image-BART: Bidirectional Context with Multinomial Diffusion for Autoregressive Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[35] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. DeepView: View Synthesis with Learned Gradient Descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to Predict New Views from the World's Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[37] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic View Synthesis from Dynamic Monocular Video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, May 2021.

[38] Leon Gatys, Alexander Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. *Journal of Vision*, September 2016.

[39] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[40] Andrew Gilbert, Matt Trumble, Adrian Hilton, and John Collomosse. Inpainting of Wide-Baseline Multiple Viewpoint Video. *IEEE Transactions on Visualization and Computer Graphics*, December 2018.

[41] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, April 2017.

[42] Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. SaiNet: Stereo aware inpainting behind objects with generative networks. *AI for Content Creation (AI4CC) at CVPR*, 2022.

[43] Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. SVS: Adversarial refinement for sparse novel view synthesis. In *Proceedings of the 33rd British Machine Vision Conference (BMVC)*, 2022.

[44] Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. ZeST-NeRF: Using temporal aggregation for zero-shot temporal NeRFs. In *Proceedings of the 34th British Machine Vision Conference Workshops (BMVC)*, 2023.

[45] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS)*, December 2014.

[46] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, August 1996.

[47] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A Style-based 3D Aware Generator for High-resolution Image Synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, October 2021.

[48] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM transactions on graphics (TOG)*, November 2019.

[49] James Hays and Alexei A. Efros. Scene completion using millions of photographs. *ACM transactions on graphics (TOG)*, 2007.

[50] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[51] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM transactions on graphics (TOG)*, December 2018.

[52] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, October 1995.

[53] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. PM-Huber: PatchMatch with Huber Regularization for Stereo Matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, December 2013.

[54] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[55] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad

Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, January 2022.

[56] Minghui Hu, Yujie Wang, Tat-Jen Cham, Jianfei Yang, and P.N. Suganthan. Global Context with Discrete Diffusion in Vector Quantised Modelling for Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[57] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. EfficientNeRF Efficient Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[58] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep Volumetric Video From Very Sparse Multi-view Performance Capture. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[59] Q. Huynh-Thu and M. Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 2008.

[60] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM transactions on graphics (TOG)*, 2017.

[61] Matthias Innmann, Kihwan Kim, Jinwei Gu, Matthias Nießner, Charles Loop, Marc Stamminger, and Jan Kautz. NRMVS: Non-Rigid Multi-View Stereo. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[62] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-Time Volumetric Non-rigid Reconstruction. *12th European Conference on Computer Vision (ECCV)*, 2016.

[63] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[64] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[65] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large Scale Multi-view Stereopsis Evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[66] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Animashree Anandkumar, Minsu Cho, and Jaesik Park. Self-Calibrating Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[67] Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace Diffusion Generative Models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[68] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. GeoNeRF: Generalizing NeRF With Geometry Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[69] James T. Kajiya and Brian P. Von Herzen. Ray Tracing Volume Densities. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1984.

[70] Takeo Kanade, Peter Rander, and P. J. Narayanan. Virtualized Reality: Constructing Virtual Worlds from Real Scenes. *IEEE MultiMedia*, January 1997.

[71] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, December 2017.

[72] Bahjat Kawar, Stefano Ermon, Michael Elad, and Jiaming Song. Denoising Diffusion Restoration Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[73] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, July 2023.

[74] Mijeong Kim, Seonguk Seo, and Bohyung Han. InfoNeRF: Ray Entropy Minimization for Few-Shot Neural Volume Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[75] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2014.

[76] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-Specific Inpainting with Deep Neural Networks. In *Pattern Recognition - 36th German Conference (GCPR)*. 2014.

[77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, January 2012.

[78] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV)*, 1999.

[79] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In *Proceedings of the 11th IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2007.

[80] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, August 1996.

[81] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D Video Synthesis From Multi-View Video.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[82] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[83] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[84] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, August 2021.

[85] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient Neural Radiance Fields for Interactive Free-viewpoint Video. In *ACM SIGGRAPH Asia*, November 2022.

[86] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[87] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image Inpainting for Irregular Holes Using Partial Convolutions. *Proceedings of the European Conference on Computer Vision (ECCV)*, April 2018.

[88] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[89] Yebin Liu, Qionghai Dai, and Wenli Xu. A Point-Cloud-Based Multiview Stereo Algorithm for Free-Viewpoint Video. *IEEE Transactions on Visualization and Computer Graphics*, July 2010.

[90] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM transactions on graphics (TOG)*, July 2018.

[91] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM transactions on graphics (TOG)*, July 2019.

[92] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting Using Denoising Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[93] Xuan Luo, Yanmeng Kong, Jason Lawrence, Ricardo Martin-Brualla, and Steven M. Seitz. KeystoneDepth: History in 3D. In *Proceedings of the International Conference on 3D Vision (3DV)*, November 2020.

[94] Wei Ma, Mana Zheng, Wenguang Ma, Shibiao Xu, and Xiaopeng Zhang. Learning across views for stereo image completion. *IET Computer Vision*, 2020.

[95] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, December 2017.

[96] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. LookinGood: Enhancing Performance Capture with Real-time Neural Re-Rendering. *SIGGRAPH Asia Technical Papers*, November 2018.

[97] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[98] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[99] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1995.

[100] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, October 2021.

[101] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural Rerendering in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[102] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[103] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[104] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[105] B. Morse, J. Howard, S. Cohen, and B. Price. PatchMatch-Based Content Completion of Stereo Image Pairs. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, October 2012.

[106] Tai-Jiang Mu, Ju-Hong Wang, Song-Pei Du, and Shi-Min Hu. Stereoscopic image completion and depth recovery. *The Visual Computer: International Journal of Computer Graphics*, June 2014.

[107] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM transactions on graphics (TOG)*, July 2022.

[108] P.J. Narayanan, P.W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV)*, 1998.

[109] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. EdgeConnect: Structure Guided Image Inpainting using Edge Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, October 2019.

[110] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[111] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[112] Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, July 2021.

[113] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, June 2022.

[114] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis From Sparse Inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[115] Michael Niemeyer and Andreas Geiger. CAMPARI: Camera-Aware Decomposed Generative Neural Radiance Fields. In *Proceedings of the International Conference on 3D Vision (3DV)*, December 2021.

[116] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[117] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[118] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable Volumetric Rendering: Learning Implicit 3D Representations Without 3D Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[119] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural Scene Graphs for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[120] Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip

Davidson, Cem Keskin, Shahram Izadi, and Sean Fanello. Volumetric Capture of Humans With a Single RGBD Camera via Semi-Parametric Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[121] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[122] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hyper-NeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ACM transactions on graphics (TOG)*, December 2021.

[123] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context Encoders: Feature Learning by Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, April 2016.

[124] Eric Penner and Li Zhang. Soft 3D reconstruction for view synthesis. *ACM transactions on graphics (TOG)*, November 2017.

[125] Francesco Pittaluga, Sanjeev J. Koppal, Sing Bing Kang, and Sudipta N. Sinha. Revealing Scenes by Inverting Structure from Motion Reconstructions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, April 2019.

[126] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[127] Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for

Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March 2022.

[128] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. LOLNeRF: Learn from One Look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[129] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[130] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J. Guibas. CaSPR: Learning canonical spatiotemporal point cloud representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, December 2020.

[131] Jimmy SJ. Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, December 2015.

[132] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. StructureFlow: Image Inpainting via Structure-Aware Appearance Flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[133] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[134] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[135] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. NeRF for Outdoor Scene Relighting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[136] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, December 2015.

[137] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-Image Diffusion Models. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, August 2022.

[138] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Hao Li, and Angjoo Kanazawa. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[139] D. Scharstein, H. Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and P. Westling. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In *GCPR*, 2014.

[140] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[141] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[142] Steven M. Seitz and Charles R. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 1997.

[143] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1998.

[144] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D Photography using Context-aware Layered Depth Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, April 2020.

[145] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor Lempitsky. Textured Neural Avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[146] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[147] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, April 2015.

[148] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[149] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[150] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, July 2015.

[151] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, October 2020.

[152] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C. C. Jay Kuo. SPG-Net: Segmentation Prediction and Guidance Network for Image Inpainting. *British Machine Vision Conference (BMVC)*, May 2018.

[153] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[154] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the Boundaries of View Extrapolation With Multiplane Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[155] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, January 1998.

[156] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[157] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[158] L.M. Tanco and A. Hilton. Realistic synthesis of novel human movements from a

database of motion capture examples. In *Proceedings Workshop on Human Motion (HUMO)*, December 2000.

[159] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, 2020.

[160] Justus Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Nießner. Image-guided Neural Object Rendering. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, April 2020.

[161] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, August 2021.

[162] Alex Trevithick and Bo Yang. GRF: Learning a General Radiance Field for 3D Representation and Rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[163] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view Supervision for Single-View Reconstruction via Differentiable Ray Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[164] Jack Valmadre and Simon Lucey. General trajectory prior for Non-Rigid reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[165] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2017.

[166] B.G. Vijay Kumar, Gustavo Carneiro, and Ian Reid. Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimizing Global Loss Functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[167] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. In *ACM SIGGRAPH Asia*, December 2009.

[168] Minh Vo, Srinivasa G. Narasimhan, and Yaser Sheikh. Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[169] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[170] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning Multi-View Image-Based Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[171] Wang, Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, April 2004.

[172] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Get Out of my Picture! Internet-based Inpainting. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009.

[173] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end View Synthesis from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[174] Yuxin Wu and Kaiming He. Group Normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[175] Jamie Wynn and Daniyar Turmukhambetov. DiffusioNeRF: Regularizing Neural Radiance Fields With Denoising Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[176] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time Neural Irradiance Fields for Free-Viewpoint Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[177] Junyuan Xie, Linli Xu, and Enhong Chen. Image Denoising and Inpainting with Deep Neural Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, January 2012.

[178] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. SinNeRF: Training Neural Radiance Fields on Complex Scenes from a Single Image. In *17th European Conference on Computer Vision (ECCV)*, October 2022.

[179] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[180] Xudong XU, Xingang Pan, Dahua Lin, and Bo Dai. Generative Occupancy Fields for 3D Surface-Aware Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[181] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ra. Deep View Synthesis from Sparse Photometric Images. *ACM transactions on graphics (TOG)*, 2019.

[182] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[183] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic Image Inpainting with Deep Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[184] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel View Synthesis of Dynamic Scenes With Globally Coherent Depths From a Monocular Camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[185] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields From One or Few Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[186] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-Form Image Inpainting With Gated Convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[187] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative Image Inpainting with Contextual Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, January 2018.

[188] Ye Yu and William A. P. Smith. Depth estimation meets inverse renderingfor single image novel view synthesis. In *European Conference on Visual Media Production on (CVMP)*, 2019.

[189] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. STaR: Self-supervised Tracking and Reconstruction of Rigid Objects in Motion with Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[190] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[191] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. NeRFusion: Fusing Radiance Fields for Large-Scale Scene Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[192] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2018.

[193] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, August 2018.

[194] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward Multimodal Image-to-Image Translation. In *Advances in Neural Information Processing Systems (NeurIPS)*, October 2018.

[195] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 2004.

[196] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. Real-time non-rigid reconstruction using an RGB-D camera. *ACM transactions on graphics (TOG)*, July 2014.