Spatial Priors and Uncertainty for Enhanced Reconstruction in Computer Graphics and Vision

Dissertation

zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Markus Plack

aus
Kirchen (Sieg)

Bonn 2024

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer:	Prof. Dr. Matthias B. Hullin
Gutachter:	Prof. Dr. Reinhard Klein

Tag der Promotion:	18.12.2024
Erscheinungsjahr:	2025

Journey before Destination

Acknowledgements

First, I would like to thank my advisor, Prof. Dr. Matthias Hullin, for giving me the opportunity to go on this incredible adventure that resulted in three papers and culminated in one thesis containing them all and for his continued support and advice along the way. I am also very grateful to Dr. Christopher Schroers for offering me an internship at Disney Research | Studios, which ultimately led to the second publication of this thesis.

Of course, none of these works would look nearly as good without the immense (and partially late-night) efforts of my nine co-authors, whom I'd like to thank (in alphabetical order): Karlis Martins Briedis, Dr. Clara Callenberg, Dr. Abdelaziz Djelouah, Dr. Hannah Dröge, Prof. Dr. Markus Gross, Prof. Dr. Matthias Hullin, Monika Schneider, Dr. Christopher Schroers, Leif Van Holland. I also would like to thank Christiane Stuke and Simone von Neffe for keeping track of all the paperwork that so often felt like Vogon bureaucracy, and Dr. Hannah Dröge and Dr. Patrick Stotko for proofreading this thesis.

I am eternally grateful to my family for their endless support and encouragement, not only during my studies but throughout my life. And finally, the biggest of thanks to my beloved wife and evenstar Hannah, for loving me through this journey, even if it was a mess sometimes, and especially when I was a mess. Thank you for making this the best story there is.

Abstract

Many tasks in computer vision aim to reconstruct unknown quantities from observations of a scene, such as estimating depth from stereo vision, generating intermediate video frames, or revealing hidden shapes from transient measurements. In recent years, deep learning and differentiable rendering have become the methods of choice to tackle problems in those domains, but both exhibit a need for extensive computational resources. This thesis introduces innovative strategies that explicitly integrate our understanding of specific problems into reconstruction algorithms via spatial priors and uncertainty representations, improving both their efficiency and output quality by taking advantage of domain-specific peculiarities.

We begin by detailing how integrating rough shapes as priors into stereo matching can refine the depth estimation. In general, for a scene with no prior information, a comprehensive search across all possible values is performed to regress the disparity map between both images. However, in certain scenarios, such as stereo rigs embedded in setups containing other cameras, additional information can be used to improve the reconstruction. Our approach employs an efficient computation of the visual hull to reduce the search range of stereo matching, which, combined with various optimizations tailored to this use case, enables the accurate computation of depth at high resolutions.

Furthermore, we explore frame interpolation of rendered sequences where – in contrast to established methods – it is possible to generate and use additional data from the intermediate frame if necessary. By predicting the uncertainty of the interpolation output and incorporating partial renderings as priors, we devise a novel two-step model based on the transformer architecture that enhances the quality of the interpolated frames even for challenging content, as demonstrated quantitatively and qualitatively through a user study. This approach facilitates replacing the computationally costly rendering of a full sequence with a cheap interpolation of partial renderings.

Lastly, we tackle Non-Line-of-Sight reconstruction and demonstrate how the efficient implementation of a backward pass of a model-driven approach can lead to accurate reconstructions while reducing the runtime from hours or days needed by the baseline approach to minutes. This enabled us to explore different priors, and we show results using Gaussian blobs with an optional color component and total variation regularized depth maps. To address scenarios where the model assumptions deviate too much from the circumstances of real-world measurements, we introduce a background network inspired by neural representations and showcase its utility in capturing the remaining uncertainties.

Contents

I Introduction 1

1 Introduction 3

1.1 Reconstruction Problems and Challenges 4
1.2 Contributions 5
1.3 List of Publications 7
1.4 Thesis Outline 8

2 Background 9

2.1 Spatial Priors 9
2.1.1 Definition and Total Variation 9
2.1.2 Deep Learning Regularization 10
2.2 Uncertainty 12
2.2.1 Concepts 12
2.2.2 Uncertainty Estimation in Deep Learning 12
2.3 Stereo Vision 14
2.3.1 Epipolar Geometry and Image Rectification 14
2.3.2 Depth from Disparity 15
2.4 Optical Flow and Image Warping 16
2.4.1 Backward Warping 17
2.4.2 Forward Warping/Splatting 18
2.5 3D Geometry Representations 18
2.5.1 Explicit 19
2.5.2 Implicit 19
2.6 Transient Imaging 20
2.6.1 Measurement Setup 20
2.6.2 Light Transport 22

3 Related Work 25

3.1 Stereo Matching 25
3.1.1 Taxonomy 25
3.1.2 Deep Learning 26
3.1.3 Recurrent Architectures 27

3.2 Frame Interpolation 28
3.2.1 Classical Methods 28
3.2.2 Direct Methods 30
3.2.3 Kernel-Based Methods 30
3.2.4 Motion-Based Methods 31
3.2.5 Frame-Interpolation Evaluation 33
3.3 Non-Line-of-Sight Reconstruction 34
3.3.1 Backprojection and Transformations 34
3.3.2 Optimization and Rendering 35
3.3.3 Learned Reconstruction 35
3.3.4 Miscellaneous Modalities 36

II Publications **37**

4 VHS: High-Resolution Iterative Stereo Matching with Visual Hull Priors **39**
4.1 Introduction 40
4.2 Related Work 41
4.2.1 Matching Cost Volume 42
4.2.2 Iterative Updates in Stereo Matching 42
4.2.3 Efficiency 43
4.3 Visual Hull Stereo 43
4.3.1 Sparse Correlation 44
4.3.2 Visual Hull Prior 45
4.3.3 Iterative Disparity Refinement 46
4.4 Training Details 47
4.4.1 Dataset Preparation 47
4.4.2 Training Strategy 49
4.5 Experiments 49
4.5.1 Benchmark Evaluation 50
4.5.2 Visual Hull Guidance 50
4.5.3 Training Scheme 53
4.6 Conclusion 54

5 Frame Interpolation Transformer and Uncertainty Guidance **55**
5.1 Summary of the Publication 55
5.2 Author Contributions of the Publication 56

6 Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction **59**
6.1 Summary of the Publication 59
6.2 Author Contributions of the Publication 61

III Conclusion 63

7 Conclusion 65
7.1 Summary and Impact 65
7.2 Limitations and Outlook 67
7.3 Closing Remarks 69

Bibliography 71

List of Figures 97

List of Tables 99

IV Appendix 101

Publication: "Frame Interpolation Transformer and Uncertainty Guidance" 103

Publication: "Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction" 117

Part I

Introduction

Chapter 1

Introduction

Around 4.7 billion photos[1] are taken every day, more or less. While many of these are surely just pictures of our beloved pets or a well-presented and hopefully equally delicious meal, it also shows that we humans seem to have a desire to capture the world around us. Images are clearly a great medium for this, as they can be understood without language and contain considerable amounts of information. However, computationally extracting data from images is far from trivial, and this is where computer vision comes into play. And while funny cat pictures certainly have their own merits, research to improve computer vision methods for health or safety applications or to reduce the environmental impact of computation, is most certainly a worthwhile endeavor.

If we take a look at the history of computer vision, we can see a paradigm shift from traditional approaches to neural networks starting in the early 2010s. The former employed hand-crafted algorithms and interpretable models of all kinds of problem domains. They were typically based on explicit reasoning to put observations and the underlying scenes and image formation processes into relation, searching answers for questions like "What do natural images look like?" and modeling them assuming e.g. piecewise smoothness or non-local similarities. However, for many problems, they were no match in terms of performance to deep learning approaches trained on millions of images, which replaced this explicit modeling and reasoning with networks capable of capturing those correlations and priors implicitly within their parameters during the optimization.

Starting from the first relatively straightforward convolutional networks various improvements have been made in terms of network architectures and training procedures. Prominent examples are residual networks [He et al., 2016], attention/transformer models [Dosovitskiy et al., 2020], and improved training through better optimizers [Kingma and Ba, 2014], regularization approaches [Kukačka et al., 2017] or training regimes like those of generative adversarial networks [Goodfellow et al., 2014]. However, despite their impressive performance, several open questions remain. First and foremost we can ask how it could be possible to boost the quality of the predictions even further. While the most trivial solution would be to increase the size of our models, such an approach often yields diminishing returns at the cost of increased energy demand and therefore worse environmental footprint [Thompson

[1] Source: `https://photutorial.com/photos-statistics/`

et al., 2021]. Additionally, adding more training data to improve performance can also be problematic in many cases, for example, because the acquisition of more data is not easily possible as is often the case in a medical context, or because the manual labeling of data that is often outsourced to counties with deficient labor laws is morally questionable at best. Instead, we consider specific reconstruction applications and study how we can improve model performance or efficiency based on our understanding of the problem domains similar to traditional approaches to these problems by integrating additional information in the form of spatial priors and implementing an explicit handling of uncertainties in the reconstruction pipeline. In the following two sections, we will first give an overview of open challenges in the problem domains we have worked on (Section 1.1) and provide an overview of our contributions (Section 1.2).

1.1 Reconstruction Problems and Challenges

Computer vision reconstruction problems focus on extracting spatial or temporal information from limited observations for geometric processing, inference of further information, or generation of new content. One central task in this field is finding matching points in images with many applications ranging from calibration over depth estimation and 3D reconstruction to interpolation. One of the greatest challenges of this task is the size of the search space which is equal to the image size for each point of interest when no additional information about an image pair is given. For **stereo matching** applications, which aim to infer depth from simultaneously captured image pairs, one can use the knowledge of the camera intrinsics and extrinsics to reduce the search space to the epipolar line. Nevertheless, the size of the full search space approach still scales quadratically with the horizontal image resolution, which poses a challenge for high-resolution inputs. While alternative techniques have been proposed to simply avoid searching across all possible matches like coarse-to-fine hierarchical networks (e.g. Gu et al. [2020]), so-called all-pairs correlation networks [Lipson et al., 2021] have demonstrated remarkable performance, which is why we investigated how they can be adapted using more efficient matching to bring their performance to higher resolutions (Chapter 4). In addition, most current approaches employ relatively standard neural network architectures combined with some form of correlation computation or warping, processing the input images in a feed-forward fashion [Laga et al., 2020]. In such models, it is unclear how additional spatial information could be incorporated. While traditional optimization methods use regularization terms which can be extended to reflect other types of priors, there is no comparable functionality in learning methods and their integration remains an open problem. A similar conundrum can be observed in **frame interpolation** methods, many of which also rely on accurate matching between images. Again, most methods use common network architectures combined with correlation, warping, and/or kernel estimation and it remains unclear how partial information about the target frame could be incorporated. Such an approach would be particularly interesting in the context of video rendering, where one retains access to the renderer enabling it to produce additional data for the intermediate frame as we will see in Chapter 5. Of course, improving the quality of interpolation per se is also an ongoing research challenge. An open question is the definition of quality, as the optimization of classical distance metrics does not necessarily correlate with the perceived quality, which is important for algorithms targeting a wide audience [Reda et al., 2022; Kiefhaber et al., 2024]. In line with this, another open problem is the estimation of the interpolation quality at test time which would be a necessary component to guide a combined rendering and interpolation method. While having access to the ground truth that would be the output of the renderer makes solving this problem trivial, it would defeat the purpose of reducing the heavy computational burden of rendering. In other applications, the underlying scene parameters may be unknown, and a renderer can be used to reconstruct them. Such an approach has been demonstrated for **Non-Line-of-Sight (NLoS) reconstruction** [Iseringhausen and Hullin, 2020] where the goal is to infer the shape of a hidden object that is not directly visible from transient images of a diffuse relay wall. An open question in this domain, besides the effort to speed up the reconstruction, is the choice of the underlying scene representations, where the tradeoffs between representational power, (inverse) rendering efficiency, and regularization capabilities should be considered. In addition, such models apply various simplifications to optimize efficiency and it is unclear how the resulting losses in model accuracy and completeness can be compensated to still enable processing of real measurements with varying degrees of unknowns.

1.2 Contributions

In this thesis, we present our efforts to address these problems using spatial priors and uncertainty modeling. Fig. 1.1 gives an overview of the proposed methods, showcasing the application domains and as well as the novel approaches we have integrated into the reconstruction problems. The main contributions of our work can be summarized as follows:

Improving Stereo Matching of High-Resolution Images with Additional Inputs We propose to integrate visual hulls computed from auxiliary views of a scene into a disparity estimation network as spatial priors for the matching. We demonstrate how the integration improves the matching performance and further implement other optimizations to enable training and inference at high resolutions. To this end, we replace the dense all-pairs correlation with a sparse-dense *k*NN-correlation and propose a training scheme that further reduces the memory footprint by splitting the computational graph for the gradient backpropagation [Plack et al., 2024].

Predicting Uncertainty and Incorporate Partial Renderings for Frame Interpolation We introduce partial inputs of the target frame to the frame interpolation problem to boost output quality for the application in video rendering. We achieve this using a transformer-based architecture, where the masked inputs serve as priors for the interpolation. Since the output quality is unknown and cannot reliably inferred from the result, we add an uncertainty prediction to the network, which is trained on the true error of the output, and demonstrate that this approach is capable of identifying problematic regions in the interpolation, that can

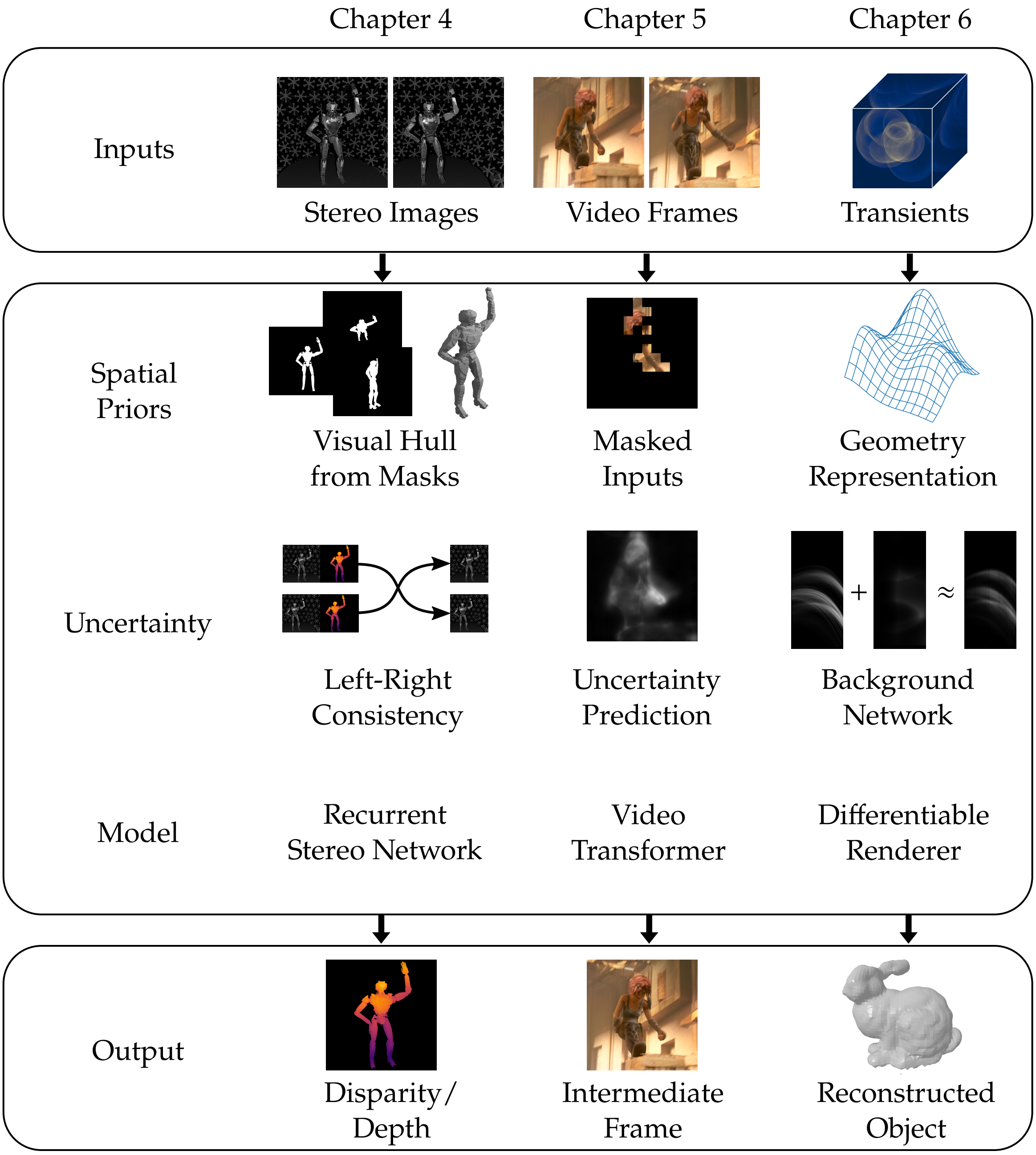


Figure 1.1: This thesis presents our efforts to integrate *spatial priors* into three different methods tackling three problem domains for improved reconstructions, either as additional inputs into neural networks (Chapters 4 and 5) or as regularization during optimization (Chapter 6). In all approaches, the *uncertainty* of the prediction or the underlying assumptions is an important factor to consider, which can be estimated from the outputs (Chapter 4), predicted from the inputs (Chapter 5), or optimized with the target (Chapter 6).

be rendered and added as partial inputs to a second pass to improve the quality [Plack et al., 2023a].

Accelerating and Expanding Differentiable NLoS Rendering We present a differentiable renderer targeting three-bounce transient imaging, which can be used for scene reconstruction or tracking applications of objects that are only observable via a diffuse relay wall. Our method improves upon a baseline renderer through an efficient implementation of gradient backpropagation, allowing us to explore scene representation priors for the differentiable reconstruction. In addition, we propose a background network to capture model inaccuracies and missing scene information and demonstrate its application on measured data [Plack et al., 2023b].

1.3 List of Publications

These publications and the preprint form the main part of this thesis:

- **Markus Plack**, Clara Callenberg, Monika Schneider, and Matthias B. Hullin.
  "Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction."
  IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023.
  DOI: `10.1109/WACV56688.2023.00308`
- **Markus Plack**, Karlis Martins Briedis, Abdelaziz Djelouah, Matthias B. Hullin, Markus Gross, and Christopher Schroers.
  "Frame Interpolation Transformer and Uncertainty Guidance."
  IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023.
  DOI: `10.1109/CVPR52729.2023.00946`
- **Markus Plack**, Hannah Dröge, Leif Van Holland, and Matthias B. Hullin.
  "VHS: High-Resolution Iterative Stereo Matching with Visual Hull Priors."
  arXiv preprint arXiv:2406.02552, 2024.
  DOI: `10.48550/arXiv.2406.02552`

In addition, I contributed to the following publications, which, however, are not part of this thesis:

- Javier Grau, **Markus Plack**, Patrick Haehn, Michael Weinmann, and Matthias B. Hullin.
  "Occlusion Fields: An Implicit Representation for Non-Line-of-Sight Surface Reconstruction."
  arXiv preprint arXiv:2203.08657, 2022.
  DOI: `10.48550/arXiv.2203.08657`
- Christopher Richard Schroers, Karlis Martins Briedis, Abdelaziz Djelouah, Ian McGonigal, Mark Meyer, Marios Papas, and **Markus Plack**.
  "Frame Interpolation for Rendered Content."
  US Patent App. 17/325,026, 2022.

1.4 Thesis Outline

The remainder of this thesis is structured as follows.

Part I: Introduction

Chapter 2 We give a summary of the foundations that are needed to understand the methods in this thesis. We outline the concepts of priors and uncertainty and explain the epipolar geometry of stereo setups and the connection between disparity and depth, followed by an overview of optical flow and image warping, which are basic building blocks of frame interpolation. Lastly, we discuss 3D geometry representations and transient imaging setups with the related light transport.

Chapter 3 We provide an overview of the related work in the application categories part of this thesis, namely stereo matching, frame interpolation, and NLoS reconstruction.

Part II: Publications

Chapter 4 We present our work "VHS: High-Resolution Iterative Stereo Matching with Visual Hull Priors" [Plack et al., 2024] which already appeared as a preprint.

Chapter 5 We summarize the peer-reviewed publication "Frame Interpolation Transformer and Uncertainty Guidance" [Plack et al., 2023a].

Chapter 6 We summarize the peer-reviewed publication "Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction" [Plack et al., 2023b].

Part III: Conclusion

Chapter 7 To conclude the main part of this thesis, we summarize the works presented herein and discuss the impact of our methods as well as their limitations along with an outlook into possible future work.

Part IV: Appendix

This last section contains copies of the published works that form chapters 5 and 6.

Chapter 2

Background

This chapter provides the necessary foundations to understand the methods introduced in this thesis. We start by outlining the basics of spatial priors in Section 2.1 and uncertainty in Section 2.2. For a better understanding of the addressed problems, we introduce the foundations of stereo vision in Section 2.3 followed by an explanation of optical flow and image warping in Section 2.4 which are an essential building block of frame interpolation methods. In Section 2.5 we outline 3D geometry representations and in Section 2.6 we give an overview of transient imaging technology and the underlying light transport model.

2.1 Spatial Priors

Prior knowledge in the form of probability distributions can drastically improve the performance of computer vision systems. While a thorough introduction is well beyond the scope of this work, we will focus on the aspects relevant to this thesis and briefly cover the definition of priors and their realization in total variation (TV) regularization as well as their application to deep learning models.

2.1.1 Definition and Total Variation

The concept of prior probabilities arises in Bayesian statistics and expresses the probability of a proposition before an observation of the evidence. In terms of Bayes' theorem

$$P(u|f) = \frac{P(f|u)P(u)}{P(f)}, \tag{2.1}$$

where u is the proposition and f the evidence, $P(u)$ denotes prior probability. In the case where no information about u is available, one can use the principle of indifference, which assigns equal probability to all outcomes and is therefore called a non-informative prior. Other examples of non-informative priors are minor restrictions on the values such as non-negativity. If more is known about u, for example from previous experiments or knowledge about similar propositions, $P(u)$ is called an informative prior. This can range from weakly informative priors, which mostly act as regularization in the inference, to strong priors, where the observation of f only marginally influences the posterior distribution $P(u|f)$.

As the focus of this thesis is on methods that operate in 3D scene or 2D image space, we are interested in spatial priors, by which we mean prior information that is localized in space (Chapters 4 and 5) or in the form of a smoothness assumption across space (Chapter 6). For the latter, the TV norm originally introduced for image denoising [Rudin et al., 1992] is used, which can be motivated from maximum a-posteriori probability (MAP) estimates as follows. Here, we are looking for the noise-free image $\hat{u}$ that maximizes the posterior probability given the noisy image f:

$$\hat{u} = \arg\max_u P(u|f) \tag{2.2}$$

Using Bayes' Theorem and dropping the constant term, this can be rewritten as a minimization of the negative log-likelihood

$$\hat{u} = \arg\min_u - \log(P(f|u)) - \log(P(u)). \tag{2.3}$$

In the context of variational methods, the first term is known as the data term and the second one is the regularization. With the assumptions of a Gaussian noise model and a Laplace distribution of the image gradients

$$P(f|u) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|u-f\|^2}{2\sigma^2}\right), \quad P(u) = \frac{1}{2\beta}\exp\left(-\frac{\|Du\|_1}{\beta}\right), \tag{2.4}$$

where D denotes the finite difference matrix, we get the TV regularized denoising

$$\hat{u} = \arg\min_u \frac{1}{2}\|u-f\|^2 + \lambda\|Du\|_1, \tag{2.5}$$

where λ is a weighting parameter based on the distribution parameters σ and β. Adding a linear operator to the data term, one can tackle various other inverse imaging tasks like deblurring, super-resolution, or computed tomography reconstruction. Such problems are typically solved using gradient descent type algorithms, including e.g. line search variants [Stanimirović and Miladinović, 2010], proximal algorithms [Parikh and Boyd, 2014] or alternating direction method of multipliers (ADMM) [Boyd et al., 2011]. We demonstrate an application of the TV regularization as a scene prior of depth and albedo maps in a differentiable rendering-based NLoS reconstruction method in Chapter 6.

2.1.2 Deep Learning Regularization

In deep learning approaches to inverse vision problems, the goal is to train a neural network on a training dataset such that the prior distribution is implicitly learned and encoded in the network weights. Several design choices of the training play a key role in solving these tasks efficiently and in a way that generalizes well to the test set. Similar to priors in the form of regularization terms in traditional loss function minimization, Kukačka et al. [2017] propose a taxonomy of such regularization techniques for deep learning:

Data Given the above definitions of the prior as knowledge from e.g. previous experiments, the choice of training data is a natural steering device to control the training. Beyond that, various augmentation methods that operate not only on input data but also on the target domain or hidden features can be interpreted as regularization techniques. Popular examples are image transformations like scaling and color transformations applied to the inputs that effectively aim at making the method invariant to such perturbations, dropout on the network weights [Srivastava et al., 2014] inducing learning of redundant features, and label smoothing [Szegedy et al., 2016] to discourage over-confidence.

Network Architecture The next choice after the selection of input and target data is that of the processing network layers which directly influences the possibilities of identifying patterns in the training data. For example, the use of convolutional layers and U-Nets [Ronneberger et al., 2015] assumes that the source and/or target domains have an inherent spatial structure and hierarchy. It targets learning of translation invariant and spatially localized features, which explains their success in imaging applications. Other examples are recurrent neural networks such as long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] or gated recurrent unit (GRU) [Cho et al., 2014] which induce temporal or sequential relationships between tokens as naturally implied in audio signals, texts or videos, or the attention mechanism/transformer architecture [Vaswani et al., 2017] which targets global relationships between tokens. Those architectures as well as other design choices such as the selected activation function, pooling operations, or dilation can be seen as implicit priors selected based on our knowledge of the problem domain.

Error Function The metric optimized during training of the neural network is chosen to steer the output in a way that matches the desired application. Aside from that, error metrics can also have a regularizing effect e.g. to handle class imbalance [Yan et al., 2003], or implicitly in multi-task learning [Ruder, 2017].

Regularization Term Perhaps the most obvious connection to traditional regularization is found in deep learning regularization terms that are added to the error metric to form the full loss function. This includes the commonly used weight decay (see e.g. Goodfellow et al. [2016]) which adds an L_2 loss on the network weights implying a normal distribution as their prior.

Optimization Finally, the selection of the training procedure has a major influence on the performance and generalization capabilities of a model. As an example, it has been shown that stochastic gradient descent can overcome saddle points for non-convex optimization [Ge et al., 2015]. Overall, optimization regularization techniques can be further categorized into methods for *initialization*, which include pre-training procedures, *updates*, like the popular Adam [Kingma and Ba, 2014], and *termination*, which aim at preventing overfitting, e.g. through observation of a validation set.

In our methods, we induce prior information into the neural network in the form of additional inputs during training and testing as visual hulls for stereo matching (Chapter 4) and masked intermediates for frame interpolation (Chapter 5).

2.2 Uncertainty

Some concepts of uncertainty are closely related to the previously discussed priors, e.g. unknown variables or stochastic processes are prominent sources of uncertainty. We will briefly outline those concepts and discuss some applications of this theory in the context of machine learning methods.

2.2.1 Concepts

Uncertainty is commonly classified as aleatoric or epistemic. The former describes a stochastic uncertainty (*alea*, Latin for dice), that cannot be reduced by additional knowledge, while the latter captures the remaining uncertainties (*epistēmē*, Greek for knowledge). Let us consider as an example the tossing of a fair coin where the probability of heads or tails is 50% each reflecting our uncertainty of the situation before a toss. No additional knowledge can help us to infer more about a future sample meaning that we have aleatoric uncertainty in this case. An example of epistemic uncertainty would be the assertion of the correctness of a factual statement in a foreign language. Without any knowledge of the language and applying the principle of indifference we again have a situation where both outcomes (i.e. the statement is correct or incorrect) have an equal chance of 50%. However, this uncertainty can be reduced by learning the language and the corresponding facts. Note that both types of uncertainty do not necessarily occur exclusively, so in many cases, both types contribute to the total uncertainty, which is also called predictive uncertainty of a model. In addition, it might not always be clear which type some effect can be attributed to as it relies on the assessment of what is possible by the addition of more knowledge, and sometimes the distinction might even not be necessary at all. For example situations in machine learning where – after the training of the model – only a single decision or prediction for some given inputs is of interest, an analysis of the types of uncertainties is ineffective. Nevertheless, the quantification of uncertainty is of great importance in many fields including but not limited to medical and safety applications [Hüllermeier and Waegeman, 2021].

In this thesis, we propose a method that handles epistemic uncertainties in a simplified inverse transient renderer such as model inaccuracies and ignored scene space (Chapter 6).

2.2.2 Uncertainty Estimation in Deep Learning

Uncertainty estimation techniques in deep learning can be categorized depending on the number of neural networks used for the estimation and the determinism of their prediction.

We provide a summary of those categories based on the work of Gawlikowski et al. [2023], to which we refer for more details:

Single Deterministic Methods This first class contains models where the uncertainty estimation is part of the network or can be derived from a single deterministic prediction. This includes in a broad sense typical classification networks, which assign probabilities to all classes usually by applying a Softmax to the network outputs, since the distribution of the probability values can be interpreted as the certainty of the output. However, it has been observed that network predictions often tend to be over-confident, which makes the interpretation difficult and gives rise to various alternative approaches [Gawlikowski et al., 2023].

Bayesian Neural Networks Instead of finding a point estimate of the optimal parameters as done in standard network training, the aim of Bayesian neural networks is to infer the posterior distribution of parameters given the training data. Then, the prediction for a given input can be estimated e.g. using Bayesian model averaging, but more importantly, this also allows quantifying the uncertainty of the prediction. Note that this approach requires a prior distribution for the network weights, where a common choice is a Gaussian prior whose MAP estimate is equivalent to training with the L_2 regularization described in Section 2.1.2 [Arbel et al., 2023].

Ensemble Methods The underlying idea of ensemble methods is to reduce the predictive error by computing the solution based on a set of predictions of so-called inducers (i.e. neural networks in our case), which ideally produce a diverse but accurate set of outputs. Nevertheless, having access to such a set allows estimating the uncertainty based on the variability of predictions [Sagi and Rokach, 2018].

Test-Time Augmentation Methods Without access to different networks, augmentation of the inputs same or similar to the processing during training can be used as an alternative approach to generate a set of outputs from which the uncertainty can be estimated [Gawlikowski et al., 2023].

Aside from the estimation of uncertainty, other means can be used which might be more informative to the user. This is especially true for vision tasks like super-resolution where the problem is under-determined and therefore the solution is not unique. In those applications, a quantification of the variability is not easily interpretable by a user, which gives rise to solution space exploration methods like the super-resolution approach of Bahat and Michaeli [2020] or the semantically-guided sparse computed tomography (CT) reconstruction of Dröge et al. [2022].

Left-right consistency checking in stereo vision can be seen as a variant of ensemble methods for uncertainty estimation (Chapter 4), and our frame interpolation method uses a single deep neural network that approximates the expected error of their prediction (Chapter 5).

2.3 Stereo Vision

Stereo vision aims to recover the depth at each pixel of an image from matches with the corresponding point in a second image and triangulating the points in space. It is a passive method opposed to active approaches like Time-of-Flight (ToF) cameras (e.g. Kolb et al. [2008]), light detection and ranging (LiDAR) sensors (e.g. Raj et al. [2020]), and structured light (e.g. Scharstein and Szeliski [2003]), where the latter works conceptually similar to stereo vision. We will briefly discuss the concept of epipolar geometry and image rectification which simplifies the search for correspondences and discuss how the depth is computed from those matches and the knowledge of the stereo camera setup.

2.3.1 Epipolar Geometry and Image Rectification

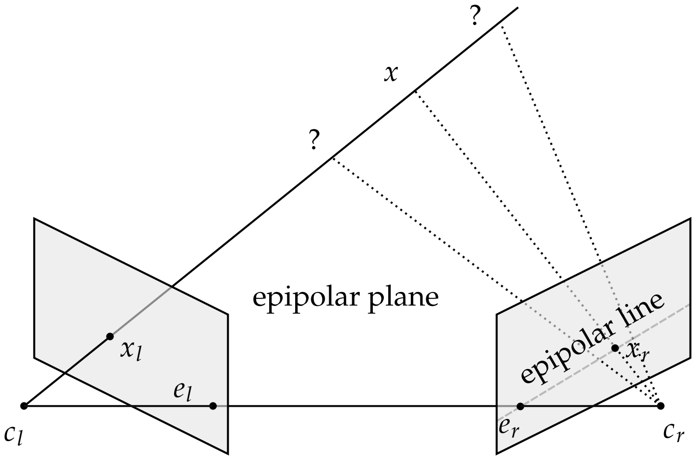


Figure 2.1: Epipolar Geometry of a stereo setup, where the camera centers c_l, c_r, and the world position x form the epipolar plane. Projecting the camera centers into the other camera's image plane we get the epipoles e_r, e_l. Any possible point on the viewing ray defined by c_l and x_l gets projected onto the epipolar line in the right camera.

In the following, we assume that our images come from an undistorted perspective camera. For most devices, this calls for a calibration step before the measurement using e.g. Brown's distortion model [Brown, 1996]. Now the search for the corresponding point of each pixel can be reduced to a search along the epipolar line instead of a naive search across the whole image as depicted in Fig. 2.1. For a given point x_l in the left image we want to find the world space position x at an unknown depth that it depicts. In this setup, the camera centers c_l and c_r, and the world point x form the so-called epipolar plane, and its projection into the other image gives us the epipolar line of x. Consequentially, all points on the line defined by c_l to x are projected onto the epipolar line.

This already simplifies the search for x_r from 2D to a 1D search for correspondences, but the complexity can be reduced even further by applying stereo image rectification resulting in

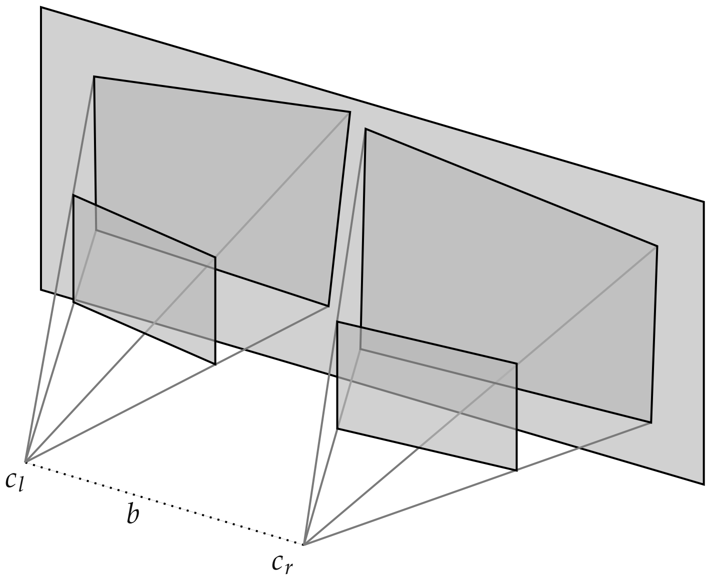


Figure 2.2: Stereo Rectification by reprojecting the images onto a common image plane parallel to the baseline b resulting in horizontal epipolar lines for all points (not shown here).

image pairs where all epipolar lines are horizontal. This can be achieved by reprojecting both images onto a common image plane as depicted in Fig. 2.2. Note that the plane needs to be parallel with the baseline, i.e. the line between c_l and c_r, and that those reprojections can be represented by a homography. We refer to Loop and Zhang [1999] for more details. For rectified stereo images, a wide variety of algorithms exist to find matches across the horizontal lines for each pixel ranging from simple intensity correlation matching approaches over regularized optimization methods to a multitude of learning-based approaches as outlined in Section 3.1.

2.3.2 Depth from Disparity

Once the disparity d, i.e. the image space distance between corresponding points in the rectified stereo images, is found we can compute the corresponding depth z from the baseline b and the focal length f as

$$z = \frac{fb}{d}. \tag{2.6}$$

This follows from multiple applications of the intercept theorem (compare Fig. 2.3) with $d = x_l - x_r$ as

$$\frac{b}{b-d} = \frac{q_0 + q_1}{q_0} = \frac{p_1}{p_0} = \frac{z}{z-f} \tag{2.7}$$

which yields

$$zb - fb = zb - zd \tag{2.8}$$

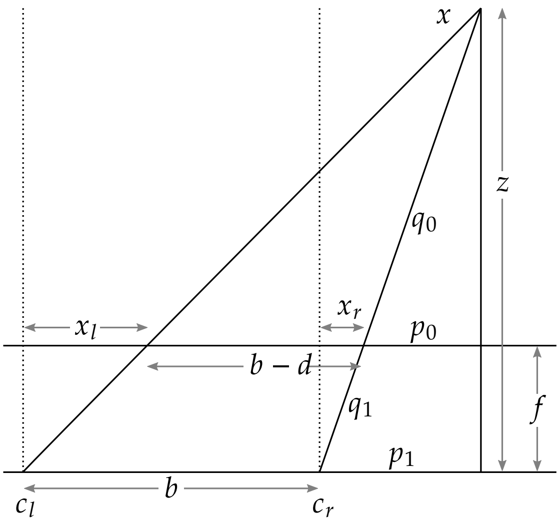
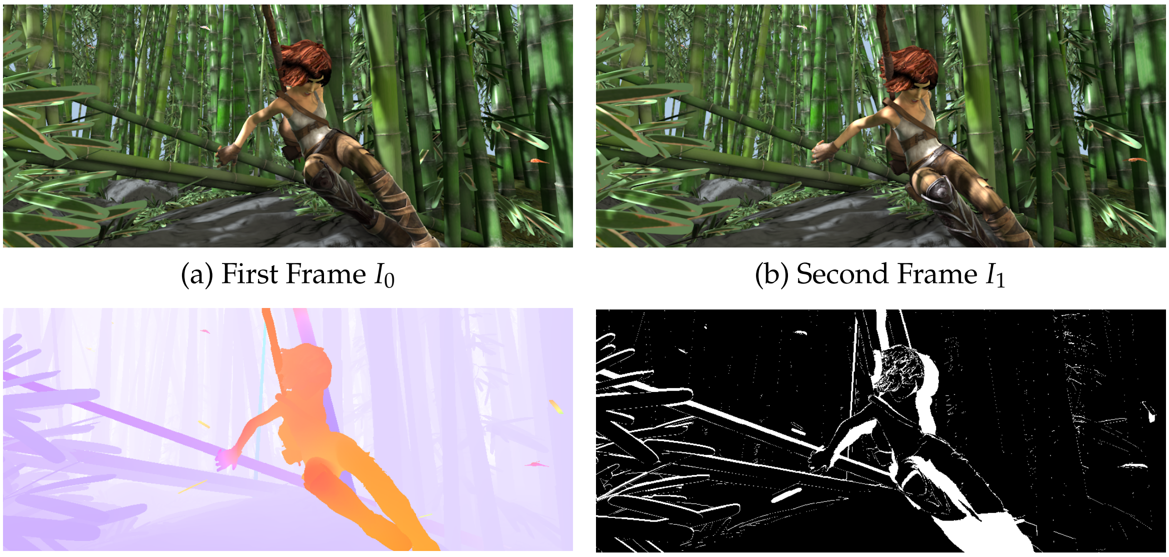


Figure 2.3: Computing depth z from disparity d with the intercept theorem via the rightmost triangles $(q_0 + q_1, p_1, z)$ and $(q_0, p_0, z - f)$. Note that $b - x_l + x_r = b - d$ and that the actual values of the intermediaries p_0, p_1, q_0, and q_1 are of no concern as they only bridge the gap to z and f.

and finally Eq. (2.6). Note that in this formulation, depth is not the Euclidean distance between the camera center and the point in 3D space ($\|x - c_l\|_2$), but rather the z coordinate of the point in the cameras reference frame. Those two values, however, are related by a projection onto the principal axis. Additionally, care must be taken to convert between pixel coordinates in image space and world space coordinates. Note that it is common to estimate the disparity for the left camera, but any such method can trivially compute the disparity of the right view by simply mirroring the images horizontally and switching the inputs.

2.4 Optical Flow and Image Warping

Knowing the correspondences between all pixels of two successive frames in a video is important for many video processing tasks. This problem is, however, much harder than the previously discussed stereo matching, since correspondences are no longer restricted to epipolar lines but can occur in the whole image, and temporal movement needs to be taken into account, by both the camera and the different parts of the scene, possibly in many different directions and with varying magnitudes. Figure 2.4 shows an example of the optical flow field between two frames in a video along with the occlusion mask, using the flow visualization technique established by Baker et al. [2011]. Similar to stereo vision, where left or right disparity can be computed, we can search for the forward or backward flow, which is often denoted by an arrow as forward flow $f_{0\to1}$ from frame I_0 to I_1 or backward flow $f_{1\to0}$ from frame I_1 to I_0 (and vice versa for the occlusion). We refer to Zhai et al. [2021] for an overview of optical and scene flow estimation methods. Those optical flow fields can be used to align images or, as prominently used in many frame interpolation methods, feature



(a) First Frame I_0

(b) Second Frame I_1

(c) Forward Flow $f_{0\to1}$

(d) Forward Occlusion $O_{0\to1}$

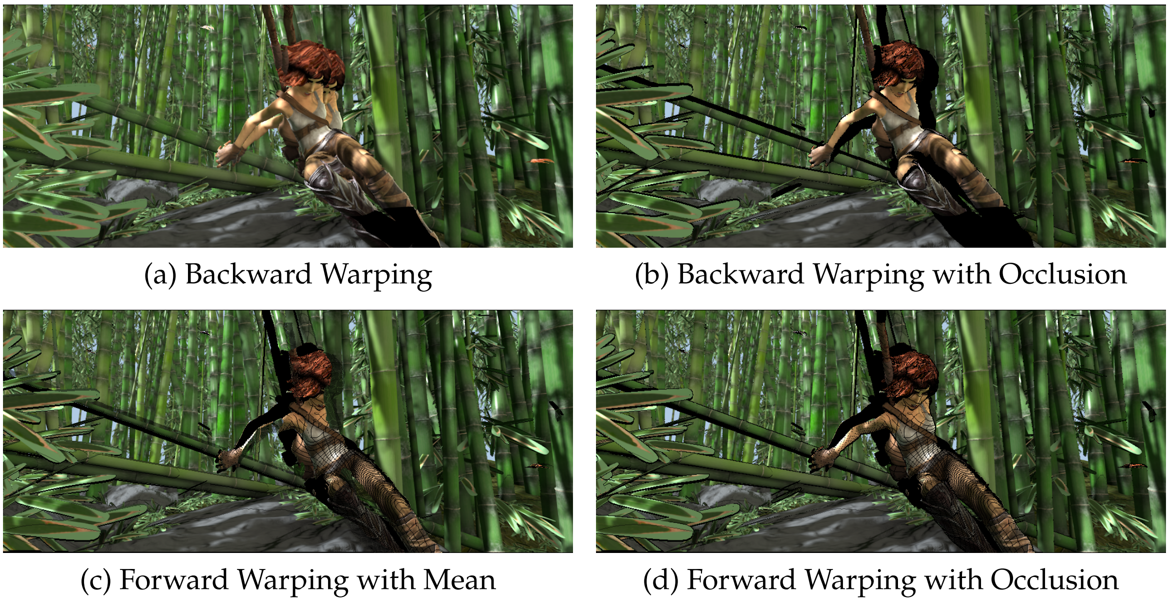
Figure 2.4: Example of the forward flow $f_{0\to1}$ (c) from the first frame I_0 (a) to the second frame I_1 (b) and the forward occlusion $O_{0\to1}$ (d) identifying all points where the target of the optical flow does not depict the same object as the source or is not within the frame.

maps between both frames. The two most commonly used approaches for this are backward warping and forward warping as described in the following sections.

2.4.1 Backward Warping

The idea of backward warping uses, as the name implies, the forward flow to backward warp content from the other frame. This can be easily implemented as a sampling operation, where the optical flow vector is added to the image coordinate of each pixel to get a target coordinate at which the other frame is sampled. Figure 2.5 shows an example output of this operation. Note that without any handling of occlusion, content may be sampled more than once, resulting in e.g. repeating patterns. Since this operation is easily differentiable with respect to both the input image and the optical flow vectors when using bilinear interpolation, it is a common module in many deep learning methods for both optical flow estimation and frame interpolation. To achieve the best results, care must be taken to sample the images correctly, such that e.g. zero flow will indeed sample the pixel value at the exact same position in the other image[1]. The greatest advantage of this method is its simplicity and the fact that, without occlusion, the resulting image will be smooth and without gaps. Additionally, the flow can be used to sample from any other frame using a scaling factor, assuming that all motion remains linear over the respective windows. By scaling $f_{0\to1}$ with -1 for example, the preceding frame I_{-1} can be sampled, while a factor of 2 allows sampling from the frame I_2.

[1] Compare e.g. the "align_corners" parameter of Pytorch's grid sampling method `https://pytorch.org/docs/stable/generated/torch.nn.functional.grid_sample.html`


(a) Backward Warping

(b) Backward Warping with Occlusion

(c) Forward Warping with Mean

(d) Forward Warping with Occlusion

Figure 2.5: Backward warping (a,b) and forward warping (c,d) examples based on Fig. 2.4. Note the duplicates in (a) and the ghosting in (c) from not handling the occlusion correctly. Since a simple nearest neighbor splatting was used in (c,d), non-smooth motions result in a tattered appearance.

2.4.2 Forward Warping/Splatting

Something that is not possible using backward warping, however, is to align the features of the frame with the known flow I_0 to the frame at some other time I_t. This operation can be realized using forward warping, which distributes the color or features of I_0 using the flow $I_{0\to t}$. This is also known as splatting since the content of the frame is "drawn" onto the other frame. The greatest advantage of this approach is its ability to warp the content to any frame (again assuming a linear motion). Specifically for frame interpolation, the features of I_0 can be warped to $I_{\frac{1}{2}}$ using the scaled flow $\frac{1}{2}f_{0\to 1}$, and vice versa for I_1 and $\frac{1}{2}f_{1\to 0}$.

Unlike backward warping, even non-occluded regions can result in images that have holes if the flow fields are not smooth enough and insufficient splatting like nearest neighbor is used. Additionally, the same pixel can receive data from multiple points that need to be blended to look correctly. This is especially problematic if the occlusion is not known, as demonstrated in Fig. 2.5. However, those drawbacks do not make forward warping an uninteresting choice as demonstrated by Niklaus and Liu [2020] and Niklaus et al. [2023], to which we refer for a more thorough treatment of the matter for frame interpolation.

2.5 3D Geometry Representations

For completeness, we provide a brief summary of 3D geometry representations with a focus on those that are relevant in the context of this thesis.

2.5.1 Explicit

One of the simplest geometry representations are point clouds, which can e.g. be computed from stereo matches by projecting the pixels into 3D space using the known depth and camera parameters. They are represented as an unordered list of 3D vectors and naturally do not contain any spatial relationships. Additional data such as color can be attached to each point and they can easily be displayed using projective matrices. However, they do not contain any surface information and are of infinitesimal size, which makes them unsuitable for the representation of closed surfaces. To achieve a satisfying rendering of point clouds additional techniques like elliptical weighted average filtering [Zwicker et al., 2001] or an extension like surfels ("surface elements") can be used, which use oriented discs of finite size to represent geometry [Pfister et al., 2000].

One of the most common representations in graphics applications are meshes, which are defined as a set of vertices $v_i \in \mathbb{R}^3$ and a set of faces, typically in terms of triangles $f_i \in \mathbb{N}^3$. Aside from assigning normals or colors to the vertices, textures can be used to add details to the surface such as albedo, normal, or height, as well as other parameters of the bidirectional reflectance distribution function (BRDF), which can be sampled using barycentric coordinates to interpolate UV coordinates.

2.5.2 Implicit

A common base to implicitly represent geometry is a 3D function $f : \mathbb{R}^3 \to \mathbb{R}$ often discretized as a 3D volume $\mathbb{R}^{x \times y \times z}$. By restricting the values to be in $\{0, 1\}$ we get an indicator function, where 1 represents points inside the object and 0 space outside. The surface is then implicitly given as the boundary of the indicator function. More information is contained in the signed distance functions (SDFs), where the absolute value of each entry describes the distance to the object's surface. The sign indicates whether the point is inside or outside and the surface lies at all points x where $f(x) = 0$. Alternatively, one can interpret the function as a density and assign the surface to some iso-level. To compute a mesh from such a volume, marching cubes [Lorensen and Cline, 1987] can be applied, which builds on the observation that for each local cube of 8 voxels and a given iso-value, there are – after removing symmetry and rotation – 14 cases which represent a surface.

Note that the volume itself can also be implicitly given through other means. For example, a sum of Gaussians can be used to represent the density of the volume [Iseringhausen and Hullin, 2020], which can be useful in geometry optimization (see Chapter 6). Alternatively, the SDF can be represented using a trained neural network as demonstrated e.g. by Park et al. [2019]. The latter is closely related to Neural Radiance Fields (NeRFs), which are also trained to predict a density for each point in space, but also predict radiance and are used in conjunction with a differentiable volume renderer for novel view synthesis [Mildenhall et al., 2020].

2.6 Transient Imaging

In the previous sections, the input images were captured from standard cameras used in everyday life and show a view of a scene as a steady-state image that closely resembles human vision. While this is sufficient for many vision problems, certain tasks like NLoS reconstruction benefit from additional information, e.g. by including our knowledge that the propagation of light is not instantaneous but that electromagnetic waves travel at the speed of light which is $299,792,458\frac{m}{s}$ in a vacuum. The goal of *transient imaging* is to explicitly take this finite speed into account for simulations, models, and measurements as introduced by Kirmani et al. [2009]. In this section, we will focus on the specific use case of NLoS imaging, and discuss the measurement as well as how it can be modeled. In this application, the goal is to retrieve the shape of an object that is hidden from the direct line of sight through an indirect observation of reflected light on a diffuse relay wall by probing the scene with ultra-short light pulses that are emitted toward the wall.

2.6.1 Measurement Setup

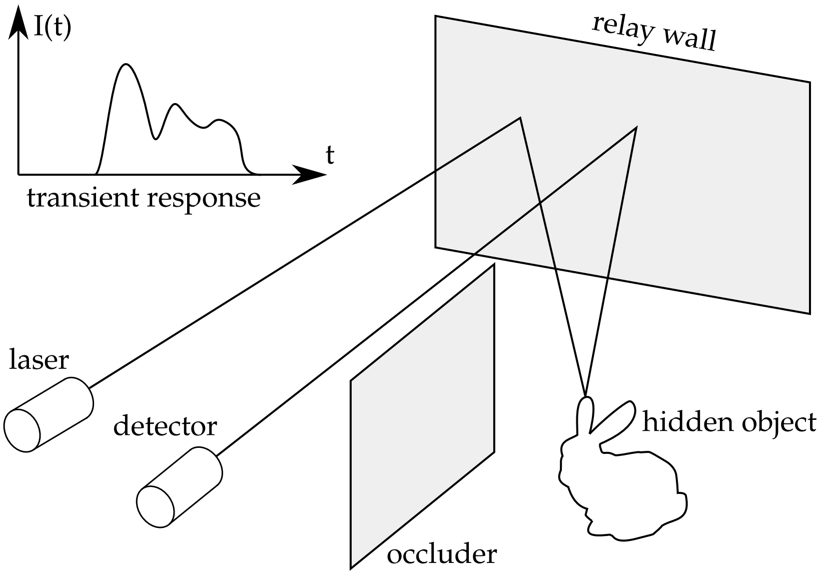


Figure 2.6: Prototypical NLoS measurement setup to capture an object that is hidden behind an occluder and only observable via the relay wall. This figure shows one possible light path going from the laser that emits an ultra-short light pulse towards the hidden object via the relay wall and is recorded by the time-resolved detector after another reflection on the wall. For one laser and detection orientation, this setup measures a histogram of arrival times as shown on the top-left.

We start by describing a prototypical setup used for NLoS reconstructions based on the seminal work of Velten et al. [2012] which is depicted in Fig. 2.6 before outlining some possible variations. The basic building blocks, however, are mostly as follows:

Light Source A laser capable of emitting ultra-short pulses (e.g. 50 fs) towards the wall is used to illuminate the scene. Usually, the laser is oriented to multiple points on the wall, which can also be arranged in a grid pattern.

Scene The scene consists of a diffuse relay wall that reflects the emitted laser light to the hidden object and the response thereof to the detector. The object is the target we aim to reconstruct. It is usually placed in front of the relay wall but not directly visible from the laser/detector.

Detector A time-resolved detector measures the incident light received back from the relay wall. Some form of synchronization between the laser and detector is needed, which can be established electronically between the components or via observation of the first reflection of the emitted light pulse on the wall.

There are several variations of the basic setup. Firstly, the arrangement of the laser and detector positions on the wall can be fixed to coincide in a regular grid using a beam splitter resulting in the so-called confocal measurement as introduced by O'Toole et al. [2018]. Several reconstruction methods are specifically tailored to this setup like the (directional) light-cone transform [O'Toole et al., 2018; Young et al., 2020] and f–k migration [Lindell et al., 2019b]. Alternatively, a circular pattern has been proposed, to reduce capture time and data size [Isogawa et al., 2020]. Regarding the choice of the time-resolved detector, the two most prominent choices are streak cameras used by Velten et al. [2012] for the measurements, which work using a time-varying deflection of incoming photons, and Single-Photon Avalanche Diode (SPAD) sensors (see e.g. Charbon et al. [2013] and Buttafava et al. [2015]), which – as the name implies – detect single photons through a high bias voltage. While the former can capture the full histogram over time for a single observed point on the wall in one measurement, the latter relies on repeated measurements that are accumulated into a histogram, but are smaller and cheaper and can hence be more easily arranged into a grid to capture multiple points simultaneously.

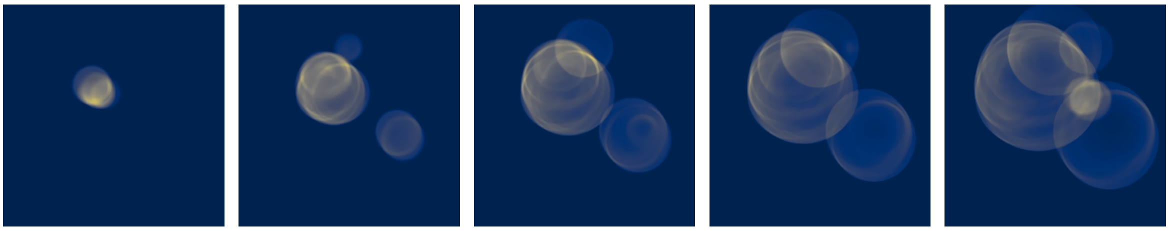


Figure 2.7: Rendered slices of an exemplary transient image, captured in a confocal setup showing the light propagation on the relay wall over time from left to right. The circular patterns can be explained if one considers a single point on the hidden object, which can be seen as a "virtual emitter" of short light pulses. It can only contribute light to points on a sphere for a given delay because of the finite speed of light and intersecting the sphere with the wall plane results in a circle.

2.6.2 Light Transport

While capturing transient images by itself is a technical challenge, further processing of the data is needed to extract interpretable information, since the content of transient images such as the one shown in Fig. 2.7 is not apparent. Towards this end, a model of light, its propagation, and image formation is needed, where the light transport equation (LTE) is a central element in this endeavor. We will briefly introduce the LTE for non-transient imaging following Pharr et al. [2023] and discuss the modifications for transient imaging in this section.

The LTE can be derived from the conservation of energy in a system. On a surface, we describe the radiance leaving a point p in direction ω_o denoted by $L_o(p, \omega_o)$ as

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos\theta_i| d\omega_i. \tag{2.9}$$

Here, $L_e(p, \omega_o)$ denotes the emitted radiance, and the second term describes the scattering of the incoming radiance $L_i(p, \omega_i)$ integrated over all directions ω_i on the hemisphere S^2 with the BRDF $f(p, \omega_o, \omega_i)$. A physically plausible BRDF must satisfy the following conditions.

- $f(p, \omega_o, \omega_i) > 0, \forall p, \omega_o, \omega_i$ (positivity)
- $f(p, \omega_o, \omega_i) = f(p, \omega_i, \omega_o)$ (symmetry/Helmholtz reciprocity)
- $\int_{S^2} f(p, \omega_o, \omega_i) |\cos\theta_o| d\omega_o \leq 1, \forall \omega_i$ (energy conservation)

If we assume that no light is absorbed along the ray between two surfaces, i.e. if no participating media are present, we can replace the term for the incoming radiance L_i with the outgoing radiance on the first surface point $t(p, \omega_i)$ that lies in the direct line of sight from p in direction ω_i, which can be found using ray casting:

$$L_i(p, \omega_i) = L_o(t(p, \omega_i), -\omega_i) \tag{2.10}$$

While an analytic solution is only feasible for the simplest of scenes, Monte Carlo integration with ray tracing can be used to find a solution. An alternative formulation known as the *surface form* or *three-point form* describes the radiance from p' to p using an integral over all surfaces A as

$$L(p' \to p) = L_e(p' \to p) + \int_A f(p'' \to p' \to p) L(p'' \to p') G(p'' \leftrightarrow p') dA(p''). \tag{2.11}$$

In this formulation, $G(p'' \leftrightarrow p')$ is the geometric coupling, which also captures the visibility between the points [Pharr et al., 2023].

This formulation has been adapted by Iseringhausen and Hullin [2020] for NLoS rendering, i.e. the light transport between a single (virtual) light emission and an observed point on the relay wall, by simplifying the model to include only light paths with a single interaction with the hidden object (three bounce assumption) and temporally distributing the irradiance contributed by each triangle using a triangular filter.

Another alternative description of the LTE is the *path integral* formulation [Veach, 1998] which describes the measurement with an integration over paths $\bar{x}$

$$I_j = \int_\Omega f_j(\bar{x}) d\mu(\bar{x}). \tag{2.12}$$

Here, Ω is the space of all paths of all lengths and f_j the measurement contribution function. For transient rendering, the finite speed of light needs to be integrated into the model as shown by Jarabo et al. [2014]:

$$I_j = \int_\Omega \int_{\Delta T} f_j(\bar{x}, \bar{\Delta t}) d\mu(\bar{\Delta t}) d\mu(\bar{x}). \tag{2.13}$$

The path contribution function f_j is extended to make emission, throughput, and the sensor importance depend on time. The additional sequence of time delays $\bar{\Delta t}$ makes a naive numerical integration even more difficult and the ultra-short light pulses needed to produce interesting transient images are close to a delta manifold, which renders random sampling futile. Note that the time delays here are results of the scattering and not propagation delays, which correlate to the length of the traveled path $\bar{x}$. Jarabo et al. [2014] show how this can be solved efficiently by reusing sampled paths and introducing special sampling strategies.

Inverse Rendering Based on the aforementioned model it is possible to synthesize images from a scene description using techniques such as ray tracing. This rendering process is a forward model and solving the inverse problem of reconstructing the scene or some parameters thereof from observations is known as inverse rendering. This is usually done in an iterative fashion using gradient descent type optimization. As the rendering function is quite complex and not even differentiable everywhere, a simple choice to compute gradients is the use of finite differences, i.e. calculating the gradient with respect to each parameter by applying two evaluations of the loss with slight variation in the parameter. While simple to implement, this technique is quite inefficient when many parameters need to be optimized. However, for certain representations like the Gaussian blobs used by Iseringhausen and Hullin [2020] and for an efficient implementation of the rendering function this can work reasonably fast. As an alternative, automatic differentiation techniques can be applied to compute the gradients by repeated application of the chain rule. This can be done either in forward mode by propagating a seed for each parameter towards the full equation, or – more commonly – in backward mode by backpropagating the gradient of the loss towards the parameters. We present a fast differentiable NLoS renderer using backpropagation in Chapter 6.

One peculiarity of the rendering equation is the visibility term, which is a binary decision function and as such not differentiable at the visibility boundary. This poses a problem for inverse rendering as it does not allow the propagation of gradients to the underlying geometry positions. While a thorough treatment is beyond the scope of this work, various approaches to enable the computation of meaningful derivatives have been proposed. We refer to the survey of Kato et al. [2020] for more details and references to recent works.

Chapter 3

Related Work

In line with the three problem areas included in this thesis, we outline the related work for each of them as follows. First, we provide an overview of stereo estimation methods in Section 3.1, followed by frame interpolation works in Section 3.2, before presenting a review of NLoS reconstruction methods in Section 3.3.

3.1 Stereo Matching

We will briefly outline the taxonomy of pre-deep learning stereo matching methods in Section 3.1.1 before giving an overview of network architectures in Section 3.1.2. As the base architecture for our work (Chapter 4), we provide more details on recurrent networks in Section 3.1.3. For works on the closely related problem of multi-view stereo matching, we refer to the survey of Stathopoulou and Remondino [2023] as it is beyond the scope of this thesis.

3.1.1 Taxonomy

To give an overview of classical stereo matching approaches, we will follow the taxonomy proposed by Scharstein and Szeliski [2002], which was also the base for the more recent survey of Hamzah and Ibrahim [2016]. In essence, all methods consist of the following four steps or a subset thereof:

Matching Cost Computation For each pixel in the image, the matching cost for each offset or disparity indicates how likely the matching point in the other image which lies on the epipolar line (see Section 2.3.1) corresponds to the same observation, where the underlying assumption of most approaches is that the same point viewed from the two different cameras of a stereo rig will be photo-consistent to some degree. Scharstein and Szeliski [2002] identified squared intensity differences and absolute intensity differences as two common choices among many possible approaches. Starting around 2010, feature-based matching costs were proposed [Hamzah and Ibrahim, 2016], which can be seen as predecessors to the learned feature representations of learning-based approaches.

Cost Aggregation As the matching cost between single pixels can be prone to noise and potentially uninformative, aggregation is a crucial step to ensure accurate matches, especially for methods that only consider local information. The underlying idea of many methods is the use of moving windows, which often can be realized through convolutions with an appropriate kernel [Scharstein and Szeliski, 2002].

Disparity Computation/Optimization To extract disparity from the cost information, methods can be classified into *local* and *global* algorithms. *Local* methods simply select the best match for each pixel as the one associated with the lowest cost in a winner-takes-all scheme. *Global* methods additionally aim at incorporating smoothness and/or consistency and typically solve a regularized optimization problem based on the given cost [Scharstein and Szeliski, 2002].

Disparity Refinement The aim of this last step depends on the target application and can include sub-pixel refinement for e.g. image-based rendering, but also handling of occlusions and mismatches [Scharstein and Szeliski, 2002], where smoothing of the disparity map is prominently done via Gaussian filters or a diffusion process [Hamzah and Ibrahim, 2016].

3.1.2 Deep Learning

The earliest works on neural networks for stereo matching were proposed by Zagoruyko and Komodakis [2015] and Zbontar and LeCun [2015], where the goal was to train a convolutional neural network to predict how well two image patches match, which can be seen as a learned matching cost computation and aggregation. Zbontar and LeCun [2015] extended this learned local approach with global optimization and subpixel refinement in a more traditional manner. Luo et al. [2016] and Mayer et al. [2016] proposed to compute the matching cost as the inner product between learned feature vectors. This approach of computing correlation or cost volumes has been the foundation of many deep learning approaches since and was originally proposed for frame interpolation by Dosovitskiy et al. [2015]. While being computationally efficient, such a technique significantly reduces the amount of information. Concatenation volumes are an alternative, where the features of the reference image and the target image are concatenated for each volume, retaining more information for the subsequent 3D-convolutional neural network (CNN) filtering that produces the cost volume as proposed by Kendall et al. [2017]. Later, an intermediate approach was presented by Guo et al. [2019b], who split the correlation computation into groups, retaining more information while still reducing the latent space. As a hybrid approach, it has been used extensively since (e.g. [Chabra et al., 2019; Nie et al., 2019; Li et al., 2022; Abd Gani et al., 2024]).

To add further context information to the cost volume processing, Zhou et al. [2017] proposed an additional image feature extraction module. For a better handling of the global context Chang and Chen [2018] proposed spatial pyramid pooling before the const volume construction and a stacked hourglass network for the processing of the cost volume. A different pyramid-based approach was presented by Tonioni et al. [2019] targeting increased speed of the network. Their work was also inspired by optical flow networks and computes the disparity in a coarse-to-fine fashion and avoids large correlation volumes by refining the rough estimate of the previous layer within a local window only. For multi-view stereo, a similar hierarchical approach was proposed by Gu et al. [2020]. Shen et al. [2021b] have shown fused cost volumes for better extraction of matching information from the low-resolution data in the hierarchy.

Besides the different treatments of the cost volume, other approaches have been proposed. Seki and Pollefeys [2017] demonstrated a network that builds on the idea of semi-global matching, and Wang et al. [2021b] integrated ideas from Patchmatch [Barnes et al., 2009] into the network. Xu et al. [2022] proposed to integrate an attention mechanism for filtering the concatenation volume and the suppression of irrelevant features before aggregation. Attention-based feature extractions were also proposed by Li et al. [2021] and Xu et al. [2023b]. Such an approach enables the exchange of features between the left and right image branches of the network already prior to the cost volume computation. Inspired by residual architectures, Pang et al. [2017] proposed to split the network into two stages, where the first outputs an initial disparity, and the second stage predicts an offset disparity that is added to the initial one to get the final output. This approach can be seen as a predecessor of the recurrent architectures discussed in the next section.

3.1.3 Recurrent Architectures

Conceptually similar to classical optimization-based methods, recurrent stereo networks refine an initial disparity iteratively by predicting offsets and updating a hidden state. This approach was initially proposed for optical flow estimation as Recurrent All-Pairs Field Transforms (RAFT) by Teed and Deng [2020] and adapted for stereo regression by Lipson et al. [2021] and works as follows. After the extraction of feature maps for both images through a convolutional network, a correlation volume is computed similar to previous works using the inner product, from which a hierarchy of volumes is computed using pooling. At the start of each iteration of the following recurrent network, the correlation values in a window around the current disparity estimate are bilinearly sampled from the volume, encoded, and passed to a GRU-based network, which updates a hidden state. From this new hidden state, a delta is predicted to update the current disparity estimate.

Several extensions to this approach have been proposed tackling initialization, iterative updates, and disparity refinement. Wang et al. [2022] applied the idea in a multi-view stereo problem and extracted depth probabilities along with a confidence measure from this hidden state instead of a single depth value. The depth is computed as the local expected value of this distribution around the highest probability. Zhao et al. [2022] replaced the GRU modules of the iterative updates with an LSTM network and refined the disparity prediction using an error-aware hourglass network. Another modification to the iterative updates was proposed by Zhao et al. [2023], who also use LSTM modules and decouple the disparity map update from the hidden state. They further proposed normalization for an improved disparity refinement and an attention mechanism for feature extraction. Xu et al. [2023a] extended the correlation volume and built a geometry encoding volume to regress the initial disparity and retain more feature information. Finally, Li et al. [2022] have shown a network design that enables running the recurrent updates in a coarse-to-fine scheme across three resolutions to improve efficiency. This aspect is also targeted by our work (Chapter 4) where we replace the dense correlation volume with a sparse approach.

3.2 Frame Interpolation

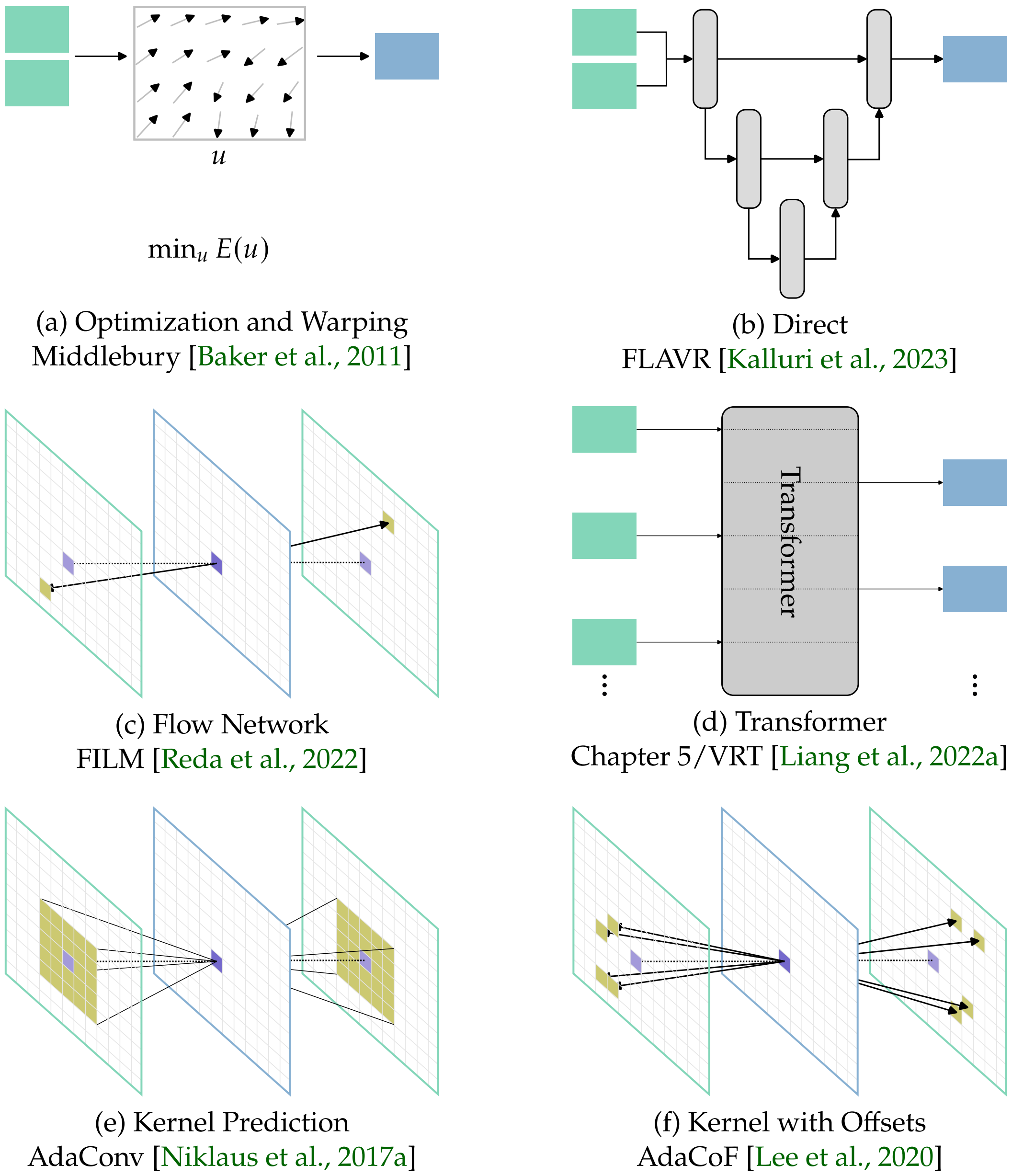
Video frame interpolation is another long-standing problem in computer vision, that is traditionally solved by splitting the problem into an optical flow estimation followed by the actual interpolation. We discuss those methods in Section 3.2.1 and refer to Section 2.4 for more details on optical flow and image warping. Those methods have been surpassed by learning-based methods which can be roughly categorized as direct (Section 3.2.2), kernel (Section 3.2.3), and motion-based methods (Section 3.2.4). See Fig. 3.1 for a visual summary of the existing classes of approaches. We conclude this section with an overview of frame interpolation benchmarks and metrics in Section 3.2.5 refer to the survey by Dong et al. [2023] for a more exhaustive treatment.

3.2.1 Classical Methods

Early frame interpolation methods typically relied on optical flow vectors, which describe the movement of points between images and as such are applicable both to view interpolation, where both views are assumed to be recorded at the same time by different cameras, and video frame interpolation, where the same – but potentially moving – camera observes a scene at different times. Finding such correspondences between views goes back to the 1980s with influential works such as Horn and Schunck [1981] and Lucas and Kanade [1981].

Once the optical flow is known, or estimated with sufficiently high quality, occlusion between frames is a central problem for frame interpolation methods, as there are pixels for which no match in the other frame exists. Herbst et al. [2009] studied those occlusion effects and presented a frame interpolation method that explicitly integrates depth reasoning into the algorithm. Another approach was presented a few years earlier by Zitnick et al. [2004] for view interpolation, which used a two-layer representation. Based on those previous works, Baker et al. [2011] included a simple, albeit efficient, algorithm for frame interpolation from optical flow vectors as a baseline model in their evaluation suite.

At the same time, Werlberger et al. [2011] proposed a more complex solution by adapting a TV-L_1-based denoising algorithm for frame interpolation using the precomputed optical



(a) Optimization and Warping
Middlebury [Baker et al., 2011]

(b) Direct
FLAVR [Kalluri et al., 2023]

(c) Flow Network
FILM [Reda et al., 2022]

(d) Transformer
Chapter 5/VRT [Liang et al., 2022a]

(e) Kernel Prediction
AdaConv [Niklaus et al., 2017a]

(f) Kernel with Offsets
AdaCoF [Lee et al., 2020]

Figure 3.1: Overview of existing frame interpolation methods, both optimization-based traditional approaches (a) and neural networks (b-f). The latter can mostly be distinguished by their approach to motion compensation: While direct methods (b) do not handle motion explicitly, optical flow (c) and kernel (e) approaches, as well as hybrid methods (f), use custom modules to improve feature propagation. Lastly, transformer methods (d) aim at treating arbitrary sequences of frames instead of just image pairs.

flow. To this end, they aim to solve the following objective,

$$\min_u \int_{\Omega \times T} |\nabla_v u| + \lambda(x,t) \| u - f| \, dx \, dt, \tag{3.1}$$

where $|\nabla_v u|$ acts as a spatial and temporal regularization that includes the optical flow v to temporally align the images, and the second term is the weighted data term. After discretization, it is solved using the primal-dual algorithm by Chambolle and Pock [2011]. Rakêt et al. [2012] presented an approach based on reparametrizing the optical flow to the intermediate frame and solved it using a bottom-up architecture. Other works have modeled the deformation between frames using homographies of decomposed regions [Stich et al., 2008] or posed it as an optimal control problem [Chen and Lorenz, 2011].

As an alternative to optical flow-based methods Meyer et al. [2015] presented a phase-based method, which is built on the idea that some motions between frames can be represented as a phase shift. They used a multi-scale pyramid and propose a bounded shift correction based on the intuition that the movement of content between frames is similar for low and high frequencies. Later, they presented an extension of their method [Meyer et al., 2018b], that introduced supervised learning into their pipeline.

3.2.2 Direct Methods

Similar to most computer vision tasks, deep learning methods have demonstrated exceedingly good performance for frame interpolation. In their seminal work Long et al. [2016] have shown that training a CNN for frame interpolation based on videos alone can also solve the image matching problem. Their architecture is based on the work of Dosovitskiy et al. [2015], which introduced the first CNN-based optical flow method, and as such does not include any motion compensation, as used by most later works presented in the following sections. Later, Choi et al. [2020] introduced channel attention blocks in their method dubbed CAIN, which is a deep convolutional network containing residual blocks operating on $\frac{1}{8}$ resolution through PixelShuffle [Shi et al., 2016]. Another method is FLAVR [Kalluri et al., 2023], which adapts a classic 3D U-Net architecture for frame interpolation and hence can handle arbitrary temporal context windows compared to the two neighboring keyframes used by previous works. Despite their success, a common problem of those direct methods is that they struggle with large motion, as convolutional networks are restricted in their receptive field and thus cannot easily propagate information across the image. To improve the feature propagation, a transformer-based architecture with convolutional layers was proposed by Liu et al. [2020b], where all input frames are treated as tokens in the encoding and the target frames are added as tokens in the decoding which are transformed into RGB in a final U-Net type network.

3.2.3 Kernel-Based Methods

Kernel-based methods aim to improve the propagation of features from keyframes to the target frame. They were originally introduced by Niklaus et al. [2017a] in their method

AdaConv (Adaptive Convolution) with the goal of not relying on the quality of the optical flow estimate. In their method, a CNN predicts kernel weights for each image point that are used to convolve patches from the input frames around those points. However, the kernels need to have a sufficiently large size to handle large motions and in their work they predict kernels of size 41 × 41, which limits the the sampling to 20 pixels in horizontal or vertical direction at a relatively high memory demand, as the kernels alone require 26 GB for the interpolation of a 1080p frame.

Various follow-up works have proposed different ideas to tackle this challenge. Foremost, Niklaus et al. [2017b] presented an extension of their method named SepConv that separates the large 2D kernel into two 1D kernels and hence reduces the memory requirement from $n \times n$ to $2n$. This technique was also used in their later work [Niklaus et al., 2021], where they demonstrated that a variety of network and pipeline improvements of their method can improve the results significantly to match the state of the art. As another optimization, hybrid approaches between kernel- and motion-based methods have been proposed, that add an offset prediction to efficiently handle large motion [Peleg et al., 2019; Cheng and Chen, 2020; Lee et al., 2020], which can be seen as an adaption of deformable convolution networks of Dai et al. [2017] for frame interpolation. Those architectures were adapted and improved by many subsequent works. Cheng and Chen [2021] proposed an extension with an additional bias estimator and added the temporal index as input to generate intermediate frames at arbitrary times and Chen et al. [2021] included a coarse-to-fine offset estimation before the deformable-convolution-based warping/blending. Another approach that splits the interpolation into more stages was presented by Gui et al. [2020] who proposed a structure-guided interpolation step based on deformable convolutions followed by a texture refinement step based on EDVR [Wang et al., 2019]. Some researchers have proposed approaches that – among other extensions – aim to improve the feature extraction of the model, either using 3D convolutions [Danier et al., 2022a] or a transformer-based architecture [Shi et al., 2022]. Liang et al. [2022a] also proposed to use transformer blocks in their video processing network that can handle tasks like super-resolution and deblurring and can also be configured to perform frame interpolation. Their method uses deformable convolutions to align features from different frames and combine them using an attention mechanism. Their follow-up work is conceptually more similar to the approach presented in Chapter 5 but does not work for frame interpolation [Liang et al., 2022b]. More recently, Zhou et al. [2023b] proposed an approach to improve the visual quality by using a cross-scale alignment of features and a texture consistency loss in their training pipeline, which uses the census transform to find the best match in the input frames and adds another L_1 loss between the prediction and the matched point.

3.2.4 Motion-Based Methods

Inspired by classical methods that rely on optical flow estimates and warping, various deep learning methods have been proposed. They can be roughly categorized by the approach they take to estimate and handle the optical flow. The two most prominent methodologies either apply a forward warping by rescaling the flow vectors between the input frames [Niklaus and Liu, 2018; Niklaus and Liu, 2020; Hu et al., 2022] or estimate the flow vectors from the intermediate frame to the keyframes and apply a backward warping [Park et al., 2020; Huang et al., 2021; Park et al., 2021; Kong et al., 2022; Reda et al., 2022]. As an alternative to the direct regression of the intermediate flow field, approaches to compute it from the flows of the keyframes have been proposed. Jiang et al. [2018] sampled and rescaled the forward and backward flow vectors at the same positions assuming a smooth flow field and trained a network to reduce artifacts around boundaries. Similar to classical interpolation methods Bao et al. [2019a] proposed to add estimated depth information for an improved resampling of the optical flow using forward warping and fill the gaps by averaging the valid neighbors. In a similar approach, Niklaus et al. [2023] proposed to adapt their softmax splatting interpolation method [Niklaus and Liu, 2020] to compute the intermediate flow. More elaborate sampling strategies combining various warping techniques and weighting/correlation have been demonstrated by Sim et al. [2021], Lee et al. [2022], and Danier et al. [2022b].

One central assumption of most models is that the motion remains linear within the observed time frame, which greatly simplifies rescaling and warping operations as well as the prediction itself at the cost of accuracy. However, the prediction of non-linear flows from just two input frames is ill-posed, which is why some researchers have proposed models that operate on three or more input frames to accurately predict non-linear motion [Xu et al., 2019; Liu et al., 2020a; Choi et al., 2021; Dutta et al., 2022; Liu et al., 2022c]. Alternatively, approaches to predict non-linear motion using a learned prior from two input frames have been proposed [Park et al., 2021; Liu et al., 2022b].

While the motion compensation techniques significantly improve the propagation of features across the frames, large motion still poses a major challenge to many approaches. This is – aside from the size of the search space – also an issue with the distribution of motion vectors in the training data as shown by Reda et al. [2022], who also proposed a sampling strategy to remove this prior from the training data and improve interpolation of large motions. Alternative strategies include an adaptive hierarchy of the network architecture which also enables high-resolution interpolation [Sim et al., 2021], or a two-step approach which first narrows the temporal gap and subsequently performs another interpolation step [Argaw and Kweon, 2022].

Several methods targeting more specific application scenarios than the general frame interpolation have been proposed for improved performance. Siyao et al. [2021] presented a method tailored to animated video content, where the interplay between sharp lines and uniform colors needs to be faithfully reconstructed for a visually pleasing interpolation and exaggerated/non-linear motions pose another challenge to the method. They used the segmentation of the inputs to improve flow computation on untextured regions and an iterative refinement to tackle those challenges. Targeting line art interpolation, an approach based on interpolating graphs extracted from the input images was presented by Siyao et al. [2023] with a transformer network for the features of the vertices and a visibility prediction network to handle occlusion. For 3D rendering applications, it is possible to use additional feature maps such as albedo, normal, or depth produced by the renderer for the interpolation as demonstrated by Briedis et al., 2021. Their approach utilizes the feature maps of the target frame, which can be computed significantly faster than the rendered images to improve the matching between frames and the compositing. In a following publication by Briedis et al. [2023] they have shown interpolations with a kernel prediction for robustness that also enables the interpolation of arbitrary additional content such as alpha masks. In addition, they have shown an adaptive interpolation of longer sequences with an error prediction that is conceptually similar to the work presented in Chapter 5.

Various other extensions and improvements have been proposed. Chi et al. [2022] split the interpolation computation into three subnetworks based on the difficulty measured as the estimated error from the optical flow. Liu et al. [2022a] used an attention mechanism for frame fusion on multiple hierarchy levels in parallel and proposed an improved sampling to overcome problems from imperfect flow predictions. More recent advances include inter-frame attention [Zhang et al., 2023], bidirectional-flow estimation with novel pyramid architectures [Jin et al., 2023a; Jin et al., 2023b], and improved correlation handling [Zhou et al., 2023a].

3.2.5 Frame-Interpolation Evaluation

For a fair comparison of the aforementioned methods, common benchmarks, and training/evaluation procedures are necessary. We will provide a short overview of the most commonly used datasets and metrics in this section.

Vimeo-90k [Xue et al., 2019] is one of the most commonly used datasets for training and testing not only frame interpolation but also other video processing networks, containing a total of 73,171 triplets and 91,701 septuplets at resolution 448×256 extracted from 89,800 video clips. For testing, popular choices are Middlebury [Baker et al., 2011], which was the first dataset for this purpose, UCF101 [Soomro et al., 2012], SNU-Film [Choi et al., 2020], which is split into four difficulty levels from *easy* to *extreme*, and X4K1000FPS or X-TEST [Sim et al., 2021], which puts a focus on high resolution and large motion. In addition, HD [Bao et al., 2019b] and Adobe240fps [Su et al., 2017] have been used.

To report the quality of the interpolated frame it is common to measure peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and the perceptual LPIPS [Zhang et al., 2018]. Note, that those metrics only evaluate the quality of the interpolated frame compared to the ground truth, and not the quality of the resulting video as a whole, where different effects like flickering blurring, or missing motion can have an immense impact on the perceived quality and are not accurately covered by the above measures. To alleviate this to some extent, measures like PFIQM [Yang et al., 2008] and VFIPS [Hou et al., 2022] have been proposed, which aim at computing a quality score/perceptual similarity between videos. Another benchmark was recently presented by Kiefhaber et al. [2024], to which we refer for further insights into frame interpolation evaluation.

3.3 Non-Line-of-Sight Reconstruction

Many different approaches to reconstruct the hidden scene from transient measurement have been proposed, differing in the object representations and their faithfulness to the underlying physical model, and most importantly the underlying measurement setup. We will give an overview of the current literature, grouping those methods into backprojection and transformation (Section 3.3.1) methods which typically represent the hidden scene as an albedo volume, rendering, and optimization-based reconstructions (Section 3.3.2), and all methods using (supervised) learning (Section 3.3.3). We close this section with a brief summary of NLoS reconstruction approaches from different measurement modalities (Section 3.3.4). For more details on the methods prior to 2020, we refer to Faccio et al. [2020].

3.3.1 Backprojection and Transformations

The transient measurement after a certain delay corresponds directly to the travel time and hence distance of the light. Assuming a single reflection on the surface of the hidden object, we know that the reflection must have happened somewhere on an ellipsoid, where the laser position on the wall and the observed point are its foci. Back-projection methods were initially introduced by Velten et al. [2012] and exploit this effect by projecting the measured light into a heat map represented by a voxelized volume of the hidden scene. To reveal the hidden shape, a filtering step using the second derivative is needed to process the heat map. Subsequently, Arellano et al. [2017] presented a faster version of the back-projection that formulates the problem as a voxelization of ellipsoids and uses GPU acceleration for a fast implementation. One major advantage of those methods is that they pose relatively few restrictions on the measurement setup: The laser/observation points do not need to lie in a regular grid or even on a planar wall, only their positions must be known. On the other hand, restricting the measurement setup enables other algorithms like the light-cone transform (LCT) presented by O'Toole et al., 2018, which is tailored for a confocal setup, i.e. a sequentially sampled regular grid of measurement points on the relay wall where each point is both observed and illuminated. For this case, they derive a closed-form solution by writing the problem as a 3D convolution using a change of variables and solving the deconvolution efficiently in the Fourier domain. This approach was extended by Young et al., 2020 to reconstruct surface normals along with the volumetric albedo. Specifically for the application to long-range NLoS reconstruction, Wu et al. [2021a] presented an iterative 3D deconvolution method. A different angle to approach to problem was presented by Lindell et al. [2019b] by modeling the light propagation using the wave equation. The reconstruction is achieved by solving a boundary value problem where the observation for one spatial slice at all times is given and needs to be migrated to the virtual emission at time 0 in space that is the object surface. This is realized using the f-k migration and they also demonstrate a pre-processing step that makes it applicable for certain non-confocal measurements. Another wave-based approach was presented by Liu et al. [2019b] using a virtual light field model and posing the problem in a way that is equal to solving the Rayleigh-Sommerfeld diffraction integral. This method was later adapted by Nam et al. [2021] to work in real time through a remapping of measurements from custom SPAD arrays. Xin et al. [2019] present a theory of Fermat paths that allows to identify scene geometry from discontinuities in the measurements which can be projected using the spatial derivatives. It is noteworthy that the forward model with the albedo volume is conceptually similar to that of computed tomography scanning, as described by Gupta et al. [2012], where the integration is done over ellipsoids instead of lines and the output has a higher dimensionality.

3.3.2 Optimization and Rendering

The previous approaches mostly exploit simplified models of the image formation that are invertible to some extent to find an approximate solution. To capture more complex effects for a better reconstruction, the image formation can be modelled in a more physically accurate or plausible manner to find a solution via optimization of the scene parameters. Pediredla et al. [2017] proposed a reconstruction of the walls of a room from transient measurements using a dictionary-based optimization approach. For a volumetric reconstruction of the full hidden scene, Heide et al. [2019] solve the problem via an alternating least-squares optimization of a factorization that includes visibility and normals along with the time-dependent transport matrix modeling the time-of-flight. The first rendering-based approach was presented by Tsai et al. [2019]. They implemented a transient renderer using the three bounce assumption and compute the image and derivative integrals using Monte Carlo integration. They optimize a triangle mesh representation in a coarse-to-fine manner and apply re-meshing to avoid degradations of the surface structure. An alternative analysis-by-synthesis reconstruction approach was presented by Iseringhausen and Hullin [2020], where they use central differences to optimize Gaussian blobs, which are meshed using marching cubes and efficiently rendered through a GPU implementation including temporal filtering specifically tailored to the three bounce NLoS reconstruction case. Since the optimization is relatively slow, we present a more efficient implementation based on gradient backpropagation in Chapter 6. The restriction to three bounces was lifted by more general renderers as presented by Yi et al. [2021], who demonstrate tracking of an object around two corners, and Wu et al. [2021b], who also show the optimization of a hidden shape. An alternative research direction aims at finding scene representations better suited for inverse rendering approaches. Inspired by advances in novel view synthesis, Shen et al. [2021a] proposed a neural transient field that replaces voxel volume albedo representations by a neural network. Two other methods using implicit scene parameterizations were presented by Choi et al. [2023] and Fujimura et al. [2023].

3.3.3 Learned Reconstruction

Similar to the other problem domains, learning-based methods have been introduced to solve the NLoS reconstruction task. Grau Chopite et al. [2020] use a 3D U-Net architecture to process the transient measurements, followed by an upsampling and regression step to regress depth maps and train their method on synthetically rendered transient images. Chen et al. [2020] propose to predict a volumetric feature representation of the hidden scene by extracting feature vectors from the transient image, which are propagated into the 3D scene using f-k migration and transformed with another embedding block before further task-specific processing. While other (differentiable) methods like filtered back-projection or LCT can be used for the propagation, f-k migration yielded the best results. The method is trained end-to-end for 2D image and depth rendering, RGB-D reconstruction, classification, and object detection. The importance of the underlying model-based propagation is also highlighted by Mu et al. [2022], who demonstrate a similar approach but for non-confocal measurements using a vectorized version of the Rayleigh-Sommerfeld diffraction of Nam et al. [2021]. As an alternative to the feature volume representations described above, Grau et al. [2022] propose to train a neural network to predict occlusion fields motivated by the observation, that some parts of the hidden object's surface are not visible from any point on the relay wall due to occlusion, making their recovery from the measurement impossible. Their implementation is based on occupancy networks for shape reconstruction [Mescheder et al., 2019; Peng et al., 2020] and trained on synthetic data with a binary-cross-entropy loss on sampled points. More recently, Li et al. [2023] have demonstrated improved results for NLoS RGB-D reconstruction from a transformer-based architecture for the feature processing after the feature propagation using f-k migration as described above.

3.3.4 Miscellaneous Modalities

Aside from the hardware specifications and the arrangement of the laser and observation points on the relay wall, the previous methods all worked with data captured in a similar methodology. In this section, we will briefly outline alternative capture modalities and processing methods for NLoS reconstruction. As capturing a transient image requires expensive hardware Heide et al. [2013] propose to capture data from photonic mixer devices (PMDs) and reconstruct a transient image solving the inverse problem with several regularization terms and an alternating optimization. This work was extended to reconstruct the hidden geometry from PMD measurements [Heide et al., 2014]. Another phase-modulation-based approach was presented by Kadambi et al. [2016], who apply beamforming for the localization of the hidden target. Reducing the capturing hardware requirements even further Klein et al. [2016] propose to track hidden objects from steady-state images of the wall with an additional coherent laser illumination solving a non-linear optimization problem using numerical derivatives of the forward model. Smith et al. [2018] present tracking from speckle images exploiting the movement of the hidden objects in the scene. Finally, Lindell et al. [2019a] have shown NLoS imaging from audio signals captured using an array of speakers and microphones and reconstructed through the LCT.

Part II

Publications

Chapter 4

VHS: High-Resolution Iterative Stereo Matching with Visual Hull Priors

In this chapter, we discuss the contributions and results developed in the following publication which already appeared as a preprint.

Markus Plack, Hannah Dröge, Leif Van Holland, and Matthias B. Hullin.
"VHS: High-Resolution Iterative Stereo Matching with Visual Hull Priors."
arXiv preprint arXiv:2406.02552, 2024.
DOI: 10.48550/arXiv.2406.02552

In the following, we include a verbatim copy of the content of this work subject to some minor editorial changes.

Author Contributions of the Publication I realized the network based on the implementation of Xu et al. [2023a] and developed the additional functions for efficient correlation. The visual hull code was provided by Patrick Stotko and all remaining scripts were implemented by me. In addition, I executed the training and evaluation of the models based on discussions with my co-authors. The schematic figures were provided by Hannah Dröge and Leif Van Holland, who also contributed greatly to the whole text.

Abstract

We present a stereo-matching method for depth estimation from high-resolution images using visual hulls as priors, and a memory-efficient technique for the correlation computation. Our method uses object masks extracted from supplementary views of the scene to guide the disparity estimation, effectively reducing the search space for matches. This approach is specifically tailored to stereo rigs in volumetric capture systems, where an accurate depth

plays a key role in the downstream reconstruction task. To enable training and regression at high resolutions targeted by recent systems, our approach extends a sparse correlation computation into a hybrid sparse-dense scheme suitable for application in leading recurrent network architectures.

We evaluate the performance-efficiency trade-off of our method compared to state-of-the-art methods, and demonstrate the efficacy of the visual hull guidance. In addition, we propose a training scheme for a further reduction of memory requirements during optimization, facilitating training on high-resolution data.

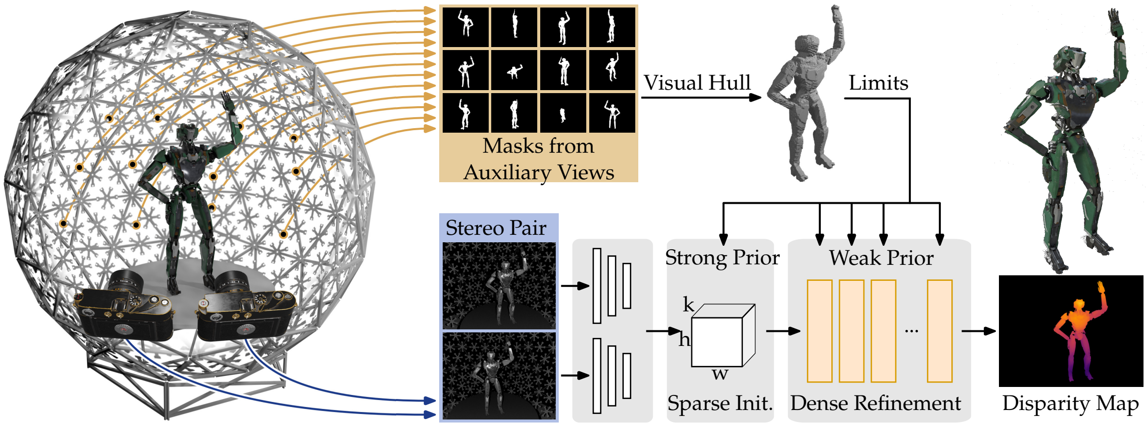


Figure 4.1: We propose a technique to induce a rough shape estimate from object masks (top) as prior information to a novel, sparse-dense stereo-matching network (bottom) for the application in capture stages (left) for accurate and memory-efficient disparity estimation (right).

4.1 Introduction

Stereo matching is a long-standing problem in the area of computer vision, driving core functionality in a wide range of applications, for example in the automotive industry, virtual and augmented reality systems, as well as in medical imaging, agriculture, remote sensing, and robotics domains. Recently, interest surged in telepresence and virtual production scenarios that use volumetric capturing systems [Collet et al., 2015; Orts-Escolano et al., 2016; Guo et al., 2019a; Heagerty et al., 2024], which rely on fast and accurate depth estimates for downstream reconstruction tasks. The disparity regression problem is typically solved by initially computing the matching cost between a stereo image pair or a suitable feature representation thereof and searching for the best correspondences along the epipolar lines resulting in a highly irregular cost landscape. Challenges include occlusion, view-dependent reflectivity, repetitive patterns, and insufficient calibration accuracy. With the rise of deep learning in the domain of computer vision, classical matching methods [Barnard and Thompson, 1980; Mühlmann et al., 2002; Scharstein and Szeliski, 2003; Hamzah and Ibrahim, 2016] are surpassed by data-driven approaches [Mayer et al., 2016; Kendall et al., 2017; Guo et al., 2019b; Xu and Zhang, 2020]. Recently, so-called all-pairs-correlation networks based on the optical flow network RAFT [Teed and Deng, 2020] have shown to perform remarkably well when applied in the stereo matching context [Lipson et al., 2021]. Those methods compute a dense correlation volume for *all* possible matches and perform stereo regression in an iterative fashion akin to gradient descent methods. One distinct drawback of such approaches is that the size of the full correlation volume scales quadratically with the horizontal input resolution, limiting their applicability on high-resolution inputs. One solution to reduce the prohibitive memory requirement is to use sparse representations [Wang et al., 2021c] that only store the k most relevant entries of the correlation volume, similar to k-nearest-neighbor (kNN) methods. While this still requires the computation of *all* correlation values, which does not reduce the computational costs, the memory demand only scale linearly with respect to the horizontal input resolution, but possibly discards valuable information.

In contrast, we propose a sparse-dense approach that allows us to consider all disparities, avoiding the limitations associated with missing values in sparse representations. We calculate disparities using a sparse method initially, followed by a refinement in a memory-efficient dense manner. As a crucial step to reduce the amount of sparse candidates, we propose to employ the visual hull [Laurentini, 1994] as a rough shape estimate that reduces the set of valid disparities to points inside the hull. The foreground segmentation masks required for this are available through the use of chroma-keying [Raditya et al., 2021] or more sophisticated image-level segmentation approaches [Guo et al., 2019a] in many capturing scenarios and thus the visual hull can be computed easily. During the refinement step, we can further use the hull as a weak prior.

In summary, our contributions are as follows:

- We present a method to induce prior knowledge of visual hulls from auxiliary views into a recurrent stereo-matching network to reduce the initial disparity search space and as guidance for the iterative refinement.
- We demonstrate a sparse-dense correlation method that effectively reduces peak memory requirements while retaining the accuracy of all-pairs correlation methods through just-in-time computation for the updates.
- We propose an optimization scheme to realize high-resolution training of recurrent stereo network architectures and show how the visual hull-guided network can benefit from pre-training on conventional training data by making the input optional.

We share the model and training implementation of our **V**isual **H**ull **S**tereo (*VHS*) network and the custom kernels along with the data used for training and testing at `https://github.com/unlikelymaths/vhs`.

4.2 Related Work

Learning-based methods using correlation volumes to predict accurate disparity maps have shown great potential in stereo matching. We briefly review approaches for generating cost volumes and discuss previous work on further refinement of the disparities by iterative update methods before giving an overview of stereo vision approaches targeting efficiency aspects.

4.2.1 Matching Cost Volume

Recent developments in end-to-end learning approaches for cost volumes have successfully captured the similarity of pixel pairs across varied degrees of disparity in stereo matching [Mayer et al., 2016; Kendall et al., 2017; Zhang et al., 2019; Gu et al., 2020].

In this context, Mayer et al. [2016] introduced a method based on *correlation* for calculating cost volume, followed by subsequent work [Liang et al., 2018; Tonioni et al., 2019]. This approach measures the correlation between the features of two images within a 1D correlation layer applied horizontally along the disparity line.

Concatenation-based methods [Chabra et al., 2019; Nie et al., 2019; Li et al., 2022; Abd Gani et al., 2024], on the other hand, follow a different strategy. Kendall et al. [2017] concatenated unary features with their corresponding features along the disparity line. They generated a 4D cost volume, subsequently processed through an encode-decoder network with 3D convolutions across spatial dimensions and disparity. To further regularize the 4D cost volume, Chang and Chen [2018] discussed the implementation of a learned regularization using a stacked hourglass network. Addressing the lack of explicit similarity measures in previous concatenation-based approaches, Guo et al. [2019b] proposed integrating group-wise correlations into the 4D cost volume by dividing features into sub-groups and calculating correlations for each. To improve the performance even in regions with less texture, recent work [Xu et al., 2022] filters the concatenation volume with attention weights to suppress unnecessary information.

To overcome storage and runtime limitations, *cascading* cost volumes were created by building a cost volume pyramid and progressively refining depth estimation with a coarse-to-fine technique [Gu et al., 2020]. Other cascade formulations have been proposed for even higher resolutions [Wang et al., 2021b] or address unbalanced disparity distributions [Shen et al., 2021b].

4.2.2 Iterative Updates in Stereo Matching

Initially proposed for optical flow estimation, deep learning approaches have successfully employed traditional optimization methods using learned updates to improve performance. These methods refine disparity maps through successive updates, as demonstrated by RAFT (Recurrent All-Pairs Field Transforms) [Teed and Deng, 2020]. RAFT consists of a feature encoding step, computation of correlation volumes containing the correlations between all pixel pairs, and a learned update operator that iteratively updates the optical flow estimation based on the correlation volumes. Based on this, Lipson et al. [2021] introduced an adaptation of RAFT for stereo disparity estimation, called RAFT-Stereo, which recurrently updates the disparity map using local cost values.

Several works introduced modifications to this idea. IGEV-Stereo [Xu et al., 2023a] introduces the geometry encoding volume to extend the all-pairs correlation volume and regress a better initial disparity. Instead of using the GRU to update the flow field, Wang et al. [2022] repurposed it to predict the depth probability of each pixel. Zhao et al. [2023] propose improvements in the iterative process to preserve detail in the hidden state by decoupling the disparity map from the hidden state and implementing a normalization strategy to handle large variations in disparities. EAI-Stereo [Zhao et al., 2022] replaced the GRU with an error-aware iterative module.

4.2.3 Efficiency

In a structured light setting [LeMoigne and Waxman, 1988; Vuylsteke and Oosterlinck, 1990; Martinez and Stiefelhagen, 2013], projected patterns are designed to uniquely identify the depth of objects at each position. Hence, the problem can be solved more efficiently for known light patterns, as demonstrated by e.g. Hyperdepth [Fanello et al., 2016] using a random forest approach and the branching network in Gigadepth [Schreiberhuber et al., 2022]. Note that this is different from our setting based on the work of Guo et al. [2019a] where multiple, potentially overlapping, patterns are projected into the scene.

Turning to wider stereo vision challenges, the bottleneck with cost volumes is their large search space, which requires considerable computation and storage to find the desired disparity. Khamis et al. [2018] reduced the computational cost by refining the disparity from a low-resolution cost volume through multiple levels of resolution. Additionally, recent works [Bangunharcana et al., 2021; Wang et al., 2021d] stress real-time disparity estimation in stereo vision. While Shamsafar et al. [2022] relies on lightweight architectures to optimize resources, Garrepalli et al. [2023] introduced DIFT as a mobile architecture for optical flow that uses just-in-time computation of the correlation to reduce peak memory use and served as the inspiration for our correlation computation in the iterative updates. SCV-Net [Lu et al., 2018] builds a sparse correlation volume that resembles dilated convolutions controlled via a fixed sparsity value and without dependence on the inputs. Lastly, SCV-Stereo [Wang et al., 2021c] is an alternative approach to sparse correlation volumes. Different from their method, we use kNN correlation for the initial disparity estimate instead of zero initialization and compute dense correlations on an ad hoc basis during the iterative stages.

4.3 Visual Hull Stereo

The overall structure of our method is based on RAFT-Stereo [Lipson et al., 2021] and is shown in Fig. 4.2. It consists of three stages. First, the pair of input images is encoded into a feature representation using a pre-trained encoding network. These features are then used to compute an initial correlation cost volume. Together with prior information attained from a set of image masks of the scene, a sparse set of k disparities with the highest correlation values is selected from which an initial disparity value is estimated (Sections 4.3.1 and 4.3.2). Following, the disparity is iteratively refined using a *Convolutional Gated Recurrent Unit* (ConvGRU)-based network and upsampling network [Xu et al., 2023a], without the need to hold the full cost volume in memory at any time (Section 4.3.3).

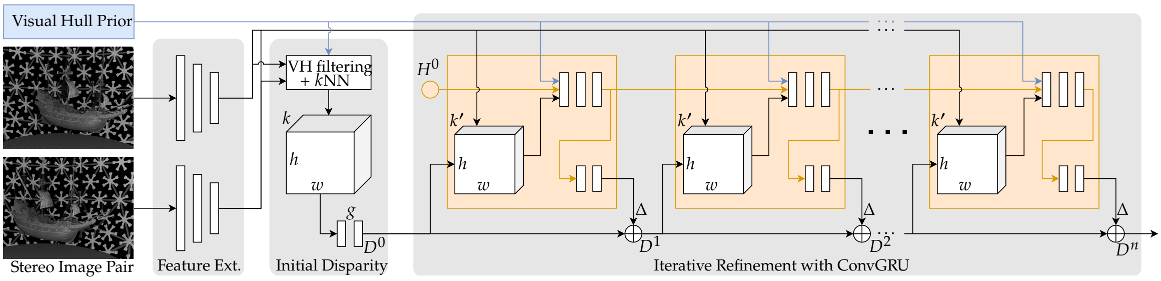


Figure 4.2: Overview of the three stages of our disparity estimation network VHS. Following the *Feature Extraction* we compute an *Initial Disparity* estimate D_0 from a sparse kNN cost volume restricted by the visual hull. Next, we perform an *Iterative Refinement* of the disparity guided by the visual hull prior using ConvGRU modules and dense local correlations with window size k'.

4.3.1 Sparse Correlation

Given a rectified stereo pair, we use a shared feature encoding network [Xu et al., 2023a] to extract features at 25% of the original image size. This representation is used to compute an initial set of the k best matches. First, we define the cost $c_p(d) \in \mathbb{R}$ of disparity $d \in [0, w]$ at pixel $p \in \mathbb{N}^2$ as the inner product of the corresponding feature vectors $f_p, g_{p-(0,d)^T}$, from the left and right pictures of size $h \times w$, where $g_{p-(0,d)^T}$ represents the feature vector at the pixel in the right image offset by d:

$$c_p(d) = f_p \cdot g_{p-(0,d)^T} \tag{4.1}$$

Storing the full set of correlation values at high resolutions can be inefficient and resource-intensive, as the dense cost volume scales quadratically with the image width when the maximal disparity is properly adjusted. To decrease the memory requirements, we instead use a sparse correlation cost volume, which assigns to each pixel p a much smaller subset of correlation values c and corresponding disparity values d,

$$\mathcal{M}_p = \{(d, c_p(d)) \,|\, d \in \mathcal{D}_p^{k\mathrm{NN}}\}, \tag{4.2}$$

where $\mathcal{D}_p^{k\mathrm{NN}}$ represents the set of k best disparities for each pixel:

$$\mathcal{D}_p^{k\mathrm{NN}} = \underset{\tilde{\mathcal{D}}_p \subset \mathcal{D}_p, |\tilde{\mathcal{D}}_p| = K}{\arg\max} \sum_{d \in \tilde{\mathcal{D}}_p} c_p(d) \tag{4.3}$$

Here, $\mathcal{D}_p$ is the set of all disparity candidates for pixel p.

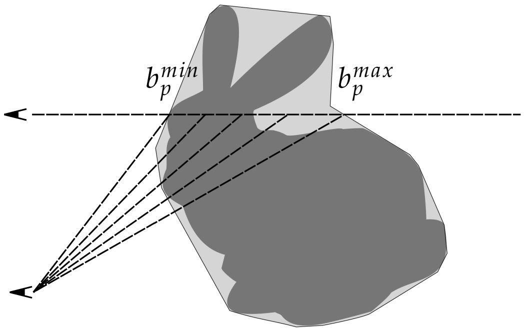


Figure 4.3: Estimation of the disparity boundaries (b_p^{min}, b_p^{max}), from two rectified views of an object's visual hull. The visual hull encloses the objects' surface, so the surface is guaranteed to lie within the disparity boundaries.

4.3.2 Visual Hull Prior

This search for the best candidates can be further improved by inducing a prior based on image masks from the scene. The visual hull, as defined by Laurentini [1994], provides an efficient approximation of an object's shape derived from silhouettes captured by multiple cameras. In adherence to the representation proposed by Scharr et al. [2017], we compute the visual hull using a collection of masked input images, which is stored within an octree structure for compact storage and fast access. The octree is designed such that each leaf node indicates whether it is inside or outside the visual hull. Given this information, we calculate the hull boundaries by sampling rays projected into the scene from the reference view and evaluating these rays for transitions between outside and inside regions of objects. From these transitions, we create depth limits for each camera viewpoint and define disparity boundaries $b_p = (b_p^{min}, b_p^{max})$ based on pixel location p, as illustrated in Figure 4.3. The insight that the surfaces of the objects are confined within the interval $[b_p^{\text{min}}, b_p^{\text{max}}]$ can be leveraged to reduce computational requirements when computing the initial disparity map D^0.

We streamline the k-nearest-neighbor search, previously performed across an expansive set of disparity candidates $\mathcal{D}_p$ for pixel p as described in (4.3), by focusing only on disparities constrained within b_p:

$$\mathcal{D}_p^* = \{d \,|\, b_p^{min} \leq d \leq b_p^{max}\}, \qquad \mathcal{D}_p^* \subseteq \mathcal{D}_p \tag{4.4}$$

This approach allows for a faster computation of the restricted correlation cost volume $\mathcal{M}_p^*$ by skipping unnecessary evaluations of the correlation. Accordingly, we define our initial disparity map as follows:

$$D_p^0 = \sum_{l=1}^{K} d_l \cdot g(c_p(d))_l, \qquad (d, c_p(d)) \in \mathcal{M}_p^* \tag{4.5}$$

where g is an attention-based transformation network with a softmax function as the last layer.

4.3.3 Iterative Disparity Refinement

We use a hierarchical ConvGRU network on three resolutions to iteratively refine the predicted disparities starting with the initial values D_p^0, similar to Xu et al. [2023a]: The network updates a hidden state H^i taking the current disparity values and contextual features extracted from the corresponding image data, and the correlated features around the current disparity estimate as input. The new state is used to predict an offset Δ_p^i from which the refined disparity values are computed as

$$D_p^{i+1} = D_p^i + \Delta_p^i. \tag{4.6}$$

Memory Efficient Correlation Instead of sampling correlation values from a pre-computed full cost volume, we propose to compute a local correlation volume ad hoc to reduce memory usage. This volume is bounded within a window W_p^i of size $2r + 1$, which is centered on the currently estimated disparity D_p^i,

$$W_p^i = [D_p^i - r, D_p^i + r], \tag{4.7}$$

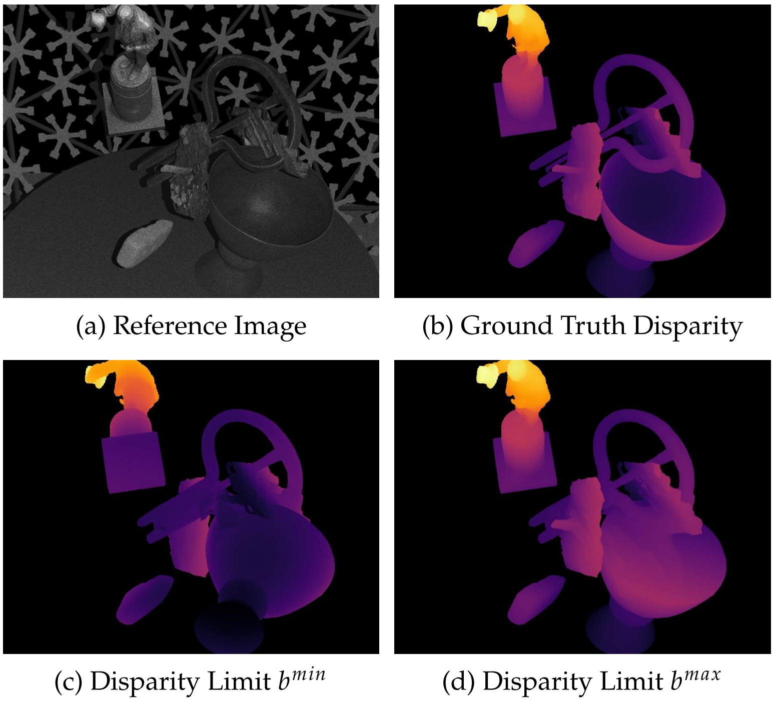
where we fix $r = 4$ following Xu et al. [2023a]. We compute the correlations group-wise, as originally proposed by Guo et al. [2019b], by dividing the feature vectors into a set of subgroups. Please note that, for the initial disparity D_p^0, we strategically omitted the group-wise correlation calculation. This is due to the complexity of uniquely defining kNN for group-wise correlations, ensuring that our approach remains computationally efficient.

Visual Hull as Weak Prior As additional information, we supply the ConvGRU with a flag $f_p(d)$ that guides the network to predict a value within the visual hull,

$$f_p(d) = \begin{cases} 1 & \text{if } d \in D_p^*, \\ -1 & \text{otherwise} \end{cases} \tag{4.8}$$

for each disparity value d within the window W_p^i. In that way, the limits b_p obtained from the visual hull operate as a weak prior guiding the disparity regression while retaining valuable correlation information for cases such as incorrect limits due to masking errors.

One distinct advantage of our visual hull guidance is that the disparity limits are an optional input to the whole pipeline. During the initial sparse correlation, we can fall back to sampling from all values below a pre-defined threshold in the same manner as established models, and during the dense updates, we set $f_p(d) = 0$ to indicate missing information. This enables the application of our sparse correlation method even without masked measurements and pre-training of our method on existing datasets.


(a) Reference Image

(b) Ground Truth Disparity

(c) Disparity Limit b^{min}

(d) Disparity Limit b^{max}

Figure 4.4: Sample from the FlyingObjaverse training dataset. Notice how the true disparity is close to the upper disparity limit except for the basin in the bottom right, which cannot be recovered from the visual hull.

4.4 Training Details

Given the particular nature of our method in terms of target application and required inputs, a boilerplate training procedure following the literature would be unproductive. Therefore, we present custom training details tailored to our use case, covering the preparation of custom data along with training strategies. We further introduce a memory-efficient approach enabling training at even higher resolutions.

4.4.1 Dataset Preparation

Common stereo datasets like SceneFlow [Mayer et al., 2016] do not contain ground truth meshes or auxiliary views, which prevents the extraction of a meaningful visual hull. As an alternative, we render a custom dataset with Mitsuba 3 [Jakob et al., 2022] and meshes from Objaverse-XL [Deitke et al., 2023] to train our network. The dataset generation loosely follows the approach of SceneFlow by placing objects on a virtual capture stage. Each scene contains a randomly transformed arrangement of $1 - 10$ objects, as shown in Fig. 4.4, with an infrared camera stereo setup using active illumination with projected patterns similar to Guo et al. [2019a] and a total of 68 cameras for the masks, all captured at a resolution of 4608×5328. We render 2 stereo pairs for 500 scenes. For testing, we follow the same rendering pipeline but select meshes from different sources to avoid contamination of the training dataset. To test performance on difficult lighting effects, we curated scenes with objects that include challenging reflectance properties and fine details using high-quality meshes from Polyhaven[1] and build eight scenes, each viewed from four different angles. As a second test set, we used SMPL [Loper et al., 2015] human models with texture from SMPLitex [Casas and Trinidad, 2023] to evaluate performance on human subjects. We create 100 scenes by combining random poses from the animations with random textures and render 2 stereo pairs for each scene.

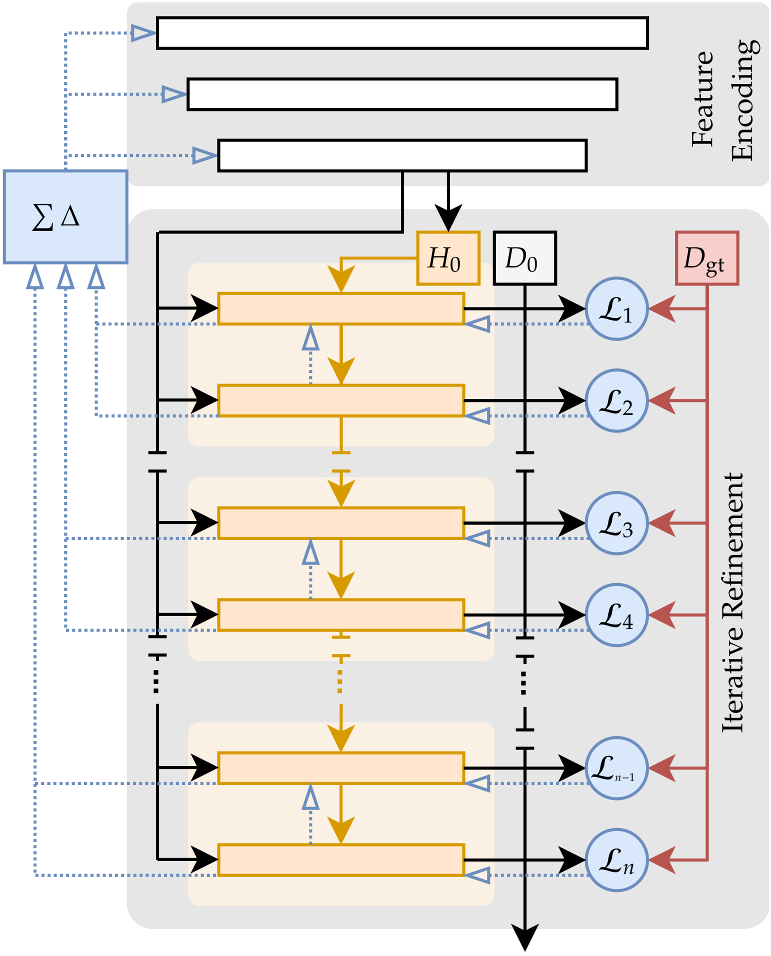


Figure 4.5: Memory efficient training scheme for $n = 2$ consecutive update steps. After the computation of the losses $\mathcal{L}_i$ and $\mathcal{L}_{i+1}$, we perform backpropagation to accumulate gradients of the update network parameters and detach the hidden state effectively freeing the computational graph. $\sum\Delta$ indicates an optional accumulation of gradients to avoid multiple backward passes through the feature extraction network.

[1] https://polyhaven.com/

4.4.2 Training Strategy

Having the visual hull guidance as an entirely optional component, allows our method to harness a more flexible training process and to predict the disparity map even without any pre-calculated masks. We use this flexibility in our experiments by pre-training a base model on Sceneflow [Mayer et al., 2016] and subsequently fine-tuning the network on our custom training data. The training is performed on SceneFlow final pass for 20 epochs using AdamW [Loshchilov and Hutter, 2017] with a one-cycle learning rate schedule with a learning rate of 0.00015 and a batch size of 4. We use random crops of size 288 × 640, random y-jitter and occlusion as augmentation, and an L_1 loss following the weighting of RAFT-Stereo [Lipson et al., 2021]. This model serves as our baseline for a benchmark evaluation on the SceneFlow test set. Subsequently, the network is fine-tuned on the simulated data of Objaverse-XL (Section 4.4.1) for high-resolution stereo following the same settings, except for a magnified random cropping of 256 × 2048, batch size of 1 and with the additional visual hull inputs, which we randomly drop for $\frac{1}{8}$ of the samples. Note that we use RGB inputs for the benchmark comparison and greyscale for the simulation of IR images for all other experiments.

Memory Efficient Training During the training of most iterative methods, each update of the disparity consumes more VRAM since the full compute graph needs to be stored in memory. We propose to split the forward and backward computation in a manner that reduces the memory requirement while still retaining accurate gradient information as shown in Fig. 4.5. For n consecutive update steps we compute the losses on the upscaled disparity predictions as usual. Then, we backpropagate the partial loss and detach the hidden state such that the computational graph can be erased. To avoid multiple backward passes through the costly feature extraction network, we propose to optionally accumulate all gradients for the feature vectors first before performing a final backpropagation after all iterations are through.

Technical Details Using CUDA, we build a visual hull octree from rendered masks from which the disparity limits are computed. Our network is implemented in Pytorch with custom CUDA kernels for the correlation computations and we use warp-level shuffle operations to make the initial kNN correlation computation efficient. As such, the number of candidates is limited to 32, but we use 8 for all experiments following Wang et al. [2021c]. All our experiments were conducted on an NVIDIA GeForce RTX 4090.

4.5 Experiments

We evaluate our method in terms of average end-point error (EPE) in pixels, proportion of errors (> 4px in %) and the D1 outlier rate [Menze and Geiger, 2015]. Runtime and video memory measurements follow the literature and employ automatic mixed precision.

Method	#Params	$\text{EPE}_{\leq 192}$	EPE_{all}
CascadeStereo [Gu et al., 2020]	10.5M	0.67	3.30
CFNet [Shen et al., 2021b]	23.0M	0.96	3.06
CoExNet [Bangunharcana et al., 2021]	3.5M	0.69	3.36
FADNet++ [Wang et al., 2021d]	12.4M	0.88	3.55
GwcNet [Guo et al., 2019b]	6.9M	0.76	3.52
IGEV-Stereo [Xu et al., 2023a]	12.6M	0.48	3.01
MSNet2D [Shamsafar et al., 2022]	2.3M	1.11	3.76
MSNet3D [Shamsafar et al., 2022]	1.8M	0.79	3.44
PSMNet [Chang and Chen, 2018]	5.2M	1.02	3.69
VHS (ours)	12.7M	0.89	2.33

Table 4.1: Comparison on SceneFlow final pass test set using the model implementations from Guo et al. [2023].

Prior	EPE_{all}	EPE_{noc}	$> 4\text{px}_{\text{all}}$	D1_{all}
No	1.48	0.83	4.6	0.93
Initial	1.29	0.75	4.3	0.68
Update	1.04	0.57	3.3	0.46
Both	0.98	0.55	3.2	0.40

Table 4.2: Ablation of the visual hull guidance on the Polyhaven Test set.

4.5.1 Benchmark Evaluation

We first validate the correctness of our sparse-dense correlation network compared to the state-of-the-art, with all methods being trained on SceneFlow. Table 4.1 shows that our method performs competitively in terms of EPE for disparities within the range that all methods can handle. Specifically, for pixels with true disparities less than or equal to 192 ($\text{EPE}_{\leq 192}$), our method matches with FADNet++ [Wang et al., 2021d], with only three methods achieving better scores. Notably, when evaluated on all pixels (EPE_{all}), our method surpasses all baseline models as we do not have any upper limit to the possible disparity.

Also, our method requires less memory during both inference and training as shown in Fig. 4.8 and is as fast as IGEV-Stereo [Xu et al., 2023a] during inference while having a minor runtime overhead during training.

4.5.2 Visual Hull Guidance

To further demonstrate our performance on high-resolution data with larger disparities using the additional visual hull input, we evaluate our method on the two test datasets after fine tuning on the training dataset as described in Section 4.4.1. As shown in Table 4.3, our

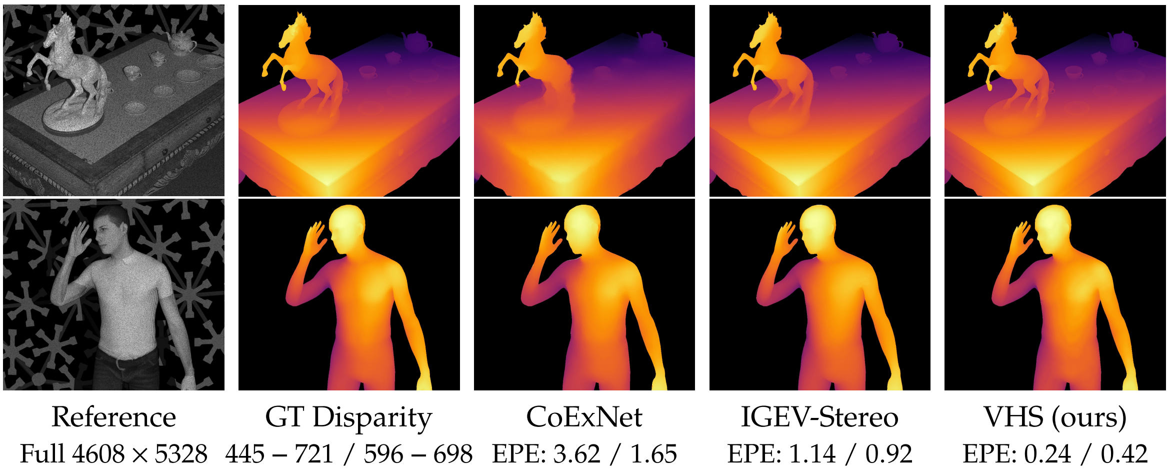


Figure 4.6: Qualitative results compared to IGEV-Stereo [Xu et al., 2023a] and CoExNet [Bangunharcana et al., 2021] on samples from the Polyheaven and SMPL test sets. Note the faithful reconstruction of the plates (top) and the chest (bottom) produced by our method. We show the range of disparity values below the GT disparity and the EPE below the methods.

method outperforms all other methods on both the Polyhaven and SMPL datasets across all metrics. Specifically, we achieve significantly lower EPE_{all} and EPE_{noc} which indicates higher overall accuracy, and a higher accuracy in non-occluded regions. We further highlight the robustness of our method by showing the lowest percentage of pixels with large disparity errors ($> 4\text{px}_{\text{all}}$, D1_{all}). We present qualitative results in Fig. 4.6. Note that most baseline models cannot perform inference on the full resolution inputs using common hardware as they exceed the available memory (24 GB in our case) and cannot capture the large disparity values in our data as the correlation volumes are typically limited to 192 pixels. For this evaluation, we resort to running the models on 2× or 4× downsampled input images and reduce the offsets by aligning them using the known minimum ground-truth disparity of the foreground, selecting the best variant of both resolutions based on the smallest EPE.

To study the performance benefit of the visual hull, we perform an ablation study on the Polyhaven test set, as shown in Table 4.2. While applying visual hull guidance only for the initial disparity calculation already shows a minor improvement across all metrics compared to an uninformed run, the weak prior during the iterative updates yields a major gain. Ultimately, we achieved the best results by employing visual hull guidance in both phases. The improvement is particularly remarkable considering that the majority of the object points do not lie directly on the visual hull.

As the quality of the visual hull depends on the correctness of the masks, we additionally study the influence of incorrect matting on the performance of our method in Fig. 4.7. We find that our method is robust against binary dilation on the masks, while larger binary erosion reduces the accuracy. Intuitively, this makes sense as a correct visual hull always encloses the true surface, which is also the case for “inflated” visual hulls from dilated masks, while “deflated” hulls from eroded masks violate this assumption.

Method	Polyhaven				SMPL			
	EPE_{all}	EPE_{noc}	$> 4\text{px}_{\text{all}}$	D1_{all}	EPE_{all}	EPE_{noc}	$> 4\text{px}_{\text{all}}$	D1_{all}
CascadeStereo [Gu et al., 2020]†	16.97	14.37	31.1	6.77	8.31	6.51	13.8	2.97
CFNet [Shen et al., 2021b]†	14.50	11.98	31.4	7.80	13.28	12.48	9.8	3.74
CoExNet [Bangunharcana et al., 2021]*	9.78	8.57	25.9	7.21	2.98	2.38	8.6	1.56
FADNet++ [Wang et al., 2021d]*	11.44	10.49	25.3	7.82	2.67	1.85	6.8	1.64
GwcNet [Guo et al., 2019b]†	19.97	17.04	35.8	9.60	11.27	10.34	14.9	3.86
IGEV-Stereo [Xu et al., 2023a]*	5.22	4.10	16.6	3.94	1.68	1.27	6.2	0.83
MSNet2D [Shamsafar et al., 2022]†	10.08	8.69	44.2	5.67	5.24	4.44	28.9	2.38
MSNet3D [Shamsafar et al., 2022]†	14.41	11.95	32.3	7.65	9.78	8.36	12.3	3.40
PSMNet [Chang and Chen, 2018]†	13.19	11.28	37.8	6.11	17.38	16.31	17.9	4.55
VHS (ours)	0.98	0.55	3.2	0.40	0.54	0.41	0.9	0.10

Table 4.3: Comparison on our data using the model implementations from Guo et al. [2023]. Methods marked with * run on half resolution with inputs aligned to set minimum disparity to zero. † on quarter resolution with inputs aligned to set minimum disparity to zero.

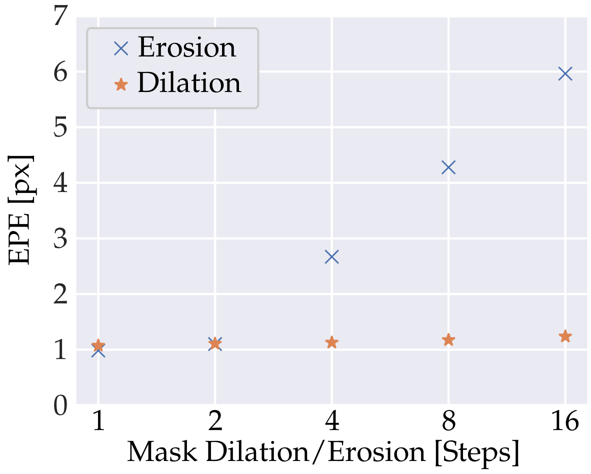


Figure 4.7: Correlation between mask accuracy and EPE, demonstrating the method's robustness to binary dilations to the correct mask.

Variant	Full Backprop.		Detached Features	
	GB	ms	GB	ms
-	14.18	377	-	-
16	8.71	441	8.49	586
8	5.85	497	5.62	584
4	4.42	611	4.19	583
2	3.69	840	3.46	583

Table 4.4: Peak memory and average runtime per iteration comparing the standard training procedure (first row) with our proposed memory-efficient training running backpropagation through the full network each time (left) and accumulating the feature gradients first (right) for different numbers of connected updates. Measured for a single stereo pair at 512 × 1024.

4.5.3 Training Scheme

To evaluate the impact of the memory-efficient training scheme on memory usage and runtime, we estimated these metrics for different numbers of connected updates before backpropagation in relation to the standard training procedure. We compared a setting with full backpropagation to a setting with the detached feature extraction and measured for the former a reduction in memory usage at the cost of increased runtime for a smaller number of connected updates, as shown in Table 4.4. In comparison, the detached features offer a stable runtime even at as few as two connected layers with an even further reduction in memory usage compared to full backpropagation.

Finally, we evaluate the impact of including pre-training on SceneFlow in our training procedure. A network trained using only our Objaverse-XL-based dataset yields an EPE of 1.33 on the Polyhaven test set, compared to 0.98 of a full training, indicating a significant benefit of the hybrid approach.

4.6 Conclusion

We have presented a technique to induce visual hull priors into recurrent stereo networks to improve matching performance. Combined with a novel sparse-dense correlation handling, our approach accurately regresses disparity for high-resolution images while retaining a favorable memory footprint and without an upper limit on the achievable disparity.

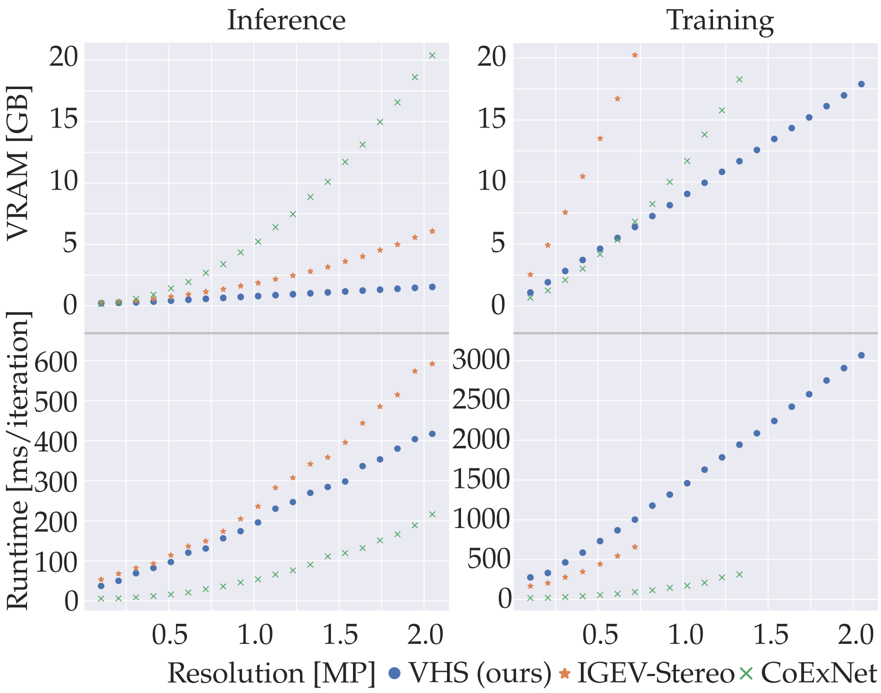


Figure 4.8: Memory and runtime statistics of our method compared to the best-performing (IGEV) and fastest (CoExNet) baseline methods. We fix the image height at 320 px and increase the width, adjusting the maximum disparity to $\frac{1}{4}$ of the latter.

Chapter 5

Frame Interpolation Transformer and Uncertainty Guidance

In this chapter, we discuss the contributions and results developed in the following peer-reviewed publication:

> Markus Plack, Karlis Martins Briedis, Abdelaziz Djelouah, Matthias B. Hullin, Markus Gross, and Christopher Schroers.
> "Frame Interpolation Transformer and Uncertainty Guidance."
> *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
> DOI: 10.1109/CVPR52729.2023.00946

5.1 Summary of the Publication

In this work, we present a frame interpolation method to address the problem of rendering high-quality videos for e.g. feature film productions. Since the accurate simulation of visual effects at high resolutions requires large amounts of computing capacity, approaches to speed up this process are highly desirable, as they not only reduce the environmental burden but also enable artists to work more efficiently. The underlying idea of our approach is to render only every second frame and use a frame interpolation method to complete the sequence. This simple approach can reduce rendering time by almost a factor of two, since the interpolation effort is negligible compared to the rendering, but has several drawbacks which we address in our work.

Most prominently, none of the existing frame interpolation methods today are capable of producing outputs with consistently high quality, as complex motions, varying illumination, and difficult visual effects still pose a significant challenge. We propose a motion-based network architecture that uses a transformer module to fuse features during the hierarchical updates. Paired with a novel deep feature extraction, we show that our method improves upon the state of the art both quantitatively and qualitatively. We demonstrate this using common benchmark datasets as well as a dataset collected from animated movies and through a user study. Our architecture employs a commonly used bottom-up processing paired with warping-based motion compensation. For the feature extraction, we build on previous work [Reda et al., 2022] which produces semantically similar feature vectors on multiple resolutions and includes a U-Net architecture to enable learning of more meaningful features even on the upper levels.

Nevertheless, the improved architecture alone does not reach the target quality for all inputs and there is no way to know how good the interpolation is. To solve the second problem, we build an uncertainty prediction into the network, by adding two maps to the outputs that aim to predict the quality of the interpolation. Since the true image is known during training, we can compare it with the prediction to train the error maps. The first map is trained on the L_2 norm between the images and the second map on the perceptual error LPIPS without spatial averaging. While simple to implement and without any overhead at inference time, we show that this method not only works well at identifying problematic areas in the intermediate frame but also very slightly improves interpolation quality. This enables us to take advantage of the fact that we work with rendered content by explicitly rendering those regions. While those additional renderings increase the computational load again, we show that even for fractions below 10% of the full frame a significant increase in quality can be attained. Furthermore, we propose to not only overlay or blend those new parts but instead use them during a second pass through the network. This is possible, since the transformer architecture of our network can distinguish the target and input frames based on a binary mask input indicating valid content. This way, the network can be trained to make use of the additional data and we show that this technique improves the output quality more than a simple replacement.

Though designed for this use case, our method also works for live-action content. Generating additional inputs might not be feasible for most applications, but for certain footages, manual creation of partial intermediate inputs might be desirable. Additionally, the uncertainty estimation can be used to guide this process or at least for quality control purposes, where certain thresholds could be determined based on the expected error.

In summary, we propose to address the problem of speeding up the rendering of sequences using a two-step approach that predicts the uncertainty of the output and incorporates partial renderings of the intermediate frame. This is implemented through a transformer-based network architecture and improved by a better feature extraction.

5.2 Author Contributions of the Publication

In this work, I implemented the network and training procedure using libraries and resources from Disney Research | Studios, where most of the work was done during an internship. The main parts of the network I implemented were an extension of the feature extraction of Reda et al. [2022], the transformer architecture of the interpolation network, the uncertainty output, and the handling of masked inputs. In addition, I implemented a procedure to test the uncertainty guidance on simulated data. Many of the ideas were developed together with Karlis Martins Briedis, Abdelaziz Djelouah, and Christopher Schroers. Finally, I performed most of the evaluation, including the quantitative comparisons with the state of the art, preparing the animated datasets from open source movies, running and evaluating the user study, the ablation study, and the additional input tests with guidance from my collaborators and advisors.

Chapter 6

Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction

In this chapter, we discuss the contributions and results developed in the following peer-reviewed publication:

> Markus Plack, Clara Callenberg, Monika Schneider, and Matthias B. Hullin.
> "Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction."
> *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.
> DOI: `10.1109/WACV56688.2023.00308`

6.1 Summary of the Publication

This work presents a differentiable transient renderer and demonstrates its practicality for various applications. It builds on the approach of Iseringhausen and Hullin [2020], but extends the work in various ways, improving speed and versatility.

A physically accurate path tracing of a NLoS scene like the method presented by Jarabo et al. [2014] is a computationally heavy task by itself and using any such technique in an optimization scheme might not seem feasible. Nevertheless, Iseringhausen and Hullin [2020] have presented a fast renderer using various approximations along with an efficient GPU implementation that enables an analysis-by-synthesis approach to NLoS reconstruction. This is achieved using central differences for the optimization of parameters, which is the major drawback addressed by our work, as the computation time scales linearly with the number of parameters. Even with a clever technique for the selection of optimization variables a full reconstruction can take a full day or even more. Instead, we propose to implement the renderer in a differentiable fashion using backpropagation. This is possible since the approximations used to render the meshes are fully differentiable functions. We achieve this by splitting the computation into several modules which we then implement as Pytorch functions and add the manually derived backward pass. First, we build a spatial volume grid of the hidden scene from the Gaussians by summing all contributions. This volume is transformed in a second step to a triangle mesh using marching cubes and finally rendered in the last step, which includes visibility testing. All functions are fully implemented as CUDA kernels and the raycasting is done using NVIDIA OptiX.

Having access to such a more flexible and fast method allowed exploring various extensions, applications, and geometry representations. We show that it is also possible to backpropagate the loss to the albedo of the triangles. First, we added an albedo component to each Gaussian, similar to an existing implementation for the baseline model, which is transformed into an albedo volume and added as an attribute to all vertices during marching cubes. We extended this by computing the derivatives needed to implement the backward functionality of all steps with respect to the albedo and evaluate this approach on a rendered example of a textured model. The ability to optimize albedo values naturally enables us to optimize colored depth maps as an alternative representation of the hidden scene. For this, we optimize a grid of offsets and another for albedo values, from which the triangles are computed and rendered as described above. To improve convergence speed, we propose to infer depth and albedo in a coarse-to-fine scheme, starting with the optimization of rough maps and iteratively optimizing the upsampled maps until the desired resolution is achieved. We add a TV regularization to the depth map and the albedo map separately to improve the stability of the optimization. The greatest advantage of this representation compared to the Gaussians is that it does not rely on an algorithm for the optimization of adding and removing blobs. In addition, we have shown that the optimization is faster and the result mesh is more accurate. Our method can also handle high-resolution inputs by switching to a stochastic optimization keeping the optimization time low, which we show on an existing measured dataset.

As differentiable rendering approaches can be prone to overfitting, especially on real-world data resulting in e.g. erroneous Gaussians or jagged depth maps we propose a background network to capture any effects that are not part of our model. This network resembles implicit representations and is trained to produce transient responses given position-encoded coordinates of the scanning positions on the relay wall. Its output is added to the rendered images and the parameters are optimized together with the scene parameters. To avoid capturing more than needed for a faithful reconstruction we propose to add a condition that limits the average power of the background transient spectra compared to the rendered ones.

Besides the inference of the hidden geometry, we show that our method can also be used for locating and tracking known objects that are hidden from sight. We achieve this by transforming the fixed object vertices using a rotation and translation for the rendering and optimizing the related position vector and orientation quaternion. We show that this also works for partially occluded objects, since our renderer explicitly checks for each triangle's visibility, albeit with a lower accuracy. Lastly, we also demonstrate the application of the differentiable renderer for self-supervised training of a reconstruction network by computing the loss between the rendering of the resulting density volume and the input images without the need for any ground truth geometry.

Overall, we present a fast and versatile differentiable transient renderer and demonstrate its use for various applications. We evaluate different geometry representations and propose a method of capturing unknown effects which improves reconstructions of captured scenes.

6.2 Author Contributions of the Publication

I implemented the differentiable rendering and the differentiable marching cubes within the Pytorch framework based on the CUDA/C++ implementation of the forward pass of Iseringhausen and Hullin [2020] and did the derivations for the backward pass. Initial tests for colored Gaussians were done by Monika Schneider within the original framework and I replicated and extended them in my work for colored depth maps. Regarding the evaluation and experiments, Matthias Hullin provided the flat-field correction for the “Diffuse S” dataset, and Clara Callenberg the visualization of the tracking metrics, but the remainder was done by me.

Part III

Conclusion

Chapter 7
Conclusion

We provide a summary of the works contained in this thesis and their impact in Section 7.1 and discuss their limitations along with possible future research directions in Section 7.2 before giving some final remarks in Section 7.3.

7.1 Summary and Impact

We presented three approaches for the integration of spatial priors and uncertainty into reconstruction methods, showcasing how they improved the quality and/or efficiency of the methods.

High-Resolution Stereo Matching

For stereo matching, we proposed an approach that takes a step towards efficient and accurate disparity estimation for high-resolution images. As the targeted application lies in depth estimation from stereo rigs that are part of capture stages, we proposed a technique to integrate a spatial prior in the form of a visual hull computed from matted images of other views of the scene. This allows us to reduce the correlation computation, which is the foundation of most state-of-the-art methods today, to a valid range and carve out space that is known to be empty. Paired with a sparse-dense approach to correlation volumes our method improves the matching efficiency of the model. In addition, we proposed a training scheme that enables learning at high resolutions without exhaustive memory requirements and closes the domain gap between train and test data. While the visual hull integration is limited to specific use cases where other views of the scene are given, our method is designed to be versatile and work without this additional input, enabling its application in other scenarios, especially for high-resolution inputs, which will still benefit from the sparse correlation computation and the proposed training scheme. Nevertheless, we can envision the integration of other forms of priors into our method in a similar fashion, since any method that assigns one or more values to each disparity estimate is easily integrated into the model without any major restrictions.

Frame Interpolation for Video Rendering

We expanded the frame interpolation problem to a use case in video rendering where generating additional partial inputs for the target frame is possible. We proposed a transformer-based architecture to handle the masked inputs as priors for the interpolation and an uncertainty prediction to guide the rendering process. With an improved feature extraction, our method is also capable of producing intermediate frames in the traditional setting without additional content and we show that it improves qualitatively and quantitatively upon the state of the art. The proposed architecture merges the approach of a transformer network with a motion-based interpolation/flow network, resulting in efficient feature propagation even for large motions. While designed for frame interpolation, this architecture can likely also be adapted to other vision problems working with videos, such as optical flow estimation, video denoising, or segmentation and classification tasks. The uncertainty estimation on the other hand can easily be incorporated into a multitude of other approaches, not limited to videos. Despite being a straightforward approach with no theoretical guarantees, which limits its application to non-safety-critical and non-medical scenarios, we have demonstrated that it performs adequately for frame interpolation and we surmise it can deliver a similar performance when applied to other problems.

Differentiable Non-Line-of-Sight Rendering

In the context of transient imaging, we have presented a fast differentiable transient renderer and its application to selected NLoS problems. We have proposed the optimization of depth maps with additional albedo information to represent the hidden scene, which improves speed and accuracy compared to the baseline approach using Gaussians. This approach was made possible by implementing gradient backpropagation for the rendering functions, which resulted in reduced computational complexity of optimizing many parameters. Aside from the efficient implementation, we proposed a background network to capture unknown effects for an improved reconstruction. Such a technique is motivated by the observation, that there is a significant gap between rendered and captured data. This can be due to inaccuracies within the calibration of the capturing system, non-homogeneous relay walls, additional reflective geometry outside the reconstruction bounds, varying surface properties, and other deviations from the model assumptions of which interreflections are the most prominent as they are ignored by our renderer. Our background network has been adapted by Fujimura et al. [2023] in their reconstruction pipeline using neural implicit surfaces. The latter offers a different approach to spatial priors compared to the ones used in our work. The importance of research in this direction is also highlighted by the work of Choi et al. [2023], where they propose among other things the optimization of a volumetric intensity which serves as a basis for an implicit surface representation yielding accurate and complete reconstructions. They show that in comparison our approach is missing geometry albeit being highly accurate on the reconstructed surface.

7.2 Limitations and Outlook

While the methods presented in this thesis have demonstrated great performance and advanced the state of the art in their respective fields, the problems are still far from being solved and certain remaining limitations need to be addressed in future research. We provide an overview of those shortcomings and the open challenges, along with suggestions for promising research directions.

Multi-View Stereo Our current stereo method is limited to the estimation of the disparity between two rectified images with additional matted views to guide the model. At the same time, a large body of work exists that builds reconstructions from multi-view stereo (MVS) setups, explicitly using the full information contained in the captured images. It would be interesting to see how the optimizations proposed in our work can be adapted and extended to MVS, enabling high-resolution matching in this domain. In this regard, an open question would be the adaptation of the sparse KNN correlation, as an MVS setup with n cameras would require the computation of a vector of size $n - 1$ for each depth, for which no straightforward ordering exists. Another particularly promising direction in the context of capture setups is the open question of how to combine images of visible light, i.e. RGB, and infrared recordings for improved depth estimation. Since some materials exhibit vastly different behavior within the visible spectrum and for infrared light a method to produce consistent feature vectors to match both domains needs to be found. This can be even more problematic if structured light projectors are used for infrared, as those patterns are not visible in the other images. Another open problem would be the training of such a method, as no training data exists, and models that contain materials that also model the behavior on infrared light are scarce.

Real Time Stereo and Temporal Consistency Many applications would greatly benefit from an accurate disparity estimation in real-time. While our method is explicitly designed to be resource-efficient to enable matching of high-resolution inputs, it is still not particularly fast and especially not real-time capable. The open question is how to further optimize the model such that this would be possible while keeping the loss in quality low. There are several possible approaches that we can suggest for future evaluation. First, hierarchical models, which regress the disparity in a coarse to fine matter could be a better choice for such an application, as they tend to be faster than iterative methods like ours. Second, we have seen that feature extraction is responsible for a significant portion of the memory and runtime requirements. This is, however, not a straightforward point of optimization, as lower-quality feature vectors will likely lead to a considerable reduction in quality while only offering a minor benefit in terms of runtime. Lastly, it would be possible to consider temporal consistency, as real-time applications imply stereo video inputs. Having an initialization from the previous time step that is likely to be close to the solution for the current frame, possibly applying a cheap extrapolation using estimated motion vectors, could reduce the required number of iterations in the estimation, and allow using smaller, and hence faster architectures.

Computational Efficiency of Video Transformers One major drawback of video transformers as used by our frame interpolation method is the large computational requirement. While negligible in our case when compared to the immense rendering times of feature film productions, improving their efficiency still remains an open problem for future research. One possible solution is to apply sparse transformer architectures similar to the one used by our stereo matching method. While unlikely to be useful for interpolation between two frames, such an approach could extend our method to work on longer sequences of frames. The importance of research in this direction is e.g. highlighted by the recent advances of Shi et al. [2023] for optical flow estimation and since accurate matches between frames are an essential part of many frame interpolation methods like ours. Another open question for larger temporal windows would be the selection of warping operations. The runtime of the all-to-all warping proposed in our work scales quadratically with the total number of input and output frames, which would be unfeasible for longer sequences. While restricting the warping to the neighboring frames would be feasible this would likely also reduce the achievable improvement from handling larger temporal windows.

Interpolation of large motion and fine objects Similar to most coarse-to-fine frame interpolation methods our proposed approach struggles with large motion of small objects since the refinement of predictions from previous layers can only recover errors within certain bounds given by the limited search range. While all-pairs correlation methods could be a solution, they tend to be computationally more demanding which would be in contradiction to the aforementioned limitation. An alternative solution could likely be found when considering our use case of video frame interpolation for video rendering. One could imagine analyzing the motion vectors which are a standard output of many renderers and identifying objects with large offsets. This could be even used to produce additional rendered patches for the first pass, enabling the method to recover from small errors in the prediction through to e.g. non-linear motion, as long as a fraction of the object was successfully located. At the same time, it might be possible to improve the uncertainty prediction to be aware of such errors. While theoretically possible as is, we found that some more work is needed for this approach to truly work.

Interpolation of longer sequences and better initialization In principle, our frame interpolation method is designed to handle arbitrary sequences. It would be interesting to see how it performs when bridging longer gaps, which would reduce the required time for rendering even further. While this is directly tied to improving the computational efficiency as described above, several other aspects need to be taken into consideration. Do we provide partial renderings for all intermediate frames or only for a subset? Can we even predict regions that are problematic prior to the first pass of the network and provide those as additional inputs? What are good sampling strategies for the additional inputs during training? Finally, in the context of video rendering, it would be possible to incorporate other features such as presented by Briedis et al. [2023], aiding the interpolation of longer sequences.

Spatial Priors for Surface Reconstruction Optimizing Gaussian blobs to represent hidden scenes has two obvious drawbacks. First, it is not trivial to find a good set of discrete steps like adding, removing, and splitting blobs during the optimization, and slight changes in the algorithm and its hyperparameters can reduce the reconstruction quality. Second, certain shapes like planes cannot be represented using isotropic Gaussians, and while anisotropic ones may be better suited for those tasks, they come with their own set of problems. In the future, implicit representations might prove to be useful, but since they do not trivially allow priors, their use in adversarial conditions could be limited.

Background Network The background network in our work is inspired by neural representations but is – by design – limited in its representation capabilities. This limitation is necessary because otherwise, the network could capture arbitrary parts of the signal or even the full input, which would defy its true purpose. Therefore, it is not trivial to improve it based on advances in neural representations. Alternatively, we think it would be interesting to study if it is possible to include other means to guide the network in the optimization, either learned from data or by designing some other system on top of the network.

7.3 Closing Remarks

This thesis is the culmination of a long journey but not necessarily an accurate summary of the whole way, as it leaves out a great deal of struggles, and things that failed, and the tiny steps that lead towards the ideas that actually worked. I hope that you, the reader, found some of the ideas presented here stimulating, that some of the methods I have worked on might prove to be useful to somebody, and that they are a step forward in research. And now, without further ado, onwards to the most important step: The next [Sanderson, 2017].

Bibliography

Abd Gani, Shamsul Fakhar, Muhammad Fahmi Miskon, Rostam Affendi Hamzah, Mohd Saad Hamid, Ahmad Fauzan Kadmin, and Adi Irwan Herman (2024). "Refining Disparity Maps Using Deep Learning and Edge-Aware Smoothing Filter." *Bulletin of Electrical Engineering and Informatics*.

Abramson, Nils (1978). "Light-in-Flight Recording by Holography." *Optics letters*.

Arbel, Julyan, Konstantinos Pitas, Mariia Vladimirova, and Vincent Fortuin (2023). "A Primer on Bayesian Neural Networks: Review and Debates." *arXiv preprint arXiv:2309.16314*.

Arellano, Victor, Diego Gutierrez, and Adrian Jarabo (2017). "Fast Back-projection for Non-line of Sight Reconstruction." *Optics express*.

Argaw, Dawit Mureja and In So Kweon (2022). "Long-Term Video Frame Interpolation via Feature Propagation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bahat, Yuval and Tomer Michaeli (2020). "Explorable Super Resolution." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Baker, Simon, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski (2011). "A Database and Evaluation Methodology for Optical Flow." *International journal of computer vision*.

Bangunharcana, Antyanta, Jae Won Cho, Seokju Lee, In So Kweon, Kyung-Soo Kim, and Soohyun Kim (2021). "Correlate-and-Excite: Real-Time Stereo Matching via Guided Cost Volume Excitation." *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Bao, Wenbo, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang (2019a). "Depth-Aware Video Frame Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bao, Wenbo, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang (2019b). "MEMC-Net: Motion Estimation and Motion Compensation Driven Neural Network for Video Interpolation and Enhancement." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Barnard, Stephen T. and William B. Thompson (1980). "Disparity Analysis of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Barnes, Connelly, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman (2009). "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing." *ACM Transactions on Graphics (TOG)*.

Bemana, Mojtaba, Joachim Keinert, Karol Myszkowski, Michel Bätz, Matthias Ziegler, Hans-Peter Seidel, and Tobias Ritschel (2019). "Learning to Predict Image-based Rendering Artifacts with Respect to a Hidden Reference Image." *Computer Graphics Forum (CGF)*.

Boominathan, Lokesh, Mayug Maniparambil, Honey Gupta, Rahul Baburajan, and Kaushik Mitra (2018). "Phase Retrieval for Fourier Ptychography under Varying Amount of Measurements." *arXiv preprint arXiv:1805.03593*.

Boyd, Stephen, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. (2011). "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." *Foundations and Trends® in Machine learning*.

Briedis, Karlis Martins, Abdelaziz Djelouah, Mark Meyer, Ian McGonigal, Markus Gross, and Christopher Schroers (2021). "Neural Frame Interpolation for Rendered Content." *ACM Transactions on Graphics (TOG)*.

Briedis, Karlis Martins, Abdelaziz Djelouah, Raphaël Ortiz, Mark Meyer, Markus Gross, and Christopher Schroers (2023). "Kernel-Based Frame Interpolation for Spatio-Temporally Adaptive Rendering." *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Brown, Duane (1996). "Decentering Distortion of Lenses." *Photogrammetric engineering*.

Buttafava, Mauro, Jessica Zeman, Alberto Tosi, Kevin Eliceiri, and Andreas Velten (2015). "Non-line-of-sight Imaging Using a Time-gated Single Photon Avalanche Diode." *Optics express*.

Casas, Dan and Marc Comino Trinidad (2023). "SMPLitex: A Generative Model and Dataset for 3D Human Texture Estimation from Single Image." *arXiv preprint arXiv:2309.01855*.

Chabra, Rohan, Julian Straub, Christopher Sweeney, Richard Newcombe, and Henry Fuchs (2019). "StereoDRNet: Dilated Residual Stereonet." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chambolle, Antonin and Thomas Pock (2011). “A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging.” *Journal of mathematical imaging and vision*.

Chang, Jia-Ren and Yong-Sheng Chen (2018). “Pyramid Stereo Matching Network.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chang, Tianyu, Xun Yang, Tianzhu Zhang, and Meng Wang (2023). “Domain Generalized Stereo Matching via Hierarchical Visual Transformation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Charbon, Edoardo, Matt Fishburn, Richard Walker, Robert K Henderson, and Cristiano Niclass (2013). “SPAD-Based Sensors.” *TOF range-imaging cameras*.

Chen, Kanglin and Dirk A. Lorenz (2011). “Image Sequence Interpolation Using Optimal Control.” *Journal of Mathematical Imaging and Vision*.

Chen, Liyan, Weihan Wang, and Philippos Mordohai (2023). “Learning the Distribution of Errors in Stereo Matching for Joint Disparity and Uncertainty Estimation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Wenzheng, Fangyin Wei, Kiriakos N. Kutulakos, Szymon Rusinkiewicz, and Felix Heide (2020). “Learned Feature Embeddings for Non-line-of-sight Imaging and Recognition.” *ACM Transactions on Graphics (TOG)*.

Chen, Zhiqi, Ran Wang, Haojie Liu, and Yao Wang (2021). “PDWN: Pyramid Deformable Warping Network for Video Interpolation.” *IEEE Open Journal of Signal Processing*.

Cheng, Xianhang and Zhenzhong Chen (2020). “Video Frame Interpolation via Deformable Separable Convolution.” *AAAI Conference on Artificial Intelligence*.

Cheng, Xianhang and Zhenzhong Chen (2021). “Multiple Video Frame Interpolation via Enhanced Deformable Separable Convolution.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Chi, Zhixiang, Rasoul Mohammadi Nasiri, Zheng Liu, Yuanhao Yu, Juwei Lu, Jin Tang, and Konstantinos N. Plataniotis (2022). “Error-Aware Spatial Ensembles for Video Frame Interpolation.” *arXiv preprint arXiv:2207.12305*.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” *arXiv preprint arXiv:1406.1078*.

Choi, Jinsoo, Jaesik Park, and In So Kweon (2021). “High-Quality Frame Interpolation via Tridirectional Inference.” *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Choi, Kiseok, Inchul Kim, Dongyoung Choi, Julio Marco, Diego Gutierrez, and Min H. Kim (2023). "Self-Calibrating, Fully Differentiable NLOS Inverse Rendering." *SIGGRAPH Asia 2023 Conference Papers*.

Choi, Myungsub, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee (2020). "Channel Attention Is All You Need for Video Frame Interpolation." *AAAI Conference on Artificial Intelligence*.

Collet, Alvaro, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan (2015). "High-Quality Streamable Free-Viewpoint Video." *ACM Transactions on Graphics (TOG)*.

Dai, Jifeng, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei (2017). "Deformable Convolutional Networks." *IEEE International Conference on Computer Vision (ICCV)*.

Danier, Duolikun, Fan Zhang, and David Bull (2022a). "Enhancing Deformable Convolution based Video Frame Interpolation with Coarse-to-fine 3D CNN." *arXiv preprint arXiv:2202.07731*.

Danier, Duolikun, Fan Zhang, and David Bull (2022b). "ST-MFNet: A Spatio-Temporal Multi-Flow Network for Frame Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deitke, Matt, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi (2023). "Objaverse-XL: A Universe of 10M+ 3D Objects." *arXiv preprint arXiv:2307.05663*.

Dong, Jiong, Kaoru Ota, and Mianxiong Dong (2023). "Video Frame Interpolation: A Comprehensive Survey." *ACM Transactions on Multimedia Computing, Communications and Applications*.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2020). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." *arXiv preprint arXiv:2010.11929*.

Dosovitskiy, Alexey, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox (2015). "FlowNet: Learning Optical Flow With Convolutional Networks." *IEEE International Conference on Computer Vision (ICCV)*.

Dröge, Hannah, Yuval Bahat, Felix Heide, and Michael Möller (2022). “Explorable Data Consistent CT Reconstruction.” *British Machine Vision Conference (BMVC)*.

Dutta, Saikat, Arulkumar Subramaniam, and Anurag Mittal (2022). “Non-linear Motion Estimation for Video Frame Interpolation using Space-time Convolutions.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Egnal, Geoffrey, Max Mintz, and Richard P. Wildes (2004). “A Stereo Confidence Metric Using Single View Imagery with Comparison to Five Alternative Approaches.” *Image and vision computing*.

Egnal, Geoffrey and Richard P. Wildes (2002). “Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Faccio, Daniele, Andreas Velten, and Gordon Wetzstein (2020). “Non-Line-of-Sight Imaging.” *Nature Reviews Physics*.

Fanello, Sean Ryan, Christoph Rhemann, Vladimir Tankovich, Adarsh Kowdle, Sergio Orts Escolano, David Kim, and Shahram Izadi (2016). “HyperDepth: Learning Depth from Structured Light Without Matching.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fienup, James R. (1982). “Phase Retrieval Algorithms: A Comparison.” *Applied optics*.

Fujimura, Yuki, Takahiro Kushida, Takuya Funatomi, and Yasuhiro Mukaigawa (2023). “NLOS-NeuS: Non-line-of-sight Neural Implicit Surface.” *IEEE International Conference on Computer Vision (ICCV)*.

Galindo, Miguel, Julio Marco, Matthew O’Toole, Gordon Wetzstein, Diego Gutierrez, and Adrian Jarabo (2019). *A Dataset for Benchmarking Time-resolved Non-line-of-sight Imaging*.

Gariepy, Genevieve, Nikola Krstajic, Robert Henderson, Chunyong Li, Robert R. Thomson, Gerald S. Buller, Barmak Heshmat, Ramesh Raskar, Jonathan Leach, and Daniele Faccio (2015). “Single-Photon Sensitive Light-in-Fight Imaging.” *Nature Communications*.

Gariepy, Genevieve, Francesco Tonolini, Robert Henderson, Jonathan Leach, and Daniele Faccio (2016). “Detection and Tracking of Moving Objects Hidden from View.” *Nature Photonics*.

Garrepalli, Risheek, Jisoo Jeong, Rajeswaran C. Ravindran, Jamie Menjay Lin, and Fatih Porikli (2023). “DIFT: Dynamic Iterative Field Transforms for Memory Efficient Optical Flow.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gawlikowski, Jakob, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. (2023). "A Survey of Uncertainty in Deep Neural Networks." *Artificial Intelligence Review*.

Ge, Rong, Furong Huang, Chi Jin, and Yang Yuan (2015). "Escaping from Saddle Points – Online Stochastic Gradient for Tensor Decomposition." *Conference on learning theory*.

Gerchberg, R. W. and W. O. Saxton (1972). "A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures." *Optik*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets." *Advances in Neural Information Processing Systems (NeurIPS)*.

Grau, Javier, Markus Plack, Patrick Haehn, Michael Weinmann, and Matthias B. Hullin (2022). "Occlusion Fields: An Implicit Representation for Non-Line-of-Sight Surface Reconstruction." *arXiv preprint arXiv:2203.08657*. DOI: 10.48550/arXiv.2203.08657.

Grau Chopite, Javier, Matthias B. Hullin, Michael Wand, and Julian Iseringhausen (2020). "Deep Non-Line-Of-Sight Reconstruction." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gu, Xiaodong, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan (2020). "Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gui, Shurui, Chaoyue Wang, Qihua Chen, and Dacheng Tao (2020). "FeatureFlow: Robust Video Interpolation via Structure-to-Texture Generation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gul, Muhammad Shahzeb Khan, Michel Bätz, and Joachim Keinert (2019). "Pixel-Wise Confidences for Stereo Disparities Using Recurrent Neural Networks." *British Machine Vision Conference (BMVC)*.

Guo, Jie, Xihao Fu, Liqiang Lin, Hengjun Ma, Yanwen Guo, Shiqiu Liu, and Ling-Qi Yan (2021). "ExtraNet: Real-Time Extrapolated Rendering for Low-Latency Temporal Supersampling." *ACM Transactions on Graphics (TOG)*.

Guo, Kaiwen, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. (2019a). "The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting." *ACM Transactions on Graphics (TOG)*.

Guo, Xianda, Juntao Lu, Chenming Zhang, Yiqi Wang, Yiqun Duan, Tian Yang, Zheng Zhu, and Long Chen (2023). "OpenStereo: A Comprehensive Benchmark for Stereo Matching and Strong Baseline." *arXiv preprint arXiv:2312.00343*.

Guo, Xiaoyang, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li (2019b). "Group-Wise Correlation Stereo Network." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gupta, Otkrist, Thomas Willwacher, Andreas Velten, Ashok Veeraraghavan, and Ramesh Raskar (2012). "Reconstruction of Hidden 3D Shapes Using Diffuse Reflections." *Optics express*.

Haeusler, Ralf and Reinhard Klette (2012). "Evaluation of Stereo Confidence Measures on Synthetic and Recorded Image Data." *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*.

Haeusler, Ralf, Rahul Nair, and Daniel Kondermann (2013). "Ensemble Learning for Confidence Measures in Stereo Vision." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hamzah, Rostam Affendi and Haidi Ibrahim (2016). "Literature Survey on Stereo Vision Disparity Map Algorithms." *Journal of Sensors*.

Han, Kai, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. (2022). "A Survey on Vision Transformer." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heagerty, Jonathan, Sida Li, Eric Lee, Shuvra Bhattacharyya, Sujal Bista, Barbara Brawn, Brandon Y. Feng, Susmija Jabbireddy, Joseph JaJa, Hernisa Kacorri, et al. (2024). "HoloCamera: Advanced Volumetric Capture for Cinematic-Quality VR Applications." *IEEE Transactions on Visualization and Computer Graphics (TVCG)*.

Heide, Felix, Matthias B. Hullin, James Gregson, and Wolfgang Heidrich (2013). "Low-Budget Transient Imaging using Photonic Mixer Devices." *ACM Transactions on Graphics (TOG)*.

Heide, Felix, Matthew O'Toole, Kai Zang, David B. Lindell, Steven Diamond, and Gordon Wetzstein (2019). "Non-Line-of-Sight Imaging with Partial Occluders and Surface Normals." *ACM Transactions on Graphics (TOG)*.

Heide, Felix, Lei Xiao, Wolfgang Heidrich, and Matthias B Hullin (2014). "Diffuse Mirrors: 3D Reconstruction from Diffuse Indirect Illumination Using Inexpensive Time-of-flight Sensors." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Herbst, Evan, Steve Seitz, and Simon Baker (2009). “Occlusion Reasoning for temporal interpolation using optical flow.” *Department of Computer Science and Engineering, University of Washington, Tech. Rep. UW-CSE-09-08-01*.

Hernandez, Quercus, Diego Gutierrez, and Adrian Jarabo (2017). “A Computational Model of a Single-Photon Avalanche Diode Sensor for Transient Imaging.” *arXiv preprint arXiv:1703.02635*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long Short-Term Memory.” *Neural computation*.

Horn, Berthold K. P. and Brian G. Schunck (1981). “Determining Optical Flow.” *Artificial intelligence*.

Hou, Qiqi, Abhijay Ghildyal, and Feng Liu (2022). “A Perceptual Quality Metric for Video Frame Interpolation.” *European Conference on Computer Vision (ECCV)*.

Hu, Ping, Simon Niklaus, Stan Sclaroff, and Kate Saenko (2022). “Many-to-many Splatting for Efficient Video Frame Interpolation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, Xiaoyan and Philippos Mordohai (2012). “A Quantitative Evaluation of Confidence Measures for Stereo Vision.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Huang, Zhewei, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou (2021). “RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation.” *arXiv preprint arXiv:2011.06294*.

Hüllermeier, Eyke and Willem Waegeman (2021). “Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods.” *Machine learning*.

Huo, Yuchi and Sung-eui Yoon (2021). “A Survey on Deep Learning-based Monte Carlo Denoising.” *Computational Visual Media*.

Iseringhausen, Julian and Matthias B Hullin (2020). “Non-Line-of-Sight Reconstruction Using Efficient Transient Rendering.” *ACM Transactions on Graphics (TOG)*.

Isogawa, Mariko, Dorian Chan, Ye Yuan, Kris Kitani, and Matthew O’Toole (2020). “Efficient Non-Line-of-Sight Imaging from Transient Sinograms.” *European Conference on Computer Vision (ECCV)*.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros (2017). “Image-to-Image Translation with Conditional Adversarial Networks.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jakob, Wenzel, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang (2022). "Mitsuba 3 Renderer." Version 3.1.1. https://mitsuba-renderer.org.

Jarabo, A., B. Masia, J. Marco, and D. Gutierrez (2016). "Recent Advances in Transient Imaging: A Computer Graphics and Vision Perspective." *arXiv preprint arXiv:1611.00939*.

Jarabo, Adrian, Julio Marco, Adolfo Munoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez (2014). "A Framework for Transient Rendering." *ACM Transactions on Graphics (TOG)*.

Jiang, Huaizu, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz (2018). "Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin, Xin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm (2023a). "A Unified Pyramid Recurrent Network for Video Frame Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin, Xin, Longhai Wu, Guotao Shen, Youxin Chen, Jie Chen, Jayoon Koo, and Cheul-hee Hahm (2023b). "Enhanced Bi-Directional Motion Estimation for Video Frame Interpolation." *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Kadambi, Achuta, Hang Zhao, Boxin Shi, and Ramesh Raskar (2016). "Occluded Imaging with Time-of-Flight Sensors." *ACM Transactions on Graphics (TOG)*.

Kalantari, Nima Khademi, Ting-Chun Wang, and Ravi Ramamoorthi (2016). "Learning-Based View Synthesis for Light Field Cameras." *ACM Transactions on Graphics (TOG)*.

Kalluri, Tarun, Deepak Pathak, Manmohan Chandraker, and Du Tran (2023). "FLAVR: Flow-agnostic Video Representations for Fast Frame Interpolation." *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Kappeler, Armin, Sushobhan Ghosh, Jason Holloway, Oliver Cossairt, and Aggelos Katsaggelos (2017). "Ptychnet: CNN based Fourier Ptychography." *IEEE International Conference on Image Processing (ICIP)*.

Kato, Hiroharu, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon (2020). "Differentiable Rendering: A Survey." *arXiv preprint arXiv:2006.12057*.

Kendall, Alex, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry (2017). "End-to-End Learning of Geometry and Context for Deep Stereo Regression." *IEEE International Conference on Computer Vision (ICCV)*.

Khamis, Sameh, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi (2018). "StereoNet: Guided Hierarchical Refinement for Real-Time Edge-Aware Depth Prediction." *European Conference on Computer Vision (ECCV)*.

Kiefhaber, Simon, Simon Niklaus, Feng Liu, and Simone Schaub-Meyer (2024). "Benchmarking Video Frame Interpolation." *arXiv preprint arXiv:2403.17128*.

Kingma, Diederik P. and Jimmy Ba (2014). "ADAM: A Method for Stochastic Optimization." *arXiv preprint arXiv:1412.6980*.

Kirmani, Ahmed, Tyler Hutchison, James Davis, and Ramesh Raskar (2009). "Looking Around the Corner Using Transient Imaging." *IEEE International Conference on Computer Vision (ICCV)*.

Klein, Jonathan, Martin Laurenzis, Matthias B. Hullin, and Julian Iseringhausen (2020). "A Calibration Scheme for Non-Line-of-Sight Imaging Setups." *Optics Express*.

Klein, Jonathan, Christoph Peters, Jaime Martín, Martin Laurenzis, and Matthias B Hullin (2016). "Tracking Objects Outside the Line of Sight Using 2D Intensity Images." *Scientific reports*.

Kolb, Andreas, Erhardt Barth, and Reinhard Koch (2008). "ToF-sensors: New Dimensions for Realism and Interactivity." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Kong, Lingtong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang (2022). "IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation." *arXiv preprint arXiv:2205.14620*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2017). "Imagenet Classification with Deep Convolutional Neural Networks." *Communications of the ACM*.

Kukačka, Jan, Vladimir Golkov, and Daniel Cremers (2017). "Regularization for Deep Learning: A Taxonomy." *arXiv preprint arXiv:1710.10686*.

Kuznetsov, Alexandr, Nima Khademi Kalantari, and Ravi Ramamoorthi (2018). "Deep Adaptive Sampling for Low Sample Count Rendering." *Computer Graphics Forum (CGF)*.

Laga, Hamid, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun (2020). "A Survey on Deep Learning Techniques for Stereo-Based Depth Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Laurentini, Aldo (1994). "The Visual Hull Concept for Silhouette-Based Image Understanding." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Lee, Hyeongmin, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee (2020). "AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lee, Sungho, Narae Choi, and Woong Il Choi (2022). "Enhanced Correlation Matching based Video Frame Interpolation." *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

LeMoigne, Jacqueline and Allen Mark Waxman (1988). "Structured Light Patterns for Robot Mobility." *IEEE Journal on Robotics and Automation*.

Li, Jiankun, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu (2022). "Practical Stereo Matching via Cascaded Recurrent Network with Adaptive Correlation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Tzu-Mao, Miika Aittala, Frédo Durand, and Jaakko Lehtinen (2018). "Differentiable Monte Carlo Ray Tracing through Edge Sampling." *ACM Transactions on Graphics (TOG)*.

Li, Yue, Jiayong Peng, Juntian Ye, Yueyi Zhang, Feihu Xu, and Zhiwei Xiong (2023). "NLOST: Non-Line-of-Sight Imaging with Transformer." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Zhaoshuo, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath (2021). "Revisiting Stereo Depth Estimation from a Sequence-to-Sequence Perspective with Transformers." *IEEE International Conference on Computer Vision (ICCV)*.

Liang, Jingyun, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool (2022a). "VRT: A Video Restoration Transformer." *arXiv preprint arXiv:2201.12288*.

Liang, Jingyun, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool (2022b). "Recurrent Video Restoration Transformer with Guided Deformable Attention." *Advances in Neural Information Processing Systems (NeurIPS)*.

Liang, Zhengfa, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang (2018). "Learning for Disparity Estimation Through Feature Constancy." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lindell, David B., Gordon Wetzstein, and Vladlen Koltun (2019a). "Acoustic Non-Line-of-Sight Imaging." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lindell, David B., Gordon Wetzstein, and Matthew O'Toole (2019b). "Wave-based Non-Line-of-Sight Imaging Using Fast fk Figration." *ACM Transactions on Graphics (TOG)*.

Lipson, Lahav, Zachary Teed, and Jia Deng (2021). "RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching." *International Conference on 3D Vision (3DV)*.

Liu, Chengxu, Huan Yang, Jianlong Fu, and Xueming Qian (2022a). "TTVFI: Learning Trajectory-Aware Transformer for Video Frame Interpolation." *arXiv preprint arXiv:2207.09048*.

Liu, Jinfeng, Lingtong Kong, and Jie Yang (2022b). "ATCA: An Arc Trajectory Based Model with Curvature Attention for Video Frame Interpolation." *IEEE International Conference on Image Processing (ICIP)*.

Liu, Meiqin, Chenming Xu, Chao Yao, Chunyu Lin, and Yao Zhao (2022c). "JNMR: Joint Non-linear Motion Regression for Video Frame Interpolation." *arXiv preprint arXiv:2206.04231*.

Liu, Xiaochun, Sebastian Bauer, and Andreas Velten (2019a). "Analysis of Feature Visibility in Non-Line-of-Sight Measurements." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Xiaochun, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten (2019b). "Non-Line-of-Sight Imaging Using Phasor-Field Virtual Wave Optics." *Nature*.

Liu, Yihao, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong (2020a). "Enhanced Quadratic Video Interpolation." *European Conference on Computer Vision (ECCV)*.

Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (2021). "Swin transformer: Hierarchical Vision Transformer using Shifted Windows." *IEEE International Conference on Computer Vision (ICCV)*.

Liu, Zhouyong, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Shilei Sun, Chunguo Li, and Luxi Yang (2020b). "ConvTransformer: A Convolutional Transformer Network for Video Frame Synthesis." *arXiv preprint arXiv:2011.10185*.

Long, Gucan, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu (2016). "Learning Image Matching by Simply Watching Video." *European Conference on Computer Vision (ECCV)*.

Loop, Charles and Zhengyou Zhang (1999). "Computing Rectifying Homographies for Stereo Vision." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Loper, Matthew, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black (2015). "SMPL: A Skinned Multi-Person Linear Model." *ACM Transactions on Graphics (TOG)*.

Lorensen, William E. and Harvey E. Cline (1987). "Marching Cubes: A high Resolution 3D Surface Construction Algorithm." *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Loshchilov, Ilya and Frank Hutter (2017). "Decoupled Weight Decay Regularization." *arXiv preprint arXiv:1711.05101*.

Lu, Chuanhua, Hideaki Uchiyama, Diego Thomas, Atsushi Shimada, and Rin-ichiro Taniguchi (2018). "Sparse Cost Volume for Efficient Stereo Matching." *Remote sensing*.

Lu, Liying, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia (2022). "Video Frame Interpolation with Transformer." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lucas, Bruce D. and Takeo Kanade (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision." *IJCAI'81: 7th international joint conference on Artificial intelligence*.

Luo, Wenjie, Alexander G. Schwing, and Raquel Urtasun (2016). "Efficient Deep Learning for Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mao, Xudong, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley (2017). "Least Squares Generative Adversarial Networks." *IEEE International Conference on Computer Vision (ICCV)*.

Marco, Julio, Wojciech Jarosz, Diego Gutierrez, and Adrian Jarabo (2017). "Transient Photon Beams." *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Martinez, Manuel and Rainer Stiefelhagen (2013). "Kinect Unleashed: Getting Control over High Resolution Depth Maps." *MVA*.

Mayer, Nikolaus, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (2016). "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mehl, Lukas, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn (2023). "Spring: A High-Resolution High-Detail Dataset and Benchmark for Scene Flow, Optical Flow and Stereo." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mehltretter, Max and Christian Heipke (2019). "CNN-based Cost Volume Analysis as Confidence Measure for Dense Matching." *Proceedings of the IEEE/CVF international conference on computer vision workshops*.

Menze, Moritz and Andreas Geiger (2015). “Object Scene Flow for Autonomous Vehicles.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mescheder, Lars, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger (2019). “Occupancy Networks: Learning 3d Reconstruction in Function Space.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Metzler, Christopher, Phillip Schniter, Ashok Veeraraghavan, et al. (2018). “prDeep: Robust Phase Retrieval with a Flexible Deep Network.” *International Conference on Machine Learning (ICML)*.

Metzler, Christopher A., Felix Heide, Prasana Rangarajan, Muralidhar Madabhushi Balaji, Aparna Viswanath, Ashok Veeraraghavan, and Richard G. Baraniuk (2020). “Deep-Inverse Correlography: Towards Real-Time High-Resolution Non-Line-of-Sight Imaging.” *Optica*.

Meyer, Simone, Victor Cornillère, Abdelaziz Djelouah, Christopher Schroers, and Markus Gross (2018a). “Deep Video Color Propagation.” *British Machine Vision Conference (BMVC)*.

Meyer, Simone, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers (2018b). “PhaseNet for Video Frame Interpolation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Meyer, Simone, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung (2015). “Phase-based Frame Interpolation for Video.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Mordohai, Philippos (2009). “The Self-Aware Matching Measure for Stereo.” *IEEE International Conference on Computer Vision (ICCV)*.

Mu, Fangzhou, Sicheng Mo, Jiayong Peng, Xiaochun Liu, Ji Hyun Nam, Siddeshwar Raghavan, Andreas Velten, and Yin Li (2022). “Physics to the Rescue: Deep Non-line-of-sight Reconstruction for High-speed Imaging.” *arXiv preprint arXiv:2205.01679*.

Mühlmann, Karsten, Dennis Maier, Jürgen Hesser, and Reinhard Männer (2002). “Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation.” *International Journal of Computer Vision*.

Nam, Ji Hyun, Eric Brandt, Sebastian Bauer, Xiaochun Liu, Marco Renna, Alberto Tosi, Eftychios Sifakis, and Andreas Velten (2021). “Low-Latency Time-of-Flight Non-Line-of-Sight Imaging at 5 Frames per Second.” *Nature communications*.

Nie, Guang-Yu, Ming-Ming Cheng, Yun Liu, Zhengfa Liang, Deng-Ping Fan, Yue Liu, and Yongtian Wang (2019). “Multi-Level Context Ultra-Aggregation for Stereo Matching.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Niklaus, Simon, Ping Hu, and Jiawen Chen (2023). “Splatting-Based Synthesis for Video Frame Interpolation.” *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Niklaus, Simon and Feng Liu (2018). “Context-aware Synthesis for Video Frame Interpolation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Niklaus, Simon and Feng Liu (2020). “Softmax Splatting for Video Frame Interpolation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Niklaus, Simon, Long Mai, and Feng Liu (2017a). “Video Frame Interpolation via Adaptive Convolution.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Niklaus, Simon, Long Mai, and Feng Liu (2017b). “Video Frame Interpolation via Adaptive Separable Convolution.” *IEEE International Conference on Computer Vision (ICCV)*.

Niklaus, Simon, Long Mai, and Oliver Wang (2021). “Revisiting Adaptive Convolutions for Video Frame Interpolation.” *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

O’Toole, Matthew, David B. Lindell, and Gordon Wetzstein (2018). “Confocal Non-Line-of-Sight Imaging Based on the Light-Cone Transform.” *Nature*.

Orts-Escolano, Sergio, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, et al. (2016). “Holoportation: Virtual 3D Teleportation in Real-Time.” *Proceedings of the 29th annual symposium on user interface software and technology*.

Pang, Jiahao, Wenxiu Sun, Jimmy S. J. Ren, Chengxi Yang, and Qiong Yan (2017). “Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching.” *Proceedings of the IEEE international conference on computer vision workshops*.

Parikh, Neal and Stephen Boyd (2014). “Proximal Algorithms.” *Foundations and Trends® in Optimization*.

Park, Jeong Joon, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove (2019). “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Park, Junheum, Keunsoo Ko, Chul Lee, and Chang-Su Kim (2020). "BMBC: Bilateral Motion Estimation with Bilateral Cost Volume for Video Interpolation." *European Conference on Computer Vision (ECCV)*.

Park, Junheum, Chul Lee, and Chang-Su Kim (2021). "Asymmetric Bilateral Motion Estimation for Video Frame Interpolation." *IEEE International Conference on Computer Vision (ICCV)*.

Patney, Anjul and Aaron Lefohn (2018). "Detecting Aliasing Artifacts in Image Sequences Using Deep Neural Networks." *Proceedings of the Conference on High-Performance Graphics*.

Pediredla, Adithya Kumar, Mauro Buttafava, Alberto Tosi, Oliver Cossairt, and Ashok Veeraraghavan (2017). "Reconstructing Rooms Using Photon Echoes: A Plane Based Model and Reconstruction Algorithm for Looking Around the Corner." *2017 IEEE International Conference on Computational Photography (ICCP)*.

Peleg, Tomer, Pablo Szekely, Doron Sabo, and Omry Sendik (2019). "IM-Net for High Resolution Video Frame Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng, Songyou, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger (2020). "Convolutional Occupancy Networks." *European Conference on Computer Vision (ECCV)*.

Perazzi, Federico, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung (2016). "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pfister, Hanspeter, Matthias Zwicker, Jeroen Van Baar, and Markus Gross (2000). "Surfels: Surface Elements as Rendering Primitives." *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Pharr, Matt, Wenzel Jakob, and Greg Humphreys (2023). *Physically Based Rendering: From Theory to Implementation*.

Plack, Markus, Karlis Martins Briedis, Abdelaziz Djelouah, Matthias B. Hullin, Markus Gross, and Christopher Schroers (2023a). "Frame Interpolation Transformer and Uncertainty Guidance." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: `10.1109/CVPR52729.2023.00946`.

Plack, Markus, Clara Callenberg, Monika Schneider, and Matthias B. Hullin (2023b). "Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction." *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. DOI: `10.1109/WACV56688.2023.00308`.

Plack, Markus, Hannah Dröge, Leif Van Holland, and Matthias B. Hullin (2024). "VHS: High-Resolution Iterative Stereo Matching with Visual Hull Priors." *arXiv preprint arXiv:2406.02552*. DOI: 10.48550/arXiv.2406.02552.

Poggi, Matteo, Seungryong Kim, Fabio Tosi, Sunok Kim, Filippo Aleotti, Dongbo Min, Kwanghoon Sohn, and Stefano Mattoccia (2021). "On the Confidence of Stereo Matching in a Deep-Learning Era: A Quantitative Evaluation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Poggi, Matteo, Fabio Tosi, and Stefano Mattoccia (2017). "Quantitative Evaluation of Confidence Measures in a Machine Learning World." *IEEE International Conference on Computer Vision (ICCV)*.

Raditya, Carolus, Muhammad Rizky, Sergio Mayranio, and Benfano Soewito (2021). "The Effectivity of Color for Chroma-Key Techniques." *Procedia Computer Science*.

Raj, Thinal, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain (2020). "A Survey on LiDAR Scanning Mechanisms." *Electronics*.

Rakêt, Lars Lau, Lars Roholm, Andrés Bruhn, and Joachim Weickert (2012). "Motion Compensated Frame Interpolation with a Symmetric Optical Flow Constraint." *International Symposium on Visual Computing*.

Rao, Zhibo, Bangshu Xiong, Mingyi He, Yuchao Dai, Renjie He, Zhelun Shen, and Xing Li (2023). "Masked Representation Learning for Domain Generalized Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Reda, Fitsum, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless (2022). "FILM: Frame Interpolation for Large Motion." *European Conference on Computer Vision (ECCV)*.

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

Ruder, Sebastian (2017). "An Overview of Multi-Task Learning in Deep Neural Networks." *arXiv preprint arXiv:1706.05098*.

Rudin, Leonid I., Stanley Osher, and Emad Fatemi (1992). "Nonlinear Total Variation Based Noise Removal Algorithms." *Physica D: nonlinear phenomena*.

Sagi, Omer and Lior Rokach (2018). "Ensemble Learning: A Survey." *Wiley interdisciplinary reviews: data mining and knowledge discovery*.

Sanderson, Brandon (2017). *Oathbringer*.

Scharr, Hanno, Christoph Briese, Patrick Embgenbroich, Andreas Fischbach, Fabio Fiorani, and Mark Müller-Linow (2017). "Fast High Resolution Volume Carving for 3D Plant Shoot Reconstruction." *Frontiers in plant science*.

Scharstein, Daniel and Richard Szeliski (1998). "Stereo Matching with Nonlinear Diffusion." *International journal of computer vision*.

Scharstein, Daniel and Richard Szeliski (2002). "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms." *International journal of computer vision*.

Scharstein, Daniel and Richard Szeliski (2003). "High-Accuracy Stereo Depth Maps Using Structured Light." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schreiberhuber, Simon, Jean-Baptiste Weibel, Timothy Patten, and Markus Vincze (2022). "GigaDepth: Learning Depth from Structured Light with Branching Neural Networks." *European Conference on Computer Vision (ECCV)*.

Schroers, Christopher Richard, Karlis Martins Briedis, Abdelaziz Djelouah, Ian McGonigal, Mark Meyer, Marios Papas, and Markus Plack (2022). *Frame Interpolation for Rendered Content*. US Patent App. 17/325,026.

Seki, Akihito and Marc Pollefeys (2016). "Patch Based Confidence Prediction for Dense Disparity Map." *British Machine Vision Conference (BMVC)*.

Seki, Akihito and Marc Pollefeys (2017). "SGM-Nets: Semi-Global Matching with Neural Networks." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shaked, Amit and Lior Wolf (2017). "Improved Stereo Matching with Constant Highway Networks and Reflective Confidence Learning." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shamsafar, Faranak, Samuel Woerz, Rafia Rahim, and Andreas Zell (2022). "MobileStereoNet: Towards Lightweight Deep Networks for Stereo Matching." *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Shangguan, Wentao, Yu Sun, Weijie Gan, and Ulugbek S. Kamilov (2022). "Learning Cross-Video Neural Representations for High-Quality Frame Interpolation." *arXiv preprint arXiv:2203.00137*.

Shechtman, Yoav, Yonina C. Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev (2015). "Phase Retrieval with Application to Optical Imaging: A Contemporary Overview." *IEEE signal processing magazine*.

Shen, Siyuan, Zi Wang, Ping Liu, Zhengqing Pan, Ruiqian Li, Tian Gao, Shiying Li, and Jingyi Yu (2021a). "Non-Line-of-Sight Imaging via Neural Transient Fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Shen, Zhelun, Yuchao Dai, and Zhibo Rao (2021b). "CFNet: Cascade and Fused Cost Volume for Robust Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shi, Wenzhe, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang (2016). "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shi, Xiaoyu, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li (2023). "VideoFlow: Exploiting Temporal Cues for Multi-Frame Optical Flow Estimation." *IEEE International Conference on Computer Vision (ICCV)*.

Shi, Zhihao, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang (2022). "Video Frame Interpolation Transformer." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sim, Hyeonjun, Jihyong Oh, and Munchurl Kim (2021). "XVFI: Extreme Video Frame Interpolation." *IEEE International Conference on Computer Vision (ICCV)*.

Siyao, Li, Tianpei Gu, Weiye Xiao, Henghui Ding, Ziwei Liu, and Chen Change Loy (2023). "Deep Geometrized Cartoon Line Inbetweening." *IEEE International Conference on Computer Vision (ICCV)*.

Siyao, Li, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu (2021). "Deep Animation Video Interpolation in the Wild." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Slaney, Malcolm and Philip A. Chou (2014). *Time of Flight Tracer*. Technical report. Microsoft Research.

Smith, Adam, James Skorupski, and James Davis (2008). "Transient Rendering."

Smith, Brandon M., Matthew O'Toole, and Mohit Gupta (2018). "Tracking Multiple Objects Outside the Line of Sight Using Speckle Imaging." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah (2012). "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild." *arXiv preprint arXiv:1212.0402*.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *The journal of machine learning research*.

Stanimirović, Predrag S. and Marko B. Miladinović (2010). "Accelerated Gradient Descent Methods with Line Search." *Numerical Algorithms*.

Stathopoulou, Elisavet Konstantina and Fabio Remondino (2023). "A Survey on Conventional and Learning-Based Methods for Multi-View Stereo." *The Photogrammetric Record*.

Stich, Timo, Christian Linz, Georgia Albuquerque, and Marcus Magnor (2008). "View and Time Interpolation in Image Space." *Computer Graphics Forum (CGF)*.

Su, Shuochen, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang (2017). "Deep Video Deblurring for Hand-Held Cameras." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, Deqing, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (2018). "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna (2016). "Rethinking the Inception Architecture for Computer Vision." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Teed, Zachary and Jia Deng (2020). "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow." *European Conference on Computer Vision (ECCV)*.

Thompson, Neil C., Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso (2021). "Deep Learning's Diminishing Returns: The Cost of Improvement is Becoming Unsustainable." *IEEE Spectrum*.

Tonioni, Alessio, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano (2019). "Real-Time Self-Adaptive Deep Stereo." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tosi, Fabio, Matteo Poggi, Antonio Benincasa, and Stefano Mattoccia (2018). "Beyond Local Reasoning for Stereo Confidence Estimation with Deep Learning." *European Conference on Computer Vision (ECCV)*.

Tosi, Fabio, Alessio Tonioni, Daniele De Gregorio, and Matteo Poggi (2023). "NeRF-Supervised Deep Stereo." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Trinidad, Marc Comino, Ricardo Martin Brualla, Florian Kainz, and Janne Kontkanen (2019). "Multi-View Image Fusion." *IEEE International Conference on Computer Vision (ICCV)*.

Tsai, Chia-Yin, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas (2019). "Beyond Volumetric Albedo–A Surface Optimization Framework for Non-Line-Of-Sight Imaging." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Uelwer, Tobias, Tobias Hoffmann, and Stefan Harmeling (2021a). "Non-Iterative Phase Retrieval With Cascaded Neural Networks." *arXiv preprint arXiv:2106.10195*.

Uelwer, Tobias, Alexander Oberstraß, and Stefan Harmeling (2021b). "Phase Retrieval Using Conditional Generative Adversarial Networks." *2020 25th International Conference on Pattern Recognition (ICPR)*.

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

Veach, Eric (1998). *Robust Monte Carlo Methods for Light Transport Simulation*.

Velten, Andreas, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G. Bawendi, and Ramesh Raskar (2012). "Recovering Three-Dimensional Shape Around a Corner Using Ultrafast Time-of-Flight Imaging." *Nature communications*.

Velten, Andreas, Di Wu, Adrian Jarabo, Belen Masia, Christopher Barsi, Corcoy Joshi, Matthew Everett Lawson, Moungi G. Bawendi, Diego Gutierrez, and Ramesh Raskar (2013). "Femto-Photography: Capturing and Visualizing the Propagation of Light." *ACM Transactions on Graphics (TOG)*.

Vogels, Thijs, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák (2018). "Denoising with Kernel Prediction and Asymmetric Loss Functions." *ACM Transactions on Graphics (TOG)*.

Vuylsteke, Piet and André Oosterlinck (1990). "Range Image Acquisition with a Single Binary-Encoded Light Pattern." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Wald, Ingo, William R. Mark, Johannes Günther, Solomon Boulos, Thiago Ize, Warren Hunt, Steven G. Parker, and Peter Shirley (2009). "State of the Art in Ray Tracing Animated Scenes." *Computer Graphics Forum (CGF)*.

Wang, Bin, Ming-Yang Zheng, Jin-Jian Han, Xin Huang, Xiu-Ping Xie, Feihu Xu, Qiang Zhang, and Jian-Wei Pan (2021a). "Non-Line-of-Sight Imaging with Picosecond Temporal Resolution." *Physical Review Letters*.

Wang, Fangjinhua, Silvano Galliani, Christoph Vogel, and Marc Pollefeys (2022). "IterMVS: Iterative Probability Estimation for Efficient Multi-View Stereo." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Fangjinhua, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys (2021b). “PatchmatchNet: Learned Multi-View Patchmatch Stereo.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Hengli, Rui Fan, and Ming Liu (2021c). “SCV-Stereo: Learning Stereo Matching from a Sparse Cost Volume.” *IEEE International Conference on Image Processing (ICIP)*.

Wang, Qiang, Shaohuai Shi, Shizhen Zheng, Kaiyong Zhao, and Xiaowen Chu (2021d). “FADNet++: Real-Time and Accurate Disparity Estimation with Configurable Networks.” *arXiv preprint arXiv:2110.02582*.

Wang, Xintao, Kelvin C. K. Chan, Ke Yu, Chao Dong, and Chen Change Loy (2019). “EDVR: Video Restoration with Enhanced Deformable Convolutional Networks.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wedel, Andreas, Annemarie Meißner, Clemens Rabe, Uwe Franke, and Daniel Cremers (2009). “Detection and Segmentation of Independently Moving Objects from Dense Scene Flow.” *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*.

Weinzaepfel, Philippe, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud (2022). “CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion.” *Advances in Neural Information Processing Systems (NeurIPS)*.

Weinzaepfel, Philippe, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jerome Revaud (2023). “CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow.” *IEEE International Conference on Computer Vision (ICCV)*.

Weinzaepfel, Philippe, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid (2013). “DeepFlow: Large Displacement Optical Flow with Deep Matching.” *IEEE International Conference on Computer Vision (ICCV)*.

Werlberger, Manuel, Thomas Pock, Markus Unger, and Horst Bischof (2011). “Optical Flow Guided TV-L 1 Video Interpolation and Restoration.” *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*.

Wu, Cheng, Jianjiang Liu, Xin Huang, Zheng-Ping Li, Chao Yu, Jun-Tian Ye, Jun Zhang, Qiang Zhang, Xiankang Dou, Vivek K. Goyal, et al. (2021a). “Non-Line-of-Sight Imaging Over 1.43 km.” *Proceedings of the National Academy of Sciences*.

Wu, Lifan, Guangyan Cai, Ravi Ramamoorthi, and Shuang Zhao (2021b). “Differentiable Time-Gated Rendering.” *ACM Transactions on Graphics (TOG)*.

Xin, Shumian, Sotiris Nousias, Kiriakos N. Kutulakos, Aswin C. Sankaranarayanan, Srinivasa G. Narasimhan, and Ioannis Gkioulekas (2019). "A Theory of Fermat Paths for Non-Line-of-Sight Shape Reconstruction." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Gangwei, Junda Cheng, Peng Guo, and Xin Yang (2022). "Attention Concatenation Volume for Accurate and Efficient Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Gangwei, Xianqi Wang, Xiaohuan Ding, and Xin Yang (2023a). "Iterative Geometry Encoding Volume for Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Haofei, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger (2023b). "Unifying Flow, Stereo and Depth Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Xu, Haofei and Juyong Zhang (2020). "AANet: Adaptive Aggregation Network for Efficient Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Xiangyu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang (2019). "Quadratic Video Interpolation." *Advances in Neural Information Processing Systems (NeurIPS)*.

Xue, Tianfan, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman (2019). "Video Enhancement with Task-oriented Flow." *International Journal of Computer Vision*.

Yan, Lian, Robert H. Dodier, Michael Mozer, and Richard H. Wolniewicz (2003). "Optimizing Classifier Performance via an Approximation to the Wilcoxon-Mann-Whitney Statistic." *Proceedings of the 20th international conference on machine learning*.

Yang, Gengshan, Joshua Manela, Michael Happold, and Deva Ramanan (2019). "Hierarchical Deep Stereo Matching on High-Resolution Images." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, Kai-Chieh, Ai-Mei Huang, Truong Q. Nguyen, Clark C. Guest, and Pankaj K. Das (2008). "A New Objective Quality Metric for Frame Interpolation Used in Video Compression." *IEEE transactions on broadcasting*.

Yi, Shinyoung, Donggun Kim, Kiseok Choi, Adrian Jarabo, Diego Gutierrez, and Min H. Kim (2021). "Differentiable Transient Rendering." *ACM Transactions on Graphics (TOG)*.

Young, Sean I., David B. Lindell, Bernd Girod, David Taubman, and Gordon Wetzstein (2020). "Non-line-of-sight Surface Reconstruction Using the Directional Light-cone Transform." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zagoruyko, Sergey and Nikos Komodakis (2015). "Learning to Compare Image Patches via Convolutional Neural Networks." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zbontar, Jure and Yann LeCun (2015). "Computing the Stereo Matching Cost With a Convolutional Neural Network." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zeng, Kai, Yaonan Wang, Wei Wang, Hui Zhang, Jianxu Mao, and Qing Zhu (2023). "Deep Confidence Propagation Stereo Network." *IEEE Transactions on Intelligent Transportation Systems*.

Zhai, Mingliang, Xuezhi Xiang, Ning Lv, and Xiangdong Kong (2021). "Optical Flow and Scene Flow Estimation: A Survey." *Pattern Recognition*.

Zhang, Cheng, Bailey Miller, Kan Yan, Ioannis Gkioulekas, and Shuang Zhao (2020). "Path-Space Differentiable Rendering." *ACM Transactions on Graphics (TOG)*.

Zhang, Feihu, Victor Prisacariu, Ruigang Yang, and Philip HS Torr (2019). "GA-Net: Guided Aggregation Net For End-to-End Stereo Matching." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Guozhen, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang (2023). "Extracting Motion and Appearance via Inter-Frame Attention for Efficient Video Frame Interpolation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Richard, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Haoliang, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao (2023). "High-Frequency Stereo Matching Network." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Haoliang, Huizhou Zhou, Yongjun Zhang, Yong Zhao, Yitong Yang, and Ting Ouyang (2022). "EAI-Stereo: Error Aware Iterative Network for Stereo Matching." *Proceedings of the Asian Conference on Computer Vision*.

Zhou, Chang, Jie Liu, Jie Tang, and Gangshan Wu (2023a). "Video Frame Interpolation With Densely Queried Bilateral Correlation." *arXiv preprint arXiv:2304.13596*.

Zhou, Chao, Hong Zhang, Xiaoyong Shen, and Jiaya Jia (2017). "Unsupervised Learning of Stereo Matching." *IEEE International Conference on Computer Vision (ICCV)*.

Zhou, Kun, Wenbo Li, Xiaoguang Han, and Jiangbo Lu (2023b). “Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhuang, Weihao, Tristan Hascoet, Ryoichi Takashima, and Tetsuya Takiguchi (2022). “Optical Flow Regularization of Implicit Neural Representations for Video Frame Interpolation.” *arXiv preprint arXiv:2206.10886*.

Zimmer, Henning, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung (2015). “Path-space Motion Estimation and Decomposition for Robust Animation Filtering.” *Computer Graphics Forum (CGF)*.

Zitnick, C. Lawrence, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski (2004). “High-Quality Video View Interpolation Using a Layered Representation.” *ACM Transactions on Graphics (TOG)*.

Zwicker, Matthias, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross (2001). “Surface Splatting.” *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

List of Figures

1.1 Overview of Contributions 6

2.1 Epipolar Geometry 14
2.2 Stereo Rectification 15
2.3 Depth from Disparity 16
2.4 Optical Flow Example 17
2.5 Image Warping 18
2.6 Prototypical NLoS Measurement Setup 20
2.7 Transient Image Example 21

3.1 Frame Interpolation Overview 29

4.1 Stereo Matching Teaser 40
4.2 VHS Network Overview 44
4.3 Visual Hull Boundaries 45
4.4 Training Dataset Sample 47
4.5 Memory Efficient Training Scheme 48
4.6 Qualitative Stereo Matching Results 51
4.7 Mask Accuracy Influence on EPE 53
4.8 Memory and Runtime Statistics 54

List of Tables

4.1 Baseline Comparison on SceneFlow 50
4.2 Ablation of the Visual Hull Guidance 50
4.3 Comparison on Polyhaven and SMPL 52
4.4 Memory and Runtime Measurements of the Efficient Training 53

Part IV

Appendix

Publication:
“Frame Interpolation Transformer and Uncertainty Guidance”

Markus Plack, Karlis Martins Briedis, Abdelaziz Djelouah,
Matthias B. Hullin, Markus Gross, and Christopher Schroers

IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

2023

DOI: `10.1109/CVPR52729.2023.00946`

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.00946

Frame Interpolation Transformer and Uncertainty Guidance

Markus Plack[1]* Karlis Martins Briedis[2,3] Abdelaziz Djelouah[3]
Matthias B. Hullin[1] Markus Gross[2,3] Christopher Schroers[3]

[1]University of Bonn
Bonn, Germany

[2]Department of Computer Science
ETH Zürich, Switzerland

[3]DisneyResearch|Studios
Zürich, Switzerland

mplack@cs.uni-bonn.de, karlis.briedis@inf.ethz.ch

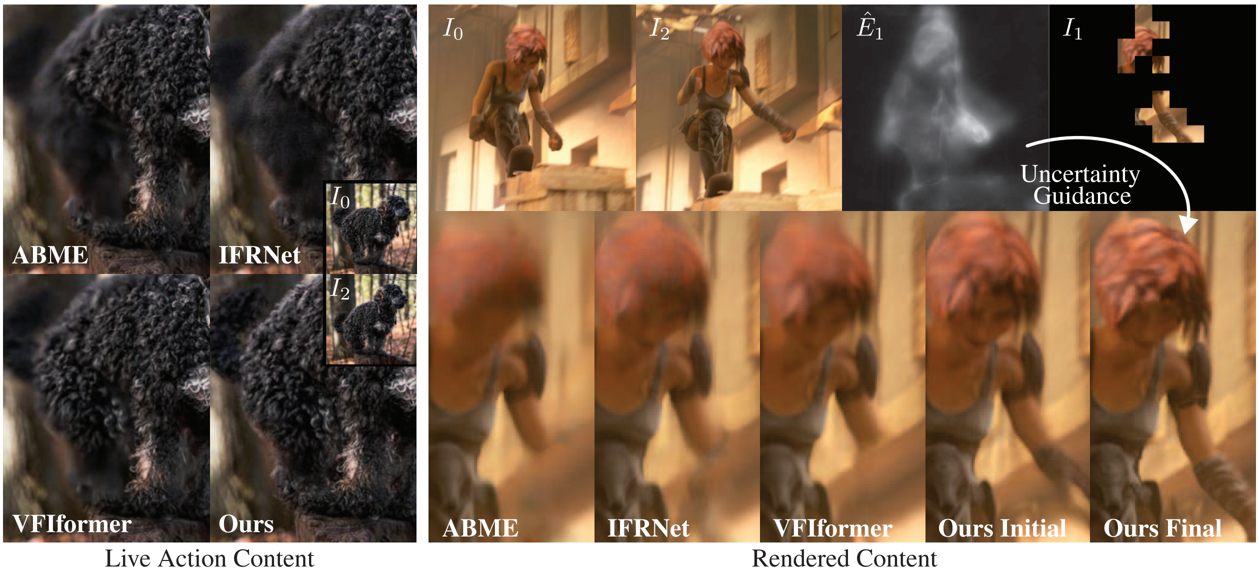


Figure 1. Our method achieves state-of-the-art results for frame interpolation. It produces sharp textures as highlighted on both live action (left) and rendered (right [15]) content. In addition to the interpolated frame, we estimate error maps that are helpful for quality checks in video production tools. More importantly, for rendered content it can be used to determine a subset of patches to render for the middle frame, which are then leveraged by our model to achieve production quality level results for a fraction of the rendering cost.

Abstract

Video frame interpolation has seen important progress in recent years, thanks to developments in several directions. Some works leverage better optical flow methods with improved splatting strategies or additional cues from depth, while others have investigated alternative approaches through direct predictions or transformers. Still, the problem remains unsolved in more challenging conditions such as complex lighting or large motion.

In this work, we are bridging the gap towards video production with a novel transformer-based interpolation network architecture capable of estimating the expected error together with the interpolated frame. This offers several advantages that are of key importance for frame interpolation usage: First, we obtained improved visual quality over several datasets. The improvement in terms of quality is also clearly demonstrated through a user study. Second, our method estimates error maps for the interpolated frame, which are essential for real-life applications on longer video sequences where problematic frames need to be flagged. Finally, for rendered content a partial rendering pass of the intermediate frame, guided by the predicted error, can be utilized during the interpolation to generate a new frame of superior quality. Through this error estimation, our method can produce even higher-quality intermediate frames using only a fraction of the time compared to a full rendering.

*Work done during an internship at DisneyResearch|Studios

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.00946

1. Introduction

Video frame interpolation (VFI) is a classical video processing problem where the aim is to restore an intermediate frame in a given video sequence. This temporal inbetweening enables many practical applications, such as video editing [38], novel-view synthesis [26], video retiming, and slow motion generation [25]. Recent advances in VFI methods [13,24,28,30,37,48,53,55] have been continuously improving the interpolation quality, but the problem remains open due to complex lighting effects and large motion that are ubiquitous in real-life videos and can introduce severe artifacts for the existing methods.

We propose a transformer-based VFI architecture that processes both source and target frames in a unified framework and compensates motion through a tightly integrated optical flow estimation and cross-backward warping. Our model improves over the current state-of-the-art as supported by our extensive quantitative experiments and a user study.

Besides the improvements in terms of results, our model also predicts the interpolation uncertainty similar to approaches for artifact detection [4, 49] and adaptive sampling [29, 60]. This is of key importance for usage in a production context, where working with long sequences requires a way to automatically identify problematic frames. Uncertainty estimation also benefits Computer Graphics (CG) applications, as we use it to determine which frame patches do not have sufficient quality and optionally mark them for rendering. Thanks to our novel transformer-based model, the rendered patches from the middle frame naturally fit in the same unified VFI framework, achieving high quality levels at the fraction of the cost of rendering the full middle frame. Our paradigm is more compatible with current production renderers than CG specialized VFI works [5, 21, 66] which require the generation of specific G-buffers for the keyframes and the intermediate frame.

In summary, our contributions are as follows.

- We introduce a novel motion-based VFI method, that treats input and target frames in the same manner through a transformer-based architecture using masks.
- Our model achieves state-of-the-art performance as shown both in quantitative experiments and a user study.
- We perform output's uncertainty estimation subtask, which can be particularly beneficial for rendered content to achieve even better quality results.

2. Related work

While classical approaches to frame interpolation relied on optical flow and image warping [2, 52, 62], they have been surpassed by learning-based methods. We start our discussion with a short review of *direct*, *phase* and *kernel* based prediction methods, before going into more details with approaches using *motion* or *transformers*.

Direct methods were proposed using purely convolutional architectures [27, 36] or combining channel attention with a deep residual network [13]. Alternatively, Meyer *et al*. [40] show a *phase-based method* based on the idea that phase-shifts can be used to represent motion, and later extended with a learning-based component [39].

Kernel-based methods, as originally introduced by Niklaus *et al*. [44], aim to predict kernels for all pixels that are applied in a convolutional layer. Offset prediction has been used [9, 30] to reduce the necessary kernel size to handle large motion, making those methods conceptually more similar to motion-based ones. Various other extensions have been proposed, including prediction of separable kernels [45, 46], time input for arbitrary frame interpolation [10], a multi-scale architecture including cost volumes [8], multi-stage networks [20], different backbones [16, 54], and improving performance [50].

Most *motion-based methods* build on the work of optical flow estimation methods [18, 57, 61]. Some methods use the estimated motion between the input frames to forward splat them [23, 42, 43], while others aim to find the flow from the intermediate frame to the reference frames, allowing for an easy backward warping, either by estimating the flows directly [24, 28, 47, 48, 53], through other means [3, 25, 31, 41, 55], or combine both forward and backward warping approaches [17]. While most methods assume linear motion between the keyframes, others estimate non-linear motion by using more than two input frames [12, 19, 33, 34, 63] or with a learned prior [48].

Various other approaches have been proposed to improve estimation of large motion by treating small and large motion with equal priority [53], dynamically adapting the flow estimation to the motion magnitude and image resolution [55], or better strategies for feature propagation [1]. We adopt equal motion treatment by extending the scale-agnostic feature extraction [53, 58]. Most recently, CG specific frame interpolation algorithms have been introduced for 2D animation [56] and 3D rendering [5].

Error estimation of the optical flow is used by Chi *et al*. [11] for specific treatment, proposing predefined fixed models for the various error levels. This is different from our method, that learns to predict perceptual and L_2-based error maps for final interpolation result.

With the introduction of the transformer [59] and its adaptation to vision tasks [22], several *transformer-based* frame interpolation approaches have been proposed. Liu *et al*. [35] use a transformer architecture that incorporates convolutions inside attention layers, but does not include any motion compensation. VFIformer [37] uses cross-scale

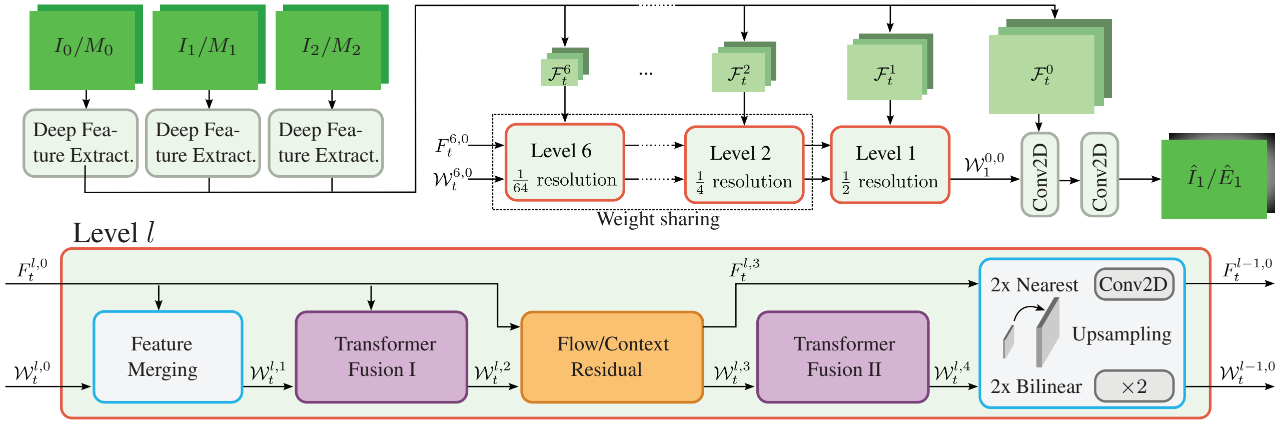


Figure 2. After extracting a feature pyramid $\{\mathcal{F}_t^l\}$ (**Deep Feature Extraction**) for each of the three frames (left) we pass a latent representation $\mathcal{W}_t$ along with a forward flow estimate F_t for each frame t through multiple levels of our reconstruction (center). At each level, after merging with the extracted features (**Feature Merging**), we update the latent representation using the initial flow estimate (**Transformer Fusion I**), followed by an update of the flow estimate and context vector from the new features (**Flow/Context Residual**) and another latent representation update using the new features and flows (**Transformer Fusion II**) before upsampling flow and features for the next level (**Upsampling**). Finally, we compute the interpolated Frame $\hat{I}_1$ and an estimate of the error $\hat{E}_1$ (top right).

window attention after warping the feature representations and TTVFI [32] uses an inconsistent region map inside a trajectory aware attention module. Both methods, however, cannot handle inputs of the middle frame and require an extra training of the upstream flow network, whereas our flow estimation is tightly integrated with the transformer fusion and trained end-to-end.

3. Method

The goal of our method is to interpolate two keyframes I_0, I_2 and find the intermediate frame $\hat{I}_1$ along with an estimate of the error $\hat{E}_1$. Subsequently, we analyze the error map and check if certain areas of the frame need to be rendered as we expect them to have insufficient quality. We then pass those additional masked inputs I_1 to the network along with the keyframes to get a final interpolated frame. Note that our method is well equipped to handle the common problem of two-frame interpolation without any changes to the architecture or training and that the additional inputs are entirely optional, *i.e.* we simply set $I_1 = 0$.

3.1. Interpolation network

Motivated by our goal to be able to handle arbitrary inputs, the overall architecture of our network is inspired by transformer architectures. This means that, opposed to common two-frame interpolation methods, there is little distinction within the network between the keyframes and the target frame. Instead, we equip each frame with a binary mask M_t indicating valid inputs to guide the interpolation. An overview of our method is given in Fig. 2.

We first extract a feature pyramid representation $\{\mathcal{F}_t^l\}_{l\in 0,\dots,6}$ for each of the inputs and process them in a coarse-to-fine manner with the same update blocks that share weights for the bottom 5 resolutions.

In each of the levels, we first merge the latent feature representations $\mathcal{W}_t^{l,i}$ with the respective input feature pyramid level. After that, they are updated in two *transformer fusion* blocks and a *flow/context residual* block in between that additionally updates the running flow estimates $F_t^{l,i}$, denoting the optical flow from t to $t+1$. Finally, the latent feature representations and flows are upsampled for processing in the next level.

In order to reduce the memory and compute costs, the processing of the topmost level is treated differently and consists of two convolutional layers.

Deep feature extraction. Our feature extraction is inspired by that of Reda et al. [53] to enable weight sharing on the lower levels of the reconstruction. We expand their idea by using a U-Net architecture instead of the original top-down approach. The reasoning behind this choice is that it more easily enables the network to capture semantically meaningful features on the upper levels of the pyramid without the need for many convolutional layers with large kernels or dilation.

First, we build image I_t^l and mask M_t^l pyramids, where image/mask l is downsampled by a factor of 2 to obtain level $l+1$. We concatenate both and pass them through a U-Net as illustrated in Fig. 3, keeping the last three layers as features. Finally, we concatenate all input and feature tensors of the same spatial resolution to build input feature pyramids $\{\mathcal{F}_t^l\}_{l\in 0,\dots,6}$ for $t \in \{0, 1, 2\}$. Note that all features from level two onward will be semantically similar

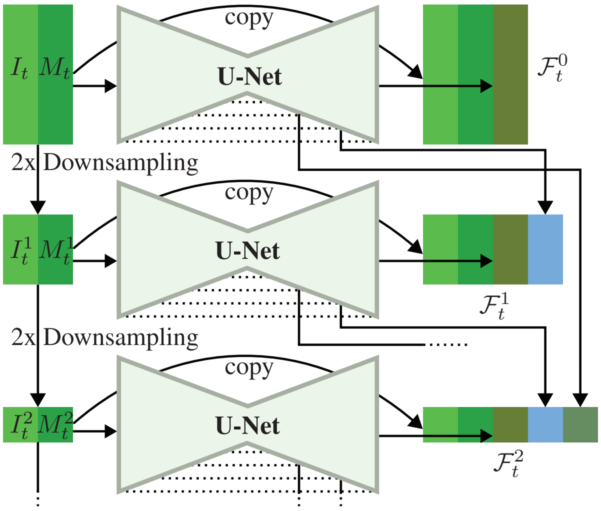


Figure 3. Illustration of our deep feature extraction module. The same U-Net is used to process the original inputs and all downsampled images/masks.

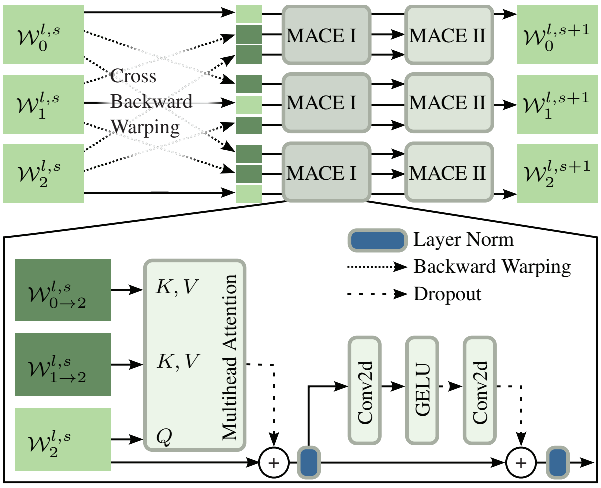


Figure 4. The transformer fusion module consists of two MACE blocks applied to all triplets after the cross backward warping.

and thus we can use weight sharing for all following modules on those levels.

Initialization and feature merging. On the lowest level we initialize the optical flows $F_t^{6,0}$ as 0 and set the latent feature representations $\mathcal{W}_t^{6,0}$ to a learned vector that is spatially repeated.

As the first step on each level, the upsampled pixelwise features of the previous level, or the initial values, $\mathcal{W}_t^{l,0} \in \mathbb{R}^{D_l}$ are merged with their respective feature pyramid features $\mathcal{F}_t^l \in \mathbb{R}^{C_l}$, where $C_0 := 52$, $C_1 := 148$, $C_{i\in\{2..6\}} := 340$, and $D_l := C_l + 15$. Therefore, we only merge the first C_l channels of $\mathcal{W}_t^{l,0}$ with $\mathcal{F}_t^l$ while keeping the remaining 15 channels unaffected:

$$\mathcal{W}_t^{l,1} = \begin{bmatrix} M_t^l \mathcal{F}_t^l + (1 - M_t^l)\,\left[\mathcal{W}_t^{l,0}\right]_{0..C_l-1} \\ \left[\mathcal{W}_t^{l,0}\right]_{C_l..D_l-1} \end{bmatrix} \tag{1}$$

The purpose of the directly passed through channels is similar to explicit occlusion maps employed by other methods, but we leave the choice on how to best use those additional channels to be learned by the network.

Transformer fusion. To update the latent feature representation of each frame $t_0 \in \{0, 1, 2\}$, we use cross-backward warping to align the features of all other frames $t_i \neq t_0$ by rescaling the current flow estimate at stage s as

$$\mathcal{W}_{t_i\to t_0}^{l,s}(x, y) = \mathcal{W}_{t_i}^{l,s}((t_0 - t_i)F_{t_i}^{l,s}(x, y)) \tag{2}$$

for spatial indices (x, y) and using bilinear interpolation for non-integer coordinates. We treat $\mathcal{W}_{t_0}^{l,s}(x, y)$, $\mathcal{W}_{t_1\to t_0}^{l,s}(x, y)$, and $\mathcal{W}_{t_2\to t_0}^{l,s}(x, y)$ as tokens processed by the multihead attention module. Specifically, for each head i the per-pixel query, key and value tensors are computed as

$$\begin{aligned} Q_i &= \mathbf{W}_i^Q \mathcal{W}_{t_0}^{l,s} && (3)\\ K_i &= \mathbf{W}_i^K \left[\mathcal{W}_{t_1\to t_0}^{l,s}, \mathcal{W}_{t_2\to t_0}^{l,s}\right] && (4)\\ V_i &= \mathbf{W}_i^V \left[\mathcal{W}_{t_1\to t_0}^{l,s}, \mathcal{W}_{t_2\to t_0}^{l,s}\right] && (5) \end{aligned}$$

and the softmax of the query/key multiplication and the residual update from the weighted sum of the values are computed as in the original transformer [59].

Since our latent feature representations have an inherent spatial structure, we opt to replace the linear layers of the standard transformer with convolutional residual layers. We use two convolutions with kernel size 3, a dropout layer before and after the second convolution and a GELU activation after the first. In addition, we use layer normalization after the multihead attention and the convolutional layers, as is common in transformer architectures. We dub those modules **m**ultihead-**a**ttention **c**onvolutional **e**ncoders (MACE) and stack two of them for all transformer fusion modules as shown in Fig. 4 except for the second module on the second layer, which uses four MACE modules.

Flow residual. Initial tests suggested that a transformer module, as used for the feature updates, is a poor choice for updating the current flow estimate. Instead, we use a convolutional module for this task. After cross-backward warping the updated features to the reference frame, we pass each pair $(\mathcal{W}_t^{l,s}, \mathcal{W}_{v\to t}^{l,s})$ through a series of convolutions. The output contains the following tensors (stacked in channel dimension): Weight α_v, flow offset Δ_v^F, and context residual $\Delta_v^{\mathcal{W}}$ (We drop the level, time, and step indices of those

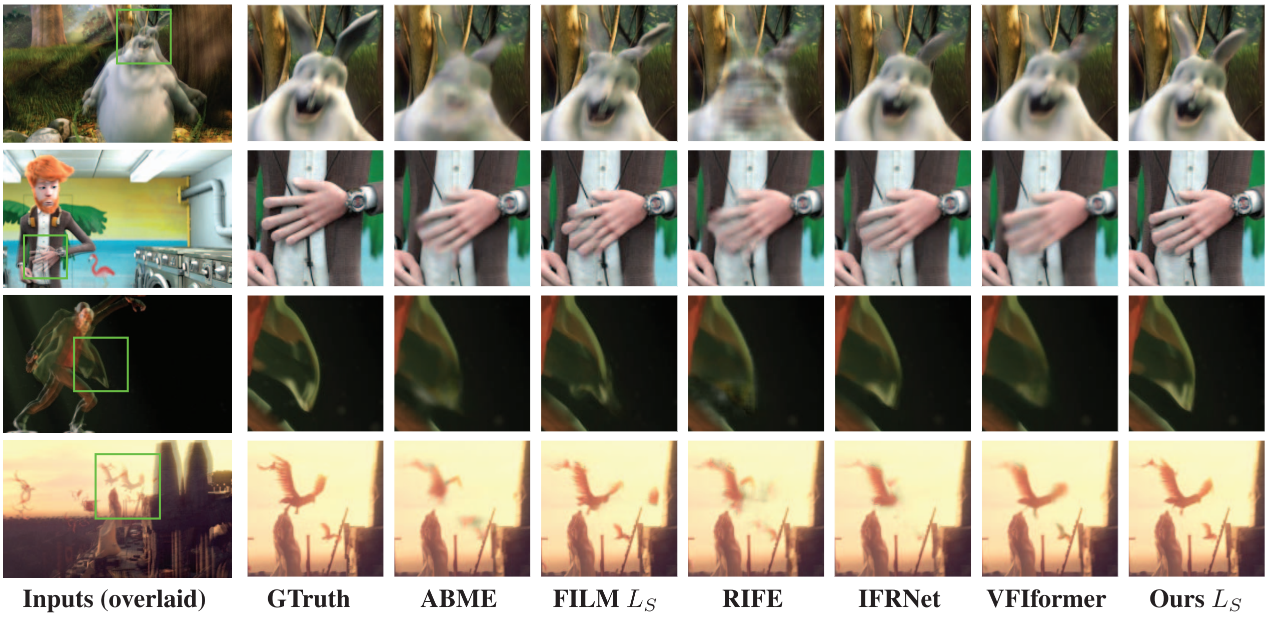


Figure 5. Visual comparison with other methods on rendered movie samples from [6, 7, 14, 15] using only keyframe inputs and no extra rendered patch.

for ease of notation). We apply softmax on the weights and update the flows and context features as

$$F_t^{l,3} = F_t^{l,2} + \frac{\sum_v e^{\alpha_v} \frac{1}{v-t} \Delta_v^F}{\sum_v e^{\alpha_v}} \tag{6}$$

$$\left[\mathcal{W}_t^{l,3}\right]_{C_l..D_l-1} = \left[\mathcal{W}_t^{l,2}\right]_{C_l..D_l-1} + \frac{\sum_v e^{\alpha_v} \Delta_v^{\mathcal{W}}}{\sum_v e^{\alpha_v}}. \tag{7}$$

Note how Δ_v^F needs to be rescaled to a forward flow for the update of $F_t^{l,3}$.

Miscellaneous. For the upsampling of the flows we use parameter-free bilinear interpolation by a scaling factor of two (Denoted by $\cdot_{\uparrow 2x}$) as

$$F_t^{l,0} = 2 {F_t^{l+1,4}}_{\uparrow 2x}. \tag{8}$$

The feature maps are passed through a resize convolution same as [53] to avoid checkerboard artifacts, *i.e.* a nearest-neighbor upsampling followed by a convolutional layer with kernel size 2 and D_l output feature channels.

For the final output, we pass the latent representations $\mathcal{W}_t^0$ together with the extracted features $\mathcal{F}_t^0$ through two convolutional layers with kernel sizes 3 and 1 respectively. The final output has five channels of which the first three form the color image $\hat{I}_t$ and the others correspond to the color error $\hat{E}_t^c$ and the perceptual error $\hat{E}_t^p$.

3.2. Uncertainty estimation

To train the error outputs $\hat{E}$ of the network we compute the target error maps as follows. Let I_t^{GT} be the ground truth frame at time t. We compute the error targets or 'ground truth' as

$$E_t^c = \|I_t^{GT} - \hat{I}_t\|_2 \tag{9}$$

where $\|\cdot\|_2$ denotes the $L2$ norm along the channel dimension. The perceptual error E_t^p follows the computation of LPIPS [65] without the spatial averaging. In order to prevent a detrimental influence of the error loss computations, we do not propagate gradients from the error map computations to the color output and only allow gradient flow to the error prediction of the network.

We want to use the error estimates $\hat{E}$ to find regions of the target frame that are expected to have insufficient quality, so we can render those areas and pass them to the network in a second pass to improve the quality. Assuming that most common renderers should be able to operate on a subset of rectangular tiles without a significant overhead, we average the error estimates for those tiles for which we chose a size of 16×16 pixels. Given a fixed budget for each frame, we simply select the tiles with the highest expected error and use them in the second interpolation pass.

3.3. Implementation and training

We follow common practice and train our network on triplets from the training set of Vimeo-90K [64]. Of the 51313 triplets of resolution 448×256 we set aside 802 for validation. For data augmentation we randomly crop windows of size 256, apply random spatial and temporal flipping and rotations in multiples of 90°. We use empty middle frames for 50% of the training samples (*i.e.* $I_1 = 0$) and otherwise retain between $\frac{1}{480}$ and $\frac{1}{4}$ of 16×16 tiles as additional input (random at first and based on the predicted error for fine-tuning).

We train our L_1 variant for 2.1M iterations with batch size 4 using the Adam optimizer and $L1$ loss for the color output with weight 1.0 and for both error estimates with weight 0.01 each. We start with a learning rate of 5×10^{-5} and reduce it every 0.75M iterations by a factor of 0.464.

For our perceptual variant (L_S), we follow the same schedule, but add VGG and Style loss from [53] after 1.9M iterations, at which point we set the weights of the color, VGG and style loss as 10.0, 0.25 and 40. All losses are computed only for the center frame outputs, as we assume the keyframes are given and complete.

4. Experiments

We evaluate the performance of our method on the standard interpolation task (Sec. 4.1) and the efficiency of the uncertainty guidance (Sec. 4.2). We close with an ablation study (Sec. 4.3) and a discussion of limitations (Sec. 4.4).

Metrics. We measure our results using the common evaluation metrics peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and the perceptual LPIPS [65]. In addition, we perform a user study for a qualitative evaluation.

Methods. We compare our method against ABME [48], AdaCoF [30], CAIN [13], FILM (L_1 and L_S) [53], IFRNet (Large) [28], RIFE [24], VFIformer [37], and XVFI [55].

Datasets. For the evaluation on traditional frame interpolation we use Vimeo90K [64], DAVIS [51], and SNU-FILM [13]. In addition, we evaluate on samples taken from the publicly available animated short films Big Buck Bunny [14], Cosmos Laundromat [7], Elephants Dream [6], and Sintel [15]. See supplementary material for more details and instructions to reproduce those datasets.

4.1. Traditional frame interpolation

We quantitatively evaluate our method on common datasets in Tab. 1 against the state of the art. Our L_1 variant shows the best PSNR and SSIM performance on all difficulty levels of SNU-FILM with a PSNR improvement of up to 0.21 dB in the hard category and a competitive performance on Vimeo90k and DAVIS. Our L_S version outperforms all others in terms of LPIPS on all datasets except DAVIS and demonstrates excellent PSNR and SSIM scores within its category. We show the performance on the animated short films in Tab. 2 where each variant outperforms all others within its category with respect to all metrics and on all datasets except Cosmos Laundromat, where both nevertheless yield good results.

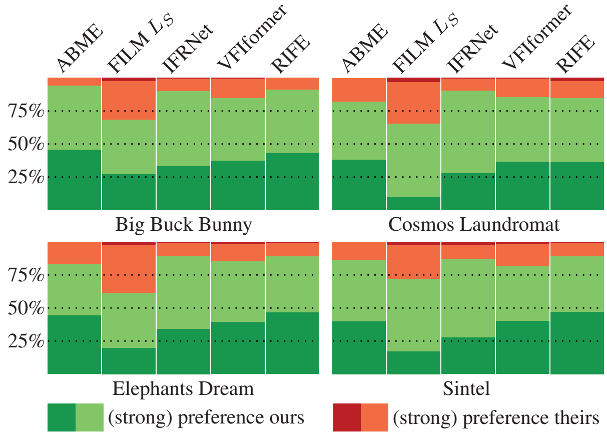


Figure 6. User study on the animated short film datasets. On average, users had a normal/strong preference for our method for 48/34% of all votes. For each of the short films, we use a representative subset of 30 samples and collected a total of 3158 AB comparisons from 69 participants, most of whom are computer graphics/vision students and graduates.

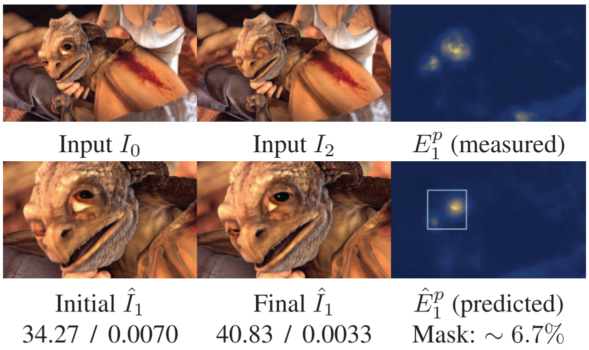


Figure 7. The closing of the eyes proves difficult to interpolate, but the expected perceptual error $\hat{E}_1^p$ closely matches the true error E_1^p. Passing the part of the middle frame indicated by the white box to the network we get a significantly improved interpolation. Numbers below are PSNR/LPIPS. Sample is from [15].

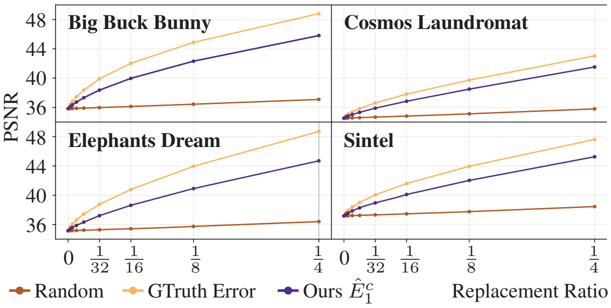


Figure 8. Replacement of tiles based on random sampling, highest ground truth error, *i.e.* the upper boundary of achievable PSNR, and our color error estimation $\hat{E}_1^c$.

Method		Vimeo90k			DAVIS			SNU-FILM												Rank	
								Easy			Medium			Hard			Extreme			Count	
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	1^{st}	2^{nd}
ABME	’21	36.22	0.9808	0.0217	26.47	0.8601	0.1481	39.74	0.9904	0.0228	35.85	0.9792	0.0380	30.62	0.9367	0.0668	25.44	0.8642	0.1271	0	1
AdaCoF	’20	34.38	0.9717	0.0309	25.10	0.8221	0.1550	38.85	0.9902	0.0202	35.07	0.9757	0.0372	29.47	0.9246	0.0764	24.31	0.8442	0.1493	0	0
FILM L_1	’22	36.06	0.9804	0.0201	27.31	0.8784	**0.0846**	40.20	0.9909	0.0186	36.01	0.9795	**0.0321**	30.49	0.9359	0.0578	25.20	0.8601	**0.1071**	3	4
IFRNet	’22	36.20	0.9808	**0.0193**	27.46	0.8797	0.0926	40.10	0.9906	0.0210	36.12	0.9797	0.0328	30.63	0.9368	**0.0570**	25.26	0.8609	0.1138	2	1
RIFE	’22	35.61	0.9780	0.0227	26.70	0.8616	0.1126	40.06	0.9907	0.0188	35.72	0.9789	0.0325	30.09	0.9331	0.0665	24.84	0.8537	0.1395	0	0
VFIformer	’22	**36.50**	**0.9816**	0.0202	**27.60**	**0.8829**	0.0939	40.13	0.9907	0.0181	36.09	0.9799	0.0333	30.67	0.9378	0.0612	25.43	0.8643	0.1190	4	5
XVFI	’21	35.06	0.9758	0.0234	25.71	0.8409	0.1365	39.99	0.9905	**0.0177**	35.36	0.9779	0.0322	29.56	0.9271	0.0752	24.14	0.8446	0.1551	1	1
Ours L_1		36.34	0.9814	0.0204	27.46	0.8803	0.0923	**40.25**	**0.9909**	0.0202	**36.29**	**0.9803**	0.0344	**30.88**	**0.9386**	0.0604	**25.61**	**0.8655**	0.1130	8	6
CAIN	’20	34.67	0.9733	0.0311	26.03	0.8415	0.1787	39.96	0.9903	0.0204	35.64	0.9779	0.0385	29.91	0.9295	0.0898	24.78	0.8510	0.1803	0	0
FILM L_S	’22	35.87	0.9790	0.0132	27.00	0.8709	**0.0679**	**40.15**	**0.9906**	0.0121	35.90	0.9786	0.0215	30.33	0.9333	0.0434	25.07	0.8552	0.0899	3	15
Ours L_S		**36.08**	**0.9799**	**0.0126**	**27.03**	**0.8712**	0.0706	40.10	0.9905	**0.0118**	**36.07**	**0.9790**	**0.0209**	**30.61**	**0.9351**	**0.0420**	**25.35**	**0.8594**	**0.0864**	15	3

Table 1. Live action VFI results. We list perceptually trained methods separately below the other methods. All metrics were obtained by running the implementations provided by the authors.

Method		Big Buck Bunny			Cosmos Laundromat			Elephants Dream			Sintel			Rank #	
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	1^{st}	2^{nd}
ABME	’21	35.60	0.9790	0.0323	34.47	0.9400	0.0823	34.80	0.9647	0.0453	36.83	0.9673	0.0495	0	0
AdaCoF	’20	34.17	0.9740	0.0413	33.83	0.9328	0.0877	33.52	0.9551	0.0560	34.73	0.9550	0.0703	0	0
FILM L_1	’22	35.50	0.9795	0.0282	34.42	0.9397	0.0678	34.70	0.9652	0.0390	36.71	0.9672	0.0395	0	4
IFRNet	’22	35.46	0.9810	0.0292	34.25	0.9399	**0.0674**	34.58	0.9659	0.0419	36.27	0.9683	0.0462	1	0
RIFE	’22	35.05	0.9767	0.0354	34.32	0.9379	0.0808	34.54	0.9615	0.0484	36.33	0.9638	0.0521	0	0
VFIformer	’22	35.97	0.9811	0.0365	**34.56**	**0.9415**	0.0750	35.06	0.9675	0.0406	36.94	0.9694	0.0432	2	6
XVFI	’21	34.64	0.9757	0.0371	34.09	0.9356	0.0774	34.00	0.9595	0.0503	35.51	0.9605	0.0585	0	0
Ours L_1		**35.98**	**0.9815**	**0.0262**	34.55	0.9407	0.0762	**35.25**	**0.9680**	**0.0372**	**37.25**	**0.9697**	**0.0393**	9	2
CAIN	’20	33.38	0.9733	0.0414	33.92	**0.9369**	0.0982	33.57	0.9571	0.0577	35.18	0.9586	0.0727	1	0
FILM L_S	’22	35.31	0.9787	0.0239	**34.20**	0.9361	0.0389	34.67	0.9643	0.0314	36.65	0.9661	0.0316	1	11
Ours L_S		**35.73**	**0.9805**	**0.0218**	34.08	0.9348	**0.0347**	**35.05**	**0.9666**	**0.0295**	**37.01**	**0.9678**	**0.0302**	10	1

Table 2. Animated short film VFI results. We list perceptually trained methods separately below the other methods. All metrics were obtained by running the implementations provided by the authors. Only keyframes were used and no extra rendered patches.

To further support our claim that our method performs well in terms of visual quality, we conduct an extensive user study. We roughly follow the approach of [42] and asked users to compare methods side by side, but included an option for a strong preference. We show one sample of each film in Fig. 5 and give the results in Fig. 6. We refer to the supplementary material for more details and results.

4.2. Uncertainty guided interpolation

We will demonstrate the advantages of our uncertainty guidance in two experiments by analyzing the ability of our error prediction to select appropriate patches in the interpolated image first, and secondly showing the quality improvement by passing additional patches to the network.

In Fig. 8 we demonstrate the PSNR improvement when we use our error estimation to replace a fraction of 16×16 tiles of the interpolated output by the corresponding ground truth. For comparison, we show the effect of random replacement as a baseline and a replacement of the tiles with the highest measured error as the optimal strategy. Replacing a quarter of the tiles, we achieve a PSNR improvement between 6.99 and 9.98 dB, whereas random replacement yields at most 1.27 dB.

Next we want to study the effect of additional inputs on the network output in separation from the error prediction. Therefore, we select tiles based on the true error and pass them into the network. We also compute the metrics when simply replacing the tiles in the interpolated output for our own method as a baseline and a selection of others for comparison. We plot the results in Fig. 9 which show that the perceptual quality is improved beyond the baseline approach.

We give a visual example of the full uncertainty guidance approach in Fig. 7, which shows how the correct region with high error is identified and the interpolation is improved by the additional inputs and refer to the supplementary material for additional results.

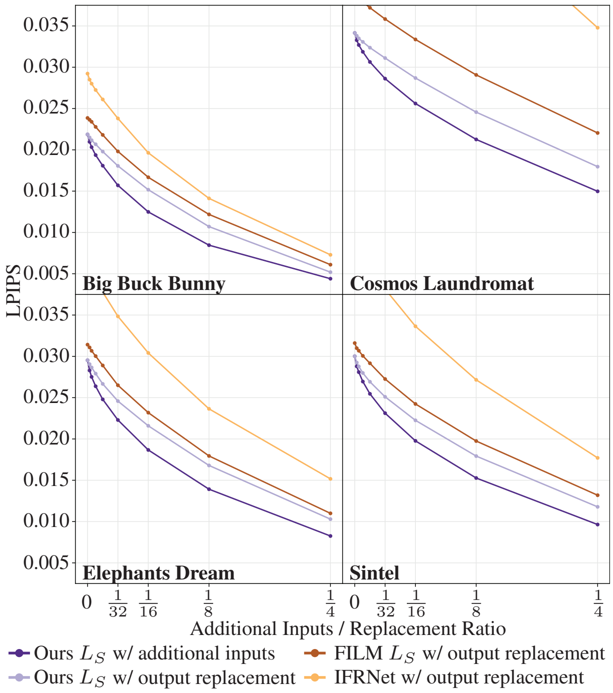


Figure 9. We show that the perceptual quality of the interpolation achieved by passing additional inputs to our method is better than the baseline approach of replacing the worst patches of the interpolation based on color error. For reference, we also show the curves when replacing the outputs of FILM L_S and IFRNet, the two follow up methods in terms of perceptual performance.

Error Est.	Deep Features	Shared Frames	Vimeo90k		Animated	
			PSNR	SSIM	PSNR	SSIM
✔	✔	✔	36.34	0.9814	35.75	0.9650
✔	✘	✔	36.28	0.9812	35.06	0.9633
✘	✔	✔	36.31	0.9813	35.71	0.9652
✘	✔	✘	35.82	0.9796	35.28	0.9634
✘	✘	✘	35.76	0.9793	35.14	0.9629

Table 3. Ablation study of our network design. We averaged the results of all animated films into a single score for each metric. We can see that the shared frame processing boosts the performance significantly, and the deep feature extraction adds a moderate improvement from the baseline, but is essential when interpolating animated content with the error estimation. The latter yields only a minor improvement, but its advantages demonstrated in Sec. 4.2 are significant.

4.3. Ablation study

For an ablation study, we train different versions of our network to show the effect of the error estimation, the deep feature extraction and the shared frame processing. We use the same training procedure and color based loss for all variants as described in Sec. 3.3. The variants without error estimation differ only in the last convolutional layer (3 instead of 5 outputs) and do not use the error losses. The deep feature representation is replaced by the feature representation proposed by Reda et al. [53] and versions without shared frame processing only update the center frame in the transformer fusion and flow/context residual modules. The results are presented in Tab. 3 and highlight the advantages of the deep feature extraction and the shared frame processing for the interpolation quality.

4.4. Limitations

Very large motion or drastic visual changes can be missed by the error prediction and are hence not recovered through a second rendering pass. We show an example of this in the supplementary material. While the shared frame processing of the network through its transformer architecture should in theory be capable of recognizing missing objects that are unlikely to be occluded, we surmise that the current training dataset lacks sufficient examples to learn such behavior.

Lastly, the current network is relatively slow and big. *E.g.* VFIformer is on average 44.2% faster on Vimeo90k and needs about 27.6% fewer parameters. This makes training with more than two input frames challenging, even though the architecture supports it without any changes. We hope to improve this in the future, which could allow for better results through *e.g.* nonlinear flow estimates, or enable using our proposed architecture for other video processing tasks such as deblurring and super-resolution.

5. Conclusion

In this work, we proposed a VFI method that incorporates optical flow motion compensation, deep feature extraction, error estimation, and shared frame processing in a transformer-based architecture. This enables our novel uncertainty-guided approach for animated content production, which can be used to greatly reduce the cost of rendering while maintaining a high visual quality as we have shown in our experiments. At the same time, our method achieves state-of-the-art results for traditional frame interpolation as demonstrated on multiple common benchmarks, and a superior visual quality confirmed by an extensive user study. Since our training procedure using masked inputs is similar to those of masked language models, a study of its properties remains an interesting direction for future work.

Acknowledgements. Markus Plack and Matthias B. Hullin are supported by the European Research Council under ERC Starting Grant “ECHO” (802192).

References

[1] Dawit Mureja Argaw and In So Kweon. Long-term video frame interpolation via feature propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3543–3552, June 2022. 2

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011. 2

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[4] Mojtaba Bemana, Joachim Keinert, Karol Myszkowski, Michel Bätz, Matthias Ziegler, H-P Seidel, and Tobias Ritschel. Learning to predict image-based rendering artifacts with respect to a hidden reference image. In *Computer Graphics Forum*, volume 38, pages 579–589. Wiley Online Library, 2019. 2

[5] Karlis Martins Briedis, Abdelaziz Djelouah, Mark Meyer, Ian McGonigal, Markus Gross, and Christopher Schroers. Neural frame interpolation for rendered content. *ACM Transactions on Graphics (TOG)*, 40(6):1–13, 2021. 2

[6] (c) copyright 2006, Blender Foundation / Netherlands Media Art Institute / www.elephantsdream.org. Elephants dream, 2006. *Licensed under Creative Commons Attribution 2.5* (https://creativecommons.org/licenses/by/2.5/). 5, 6

[7] (CC) Blender Foundation | gooseberry.blender.org. Cosmos laundromat - first cycle - 2k, 2015. *Licensed under Creative Commons Attribution-ShareAlike 4.0* (https://creativecommons.org/licenses/by-sa/4.0/). 5, 6

[8] Zhiqi Chen, Ran Wang, Haojie Liu, and Yao Wang. PDWN: Pyramid deformable warping network for video interpolation. *IEEE Open Journal of Signal Processing*, 2:413–424, 2021. 2

[9] Xianhang Cheng and Zhenzhong Chen. Video frame interpolation via deformable separable convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, Number 07, pages 10607–10614, 2020. 2

[10] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[11] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Yuanhao Yu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. Error-aware spatial ensembles for video frame interpolation. *arXiv preprint arXiv:2207.12305*, 2022. 2

[12] Jinsoo Choi, Jaesik Park, and In So Kweon. High-quality frame interpolation via tridirectional inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 596–604, 2021. 2

[13] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, Number 07, pages 10663–10671, 2020. 2, 6

[14] Copyright (C) 2008 Blender Foundation — peach.blender.org. Big buck bunny, 2008. *Licensed under Creative Commons Attribution 3.0* (http://creativecommons.org/licenses/by/3.0/). 5, 6

[15] Copyright (c) Blender Foundation — durian.blender.org. Sintel, 2010. *Licensed under Creative Commons Attribution 3.0* (http://creativecommons.org/licenses/by/3.0/). 1, 5, 6

[16] Duolikun Danier, Fan Zhang, and David Bull. Enhancing deformable convolution based video frame interpolation with coarse-to-fine 3d cnn. *arXiv preprint arXiv:2202.07731*, 2022. 2

[17] Duolikun Danier, Fan Zhang, and David Bull. St-mfnet: A spatio-temporal multi-flow network for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3521–3531, June 2022. 2

[18] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 2

[19] Saikat Dutta, Arulkumar Subramaniam, and Anurag Mittal. Non-linear motion estimation for video frame interpolation using space-time convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1726–1731, 2022. 2

[20] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14004–14013, 2020. 2

[21] Jie Guo, Xihao Fu, Liqiang Lin, Hengjun Ma, Yanwen Guo, Shiqiu Liu, and Ling-Qi Yan. Extranet: Real-time extrapolated rendering for low-latency temporal supersampling. *ACM Trans. Graph.*, 40(6), dec 2021. 2

[22] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[23] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[24] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. RIFE: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2021. 2, 6

[25] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 2

[26] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field

cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, 35(6), 2016. 2

[27] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arxiv*, 2021. 2

[28] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. IFRNet: Intermediate feature refine network for efficient frame interpolation. *arXiv preprint arXiv:2205.14620*, 2022. 2, 6

[29] Alexandr Kuznetsov, Nima Khademi Kalantari, and Ravi Ramamoorthi. Deep adaptive sampling for low sample count rendering. In *Computer Graphics Forum*, volume 37, pages 35–44. Wiley Online Library, 2018. 2

[30] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. AdaCoF: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020. 2, 6

[31] Sungho Lee, Narae Choi, and Woong Il Choi. Enhanced correlation matching based video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2839–2847, 2022. 2

[32] Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian. Ttvfi: Learning trajectory-aware transformer for video frame interpolation. *arXiv preprint arXiv:2207.09048*, 2022. 3

[33] Meiqin Liu, Chenming Xu, Chao Yao, Chunyu Lin, and Yao Zhao. Jnmr: Joint non-linear motion regression for video frame interpolation. *arXiv preprint arXiv:2206.04231*, 2022. 2

[34] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *European Conference on Computer Vision*, pages 41–56. Springer, 2020. 2

[35] Zhouyong Liu, Shun Luo, Wubin Li, Jingben Lu, Yufan Wu, Shilei Sun, Chunguo Li, and Luxi Yang. Convtransformer: A convolutional transformer network for video frame synthesis. *arXiv preprint arXiv:2011.10185*, 2020. 2

[36] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016. 2

[37] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022. 2, 6

[38] Simone Meyer, Victor Cornillère, Abdelaziz Djelouah, Christopher Schroers, and Markus Gross. Deep video color propagation. In *Proceedings of the British Machine Vision Conference BMVC*, 2018. 2

[39] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[40] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1418, 2015. 2

[41] Simon Niklaus, Ping Hu, and Jiawen Chen. Splatting-based synthesis for video frame interpolation. *arXiv preprint arXiv:2201.10075*, 2022. 2

[42] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 2, 7

[43] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 2

[44] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 2

[45] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 2

[46] Simon Niklaus, Long Mai, and Oliver Wang. Revisiting adaptive convolutions for video frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1099–1109, 2021. 2

[47] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *European Conference on Computer Vision*, pages 109–125. Springer, 2020. 2

[48] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14539–14548, 2021. 2, 6

[49] Anjul Patney and Aaron Lefohn. Detecting aliasing artifacts in image sequences using deep neural networks. In *Proceedings of the Conference on High-Performance Graphics*, pages 1–4, 2018. 2

[50] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2019. 2

[51] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 6

[52] Lars Lau Rakêt, Lars Roholm, Andrés Bruhn, and Joachim Weickert. Motion compensated frame interpolation with a symmetric optical flow constraint. In *International Symposium on Visual Computing*, pages 447–457. Springer, 2012. 2

[53] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame interpolation for large motion. In *European Conference on Computer Vision*, 2022. 2, 3, 5, 6, 8

[54] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. *arXiv preprint arXiv:2111.13817*, 2021. 2

[55] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14489–14498, 2021. 2, 6

[56] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6587–6595, June 2021. 2

[57] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 2

[58] Marc Comino Trinidad, Ricardo Martin Brualla, Florian Kainz, and Janne Kontkanen. Multi-view image fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4101–4110, 2019. 2

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2, 4

[60] Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018. 2

[61] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013. 2

[62] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. Optical flow guided TV-L 1 video interpolation and restoration. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 273–286. Springer, 2011. 2

[63] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[64] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 5, 6

[65] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 6

[66] Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Path-space motion estimation and decomposition for robust animation filtering. *Computer Graphics Forum (Proceedings of EGSR)*, 34(4), June 2015. 2

Publication:
“Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction”

Markus Plack, Clara Callenberg, Monika Schneider, and
Matthias B. Hullin

IEEE/CVF Winter Conference on Applications of Computer Vision
(WACV)

2023

DOI: 10.1109/WACV56688.2023.00308

2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

Fast Differentiable Transient Rendering for Non-Line-of-Sight Reconstruction

Markus Plack Clara Callenberg Monika Schneider Matthias B. Hullin
University of Bonn
Bonn, Germany
{mplack,callenbe,hullin}@cs.uni-bonn.de, moschn@uni-bonn.de

Abstract

Research into non-line-of-sight imaging problems has gained momentum in recent years motivated by intriguing prospective applications in e.g. *medicine and autonomous driving. While transient image formation is well understood and there exist various reconstruction approaches for non-line-of-sight scenes that combine efficient forward renderers with optimization schemes, those approaches suffer from runtimes in the order of hours even for moderately sized scenes. Furthermore, the ill-posedness of the inverse problem often leads to instabilities in the optimization.*

Inspired by the latest advances in direct-line-of-sight inverse rendering that have led to stunning results for reconstructing scene geometry and appearance, we present a fast differentiable transient renderer that accelerates the inverse rendering runtime to minutes on consumer hardware, making it possible to apply inverse transient imaging on a wider range of tasks and in more time-critical scenarios. We demonstrate its effectiveness on a series of applications using various datasets and show that it can be used for self-supervised learning.

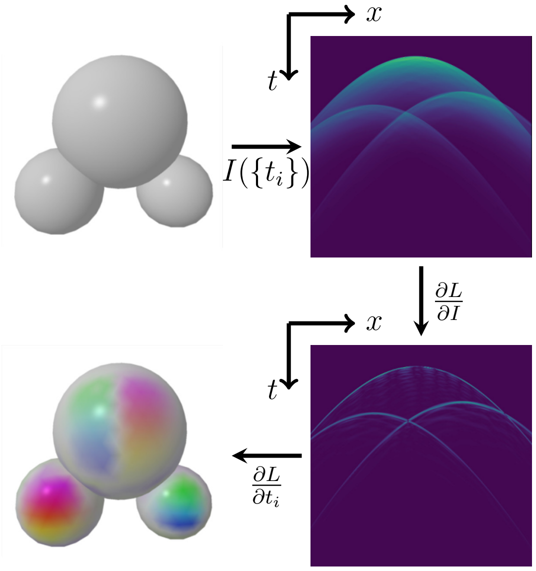

x
t
$I(\{t_i\})$
$\frac{\partial L}{\partial I}$
x
t
$\frac{\partial L}{\partial t_i}$

Figure 1. The triangle mesh $\{t_i\}$ is rendered into a transient image using a physically plausible forward model. After computing the loss, the gradient with respect to the pixel values is backpropagated onto triangle coordinates and their optional attributes. We show a false color visualization, where hue represents the direction and saturation the length of the xy gradients.

1. Introduction

Extending the vision beyond what is in the direct line of sight of an observer is a challenging problem with possible applications ranging from autonomous driving and robotic vision to safety and medical scenarios. Researchers have approached this non-line-of-sight (NLoS) imaging problem by pointing an ultrafast laser source at a wall which is in view of the observer as well as the hidden hidden target scene [35]. Using sensors that are able to resolve the travel time of the laser's light to observe reflections on the same wall, recording *transient images*, objects "around a corner" can be identified and further analyzed.

Many recent methods that use transient images for NLoS reconstruction represent the hidden scene as a volumetric albedo distribution [35, 9, 27]. While they are relatively fast and often yield convincing results, most of those approaches do not take important physical effects such as visibility/occlusion and surface normals into account. On the other hand, it has been proposed to reconstruct the hidden shape as a mesh using an analysis-by-synthesis approach, i.e., by making repeated forward simulations of light transport. Such methods are typically slow and need hours for the reconstruction [33, 11].

This work is inspired by the recent trend to solve inverse problems using task-specific differentiable renderers. The proposed differentiable renderer is specifically targeted to NLoS reconstruction. It extends the forward rendering ap-

2642-9381/23/$31.00 ©2023 IEEE
DOI 10.1109/WACV56688.2023.00308

Table 1. Comparison of relevant NLoS reconstruction approaches in terms of scene representation (**V**olume/**S**urface), usage of a physically-based image formation model (included ✔, somewhat included (✔), not part of the model ✘), their reconstruction time scales ranging from the order of milliseconds (**ms**) to hours (**h**) and their capability to generalize and adapt to new measurement geometries and higher resolutions, ranging from high (+) to intermediate (○) and to low/very low (−/−−) flexibility.

	Backprojection [35, 2]	(Directional) LCT [27, 41]	Occluders and Normals [8]	f-k migration [20]	Transient Rendering [11]	Surf. Optimization [33]	Deep NLoS [7]	Ours
Scene representation	V	V/S	V/S	V	S	S	S	S
Albedo reconstruction	✔	✔	✔	✔	✘	✘	✘	✔
Forward/Inverse Consistency	✘	✘	(✔)	✘	✔	✔	✘	✔
Normals, Occlusion	✘	(✔)	✔	✘	✔	✔	(✔)	✔
Reconstruction time	**s**	**s**	**h**	**s**	**h**	**h**	**ms**	**min**
Generalizability/adaptability	+	−	+	−	+	+	−−	+
Resolution	+	+	−	+	○	−	−	+

proach by Iseringhausen and Hullin [11] with additional degrees of freedom, such as surface albedo, and pairs it with an efficient implementation of the backward pass to backpropagate gradients to the parameters of the scene representation (Fig. 1). This enables the implementation of inverse solvers for a variety of NLoS sensing setups. A key feature of reconstructions obtained this way is that they are inherently consistent with a physically justifiable image formation model, a feature still missing in most recent reconstruction techniques.

We consider the following to be the main contributions of this work:

- We introduce a fast differentiable transient renderer for NLoS light transport. It extends an existing image formation model [11] by spatially varying albedo that is optimized jointly with the scene geometry in a simplified global optimization scheme.
- We demonstrate the effectiveness of the renderer for reconstructing NLoS scenes represented as radial basis functions and depth maps on simulated and real data. We further show that the framework generalizes to very high input resolution and object tracking tasks, thanks to its adaptability to irregular samplings and the use of stochastic optimization algorithms.
- We provide a complete PyTorch implementation of our renderer, along with the implementation of other NLoS reconstruction algorithms and various useful tools.[1]

Our framework runs on a consumer-grade GPU, and has proven to accept a wide range of input configurations. It can therefore serve as a portable and flexible development and test environment for future NLoS reconstruction approaches. We demonstrate this on the example application of a self-supervised network training that is based on our differentiable renderer.

2. Related Work

Transient/NLoS Imaging. Transient imaging allows to capture a scene's light response in space and time. Proposed originally by Abramson as early as 1978 using holographic techniques [1], it has become an increasingly relevant imaging modality with the development and growing accessibility of ultrafast photodetecting devices like streak cameras, single-photon avalanche diodes (SPADs) and photonic mixer devices (PMDs). A comprehensive overview of transient imaging advances can be found in [14].

In NLoS imaging, the light response of a scene is observed not directly, but via its reflection on a relay wall, while the target scene itself is outside the camera's view. Key tasks in this sensing mode are the reconstruction of position, shape and albedo of objects that are hidden both from direct illumination and observation. The reconstruction of NLoS scenes using transient data has been studied intensively using different types of measurement hardware, and different approaches exist in the literature [35, 39, 21, 3, 15, 9, 19, 26, 37, 36]. We compare the most important representatives by their different aspects and features in Table 1. Backprojection-based methods [35, 2] represent the hidden scene as a voxel grid and calculate a heat map of possible locations contributing to the measured space-time data, followed by a filtering step. Furthermore, Shen et al. [30] have proposed to optimize a neural transient field to reconstruct the hidden volume with arbitrary resolution. A different approximative approach, the light-cone transform (LCT) [27],

[1] `https://github.com/unlikelymaths/totrilib`

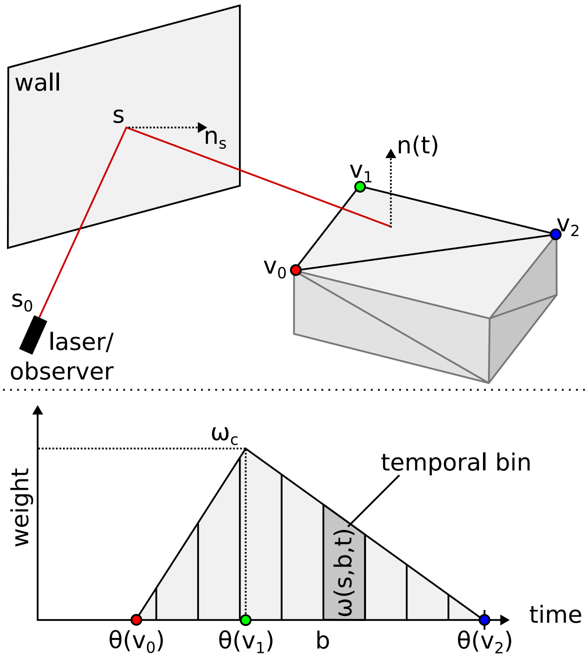


Figure 2. Coaxial measurement setup with the occluded scene represented as triangle mesh (top) and antialiasing of the corresponding temporal response using a trapezoidal filter (bottom).

provides a closed-form solution to the problem in a coaxial setup, where the relay wall is scanned in a regular grid with a beam-combined light source and detector. To reduce the acquisition time of transient images, circular sensing patterns have been proposed [12].

Since scenes represented as scattering density volumes by default do not support surface normals and occlusion effects, extensions with directional kernels [41] and iteratively adjusted linear weights [8] have been proposed. By modelling the light transport as the propagation of a (virtual) wave field, algorithms from wave optics and seismic tomography, like f-k migration, have successfully been adopted to solve the problem for regularly gridded input data [22, 20].

Instead of treating the hidden scene as a voxel-based albedo volume, several recent NLoS algorithms have introduced surface representations, for which physically justifiable light transport models are easier to achieve. After early attempts using planar walls [28], more recent approaches attempt to optimize triangle meshes and their reflectance properties by wrapping stochastic [33] or deterministic [11] renderers a task-specific optimization scheme. The renderer proposed in this paper builds upon the model by Iseringhausen and Hullin [11] and achieves significantly improved reconstruction times by introducing analytical derivatives and utilizing a modern deep learning infrastructure.

Lastly, the availability of large amounts of synthetically generated data has enabled the training of feed-forward networks for the NLoS reconstruction problem for surface-oriented [7], volumetric [4, 25] and implicit [6] scene representations.

Differentiable Rendering. In the case of direct-line-of-sight inverse rendering a number of studies have investigated approaches to compute the gradient of the visibility between two points, which is not differentiable as it is either 0 or 1. This is especially problematic as those gradients are needed to properly move edges across pixels/the visible hemisphere of a surface. One of the first general approaches was published by Li et al. [18]. They compute the gradient through Monte Carlo sampling rays along the edges of triangles. More recently, Zhang et al. [42] have proposed a method to directly differentiate path integrals through a reparametrization. However, in line with the work of Tsai et al. [33], we do not take visibility gradients into account, as the computation would increase the complexity. We still demonstrate that our method works even for cases where occlusion happens in the scene.

In the setting of transient imaging, various approaches have been proposed to address the forward rendering problem [32, 13, 31, 24] and to model sensors for accurate simulation of transient images [10]. General differentiable renderers such as [40, 38] aim to facilitate analysis-by-synthesis reconstruction approaches. However, their universality comes at the cost of computational complexity and they suffer from long runtimes even in cloud computing environments. By restricting the image formation model to the three-bounce NLoS setting, our renderer runs fast on consumer-grade GPUs with moderate amounts of memory.

3. Differentiable Transient Rendering

The key part of our method is the formulation of the transient image formation model as a differentiable function and the efficient backpropagation of gradients through the renderer. We discuss the forward model and the gradient computation in Section 3.1. To increase stability of optimization problems on measurement data, we propose to add a background network in Section 3.2.

3.1. Image Formation

Our image formation model follows that by Iseringhausen and Hullin [11]. Here, we recall it for the coaxial capture geometry, where laser and detector are combined in a single beam, before outlining the computation of gradients. More detailed gradient equations, special cases, and their derivation for both coaxial and independent scanning geometries are given in a supplemental document.

Forward Model. Fig. 2 depicts the measurement setup that is approximated by our renderer and a visualization of the distribution of the recorded light into temporal bins of the time-resolved detector. As interreflections on the object contribute little to the rendered transients, we follow the common three-bounce assumption which only takes light paths into account that move from the laser source s_o to a point on the wall s, onto a triangle $t = (v_0, v_1, v_2)$ of the object surface, back to the wall point s, and are recorded by the time-resolved sensor that is collocated with the laser at s_o.

We approximate the incoming radiance for each triangle by the constant radiance of the triangle centroid $c(t)$ over the full area of the triangle as

$$\alpha(s,t) = f(s \to c(t) \to s)\eta(s \to c(t))\eta(c(t) \to s)A(t), \tag{1}$$

where f denotes the BRDF, $\eta(x \to y)$ the geometric coupling between the two points x and y, and A the area of the triangle. Using $n(t) = (v_1 - v_0) \times (v_2 - v_0)$ as the unnormalized normal vector of the triangle, and n_s as the surface normal of the wall at s, and further assuming Lambertian reflection with albedo $a(t)$, the full expression for α can be simplified to

$$\alpha(s,t) = a(t)\frac{\langle n_s, c(t) - s\rangle^2 \langle n(t), c(t) - s\rangle^2}{\|n(t)\|\|c(t) - s\|}. \tag{2}$$

However, Lambertian reflection is no restriction of our method and any differentiable BRDF model can be used. We have removed the visibility term from α for ease of notation as it is not differentiable, but still perform a visibility check $\nu(s, c(t))$ between the triangle centroid and the wall as seen in Eq. (6).

To compute the total irradiance contributed by a triangle to each transient bin b, $\alpha(s,t)$ is distributed according to a weighting function $w(s,t,b)$ as shown in Fig. 2 according to the length of the light paths and hence the time of flight. Assuming rectified measurements, the corresponding bin of each vertex is given by

$$\theta(v_i) = (2\|v_i - s\|_2 - \phi)/\delta, \tag{3}$$

where ϕ denotes the offset and δ the bin width of the scanning setup. Note that θ is not an integer value and as such is differentiable. We assume that the vertices are sorted in ascending order of total distance. The weight at the center is given as

$$\omega_c(t) = \frac{2}{\theta(v_2) - \theta(v_0)}. \tag{4}$$

For the bins that fall between the points $\theta(v_0)$ and $\theta(v_1)$, we compute the weight as the area under the left triangle as

$$\omega(s,b,t) = \left(b + \frac{1}{2} - \theta(v_0)\right)\frac{\omega_c(t)}{\theta(v_1) - \theta(v_0)}. \tag{5}$$

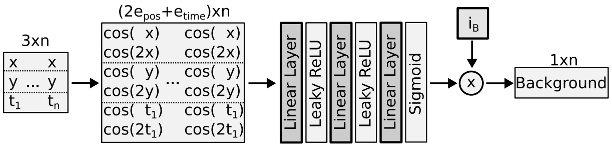


Figure 3. Architecture of our background network. The position of the scan points and the temporal bin are encoded using cosine terms (two each in this example), followed by a linear neural network operating on the first dimension and a scaling. Layers with learnable parameters are highlighted.

The equation for weights between $\theta(v_1)$ and $\theta(v_2)$ follows analogously. The full rendering function of a set of n triangles can be written as

$$I(\{t_0, \ldots, t_{n-1}\}) = \left(\sum_{i=0}^{n-1} \nu(s, c(t_i))\alpha(s, t_i)\omega(s, b, t_i)\right)_{s,b} \tag{6}$$

Backpropagation. To avoid the need for numerical derivatives [11], we explicitly compute gradients through backpropagation of the gradient of a loss function $L(I)$. During the backward pass, we evaluate

$$\nabla_{t_i} L = \sum_s \sum_b \frac{\partial L}{\partial I_{s,b}} \nabla_{t_i} I_{s,b} \tag{7}$$

for each triangle t_i. We can reformulate this as

$$\begin{aligned}\nabla_{t_i} L = \sum_s v(s, t_i)\Bigg(\nabla_{t_i}\alpha(s,t)\sum_b \frac{\partial L}{\partial I_{s,b}}\omega(s,b,t) + \\ \alpha(s,t)\sum_b \frac{\partial L}{\partial I_{s,b}}\nabla_{t_i}\omega(s,b,t)\Bigg).\end{aligned} \tag{8}$$

The gradient of α can be computed using logarithmic derivatives as shown in the supplemental document. In order to efficiently evaluate the gradients, we implement all computations as NVIDIA Optix programs. This enables us to directly continue with the radiance/gradient computation after the visibility test. Note that there is no need to evaluate the full sums in Eq. (8), but only the subset between the bins $\theta(v_0)$ and $\theta(v_2)$ which are evaluated first.

3.2. Background Model and Reconstruction Loss

Even though the formulated model is physically motivated, inconsistencies with real measurements can be expected. This can be due to approximations or in the case where the true BRDF is different from the model. More prominently, there can be background illumination, for instance from other surfaces that are not part of the scene. Those effects would lead to incorrect gradients and reduce the quality of the reconstruction.

To remedy the influence of such effects, we propose to add a background prediction network (Fig. 3) to the optimizations that use the differentiable rendering proposed above. The network takes each scan position (x, y) together with the temporal position t_i and transforms them into positional and temporal encodings using cosines similar to the approach originally proposed by Vaswani et al. [34]. Those encodings are passed through a simple neural network to produce a transient response. To improve performance, the temporal resolution is reduced by a factor of 8 and the transient image produced by the network is linearly upsampled to the final resolution.

We also add a condition to prevent the transient background from capturing too much of the true image as follows. The output of the network $I_B \in (0,1)^{S \times B}$ is scaled using an intensity value i_B that is part of the network parameters. Defining the average power of the transient spectra $P(I)$ we add the condition

$$P(I_B) \leq \lambda_I P(I_R), \quad P(I) = \frac{1}{S}\sum_{i=0}^{S-1} \|I_{i,:}\|_2 \tag{9}$$

which we enforce by clamping i_B appropriately after each optimization step, where I_R is the rendered transient image of the current iterate. The parameter λ_I can be used to control the total amount of light in the transient background. For most of our experiments we set it to 1, which we found to work well.

The benefit of using such a network is that it is independent of the arrangement of scan and laser points and that both sharp jumps as well as smooth gradients can be represented, depending on the input and the effects easily captured by our forward model.

We formulate the reconstruction loss as

$$L(\rho, \phi) = \min_{\gamma} \|\gamma(I_R(\rho) + I_B(\phi)) - I_{\text{in}}\|_2, \tag{10}$$

where ρ is the scene parameterization, ϕ the parameters of the background network, and γ the unknown scaling between the input and the reconstruction. For the optimization of depth maps in Section 4.2 we add γ to the set of parameters after initializing it appropriately. Unfortunately, we found that this approach is problematic in the case of radial basis function optimization as the addition and the removal of blobs can lead to a significant change in the transient image. Instead, we replace γ with the minimizer of Eq. (10).

An extension to other loss functions, that more accurately represent the noise model of transient images, is possible, but similar to [33] we found $L2$ loss to work well over a large range of datasets.

4. Applications

To demonstrate the effectiveness of our implementation, we show its application on three different parametrizations of the geometry used for reconstruction (Section 4.1 and Section 4.2) and tracking (Section 4.3) of hidden objects. In addition, we show that our method can also be used for self-supervised training in Section 4.4.

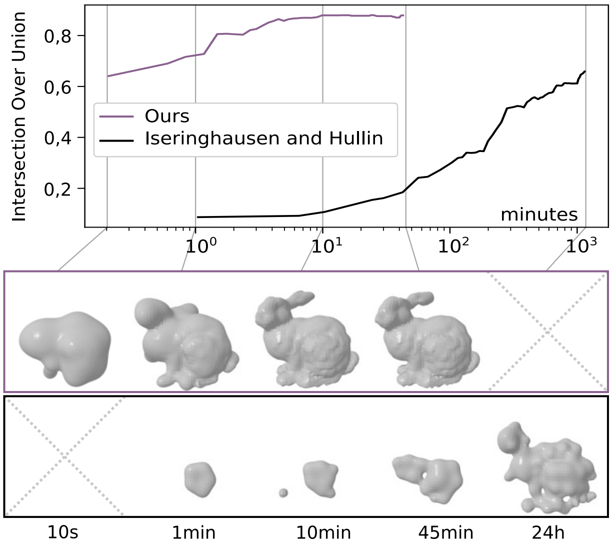


Figure 4. Runtime comparison with the baseline method of Iseringhausen and Hullin [11]. Both methods yield accurate meshes with a mean absolute depth error of 2.91cm (ours) and 2.98cm (baseline) at the end of the optimization for this synthetic 2x2m scene.

We evaluate our method on common datasets using both simulated data from [5] and our own renderer, as well as measurements from [35], [20], and [27].

4.1. Radial Basis Function Approximation

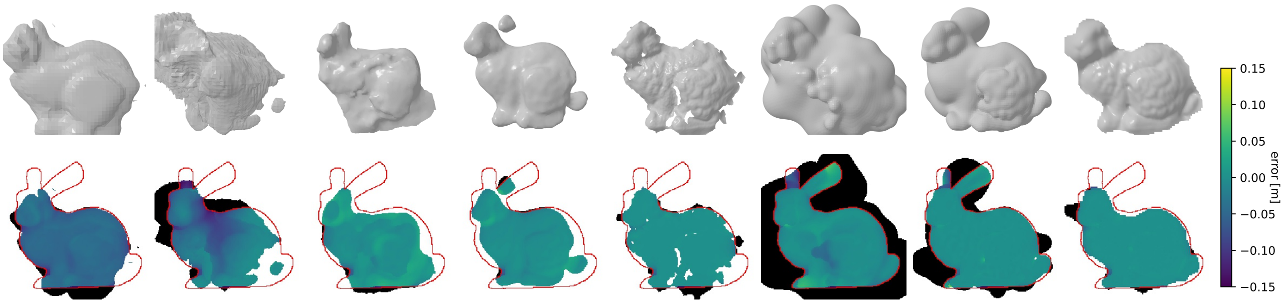
As a direct optimization of triangular meshes is difficult due to e.g. self intersections, we follow the approach of [11] and optimize a set of radial basis functions that approximate the density inside a volume. We generate a mesh by extracting the isosurface using a differentiable marching cubes [23] implementation.

For a set of Gaussian basis functions f_i with parameters p_i and σ_i the density at a position $x \in \mathbb{R}^3$ is given as

$$d(x) = \sum_i f_i(x), \quad f_i(x) = e^{-\frac{\|x-p_i\|_2}{2\sigma_i}}. \tag{11}$$

Additionally, we allow the basis functions to carry attributes such as an albedo value. This yields another volume by computing the weighted average of the attribute values. Those values are interpolated along with the vertex positions in our implementation of the marching cube algorithm.

Note that in this scenario, the computational complexity of the derivative of the rendering as well as the marching cubes step does not depend on the number of radial basis



(a) FBP [35] (b) $f - k$ Migration [20] (c) Fermat [39] (d) D-LCT [41] (e) Tsai et al. [33] (f) Rbf (Sinogram) (g) Rbf (h) Depth Map

Figure 5. Reconstructions of the simulated bunny from [5] of various methods (a–e) compared to our results (f–h). The first row shows the resulting meshes and the second row plots the corresponding depth errors between the respective reconstructions and the ground truth.

functions. Therefore, the iterative algorithm of [11] can be adapted to allow an optimization of all basis parameters in all steps, because there is less need to reduce the number of derivatives that are computed. Additionally, we add another sampling of new blobs that is focused on modifying the surface of the mesh. By backpropagating the current loss to the vertices, we add new blobs at the vertex positions with probability proportional to the length of the vertex gradients. We reduce runtime of the optimization by choosing a rough resolution at the initial iterations and doubling the resolution at certain intervals. More details are given in the supplemental document.

We demonstrate the runtime improvement of our method over the baseline of Iserinhausen and Hullin [11] in Fig. 4. Both methods reconstruct the same synthetically rendered mesh on the same hardware setup. Our method yields convincing results after a few minutes, while the baseline method takes a full day to produce a recognizable solution.

To further evaluate the correctness of our model we use the simulated bunny data from [5] and compare our results qualitatively (Fig. 5) and quantitatively (Table 2) against various other reconstruction methods. To convert volumetric reconstructions into a mesh we use marching cubes [23] and search for a threshold that maximizes the intersection over union (IoU). While our GPU implementations of those methods run much faster, we found that the quality of the results deteriorates quickly when using lower resolution input. At the same time, we needed to use a scanning resolution of 64×64 for a fair comparison with the method of Tsai et al. [33], which also uses differentiable rendering, but is much slower than our method.

While our Rbf-based reconstruction overestimates the shape of the bunny, it manages to reconstruct one ear and the overall shape very accurately, which is confirmed by a IoU value that is only surpassed by our depth map based reconstruction shown in the next section. We also include results for a reconstruction from a transient sinogram as proposed by [12], where the overall shape is even larger, but it still yields convincing results and an error comparable with volume based methods even though only 8.7% of the transient spectra are used.

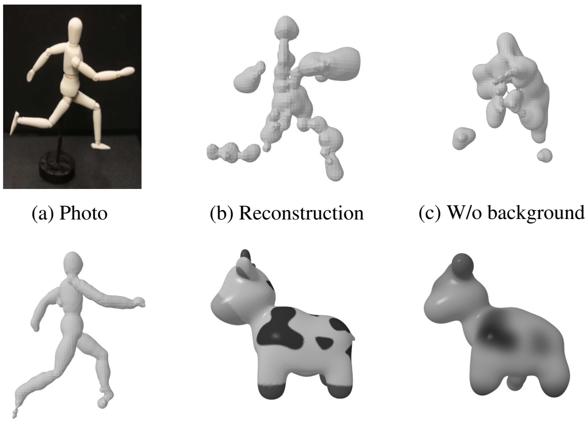
We test the reconstruction of objects with spatially varying albedo on the Spot model and show results in Fig. 6. Although the albedo information is associated with the radial basis functions and not provided as a high-resolution texture, simple changes in albedo are faithfully reconstructed, as can be seen with features like the cow model's dark spots and hooves.

We also demonstrate the application of out method on real data using the mannequin measurements of Velten et al. [35] and show the reconstructions in Fig. 6 along with a reconstruction using a rendered mannequin using the same setup. The overall shape of the reconstruction matches the mannequin from the reference, even though details are lacking when compared to the synthetic reconstruction. As the data was acquired using a non-confocal setup, there are only a few methods that can reconstruct such a measurement. Figure 6 also highlights the ability of our background network to deal with an arbitrary scanning setup and its importance for the reconstruction.

4.2. Depth Map Optimization

In this example application, we optimize the vertex positions similarly to [33]. To remove the need for additional mesh operations we restrict the optimization of the position to the depth values of a grid, i.e. only the z-coordinate is optimized. As such an object would lead to a large amount of unwanted background we also optimize the albedo of the vertices.

To improve stability of this approach we opt to add a total variation regularization [29] to our loss. We regularize both the color attribute as well as the depth. As the color values $c \in [0,1]^{H \times W}$ are naturally bounded to the $[0,1]$ interval, we choose to limit the depth map $d \in [0,1]^{H \times W}$



(a) Photo (b) Reconstruction (c) W/o background

(d) Reconstruction (e) Spot Model (f) Reconstruction

Figure 6. Reconstructions of measured [35] (a–c) and a synthetic mannequin dataset [11] (d), and a reconstruction of the "Spot" model (synthetic), represented using radial basis functions with spatially varying albedo (e,f).

Table 2. Quantitative comparison of reconstructions from the simulated measurements of the bunny [5] with various other methods showing the runtime (minutes:seconds), intersection over union (IoU, higher is better), as well as mean absolute error (MAE, lower is better) and root-mean-square error (RMSE, lower is better) in cm. For each metric, the best value is highlighted in red and the best follow-up in blue.

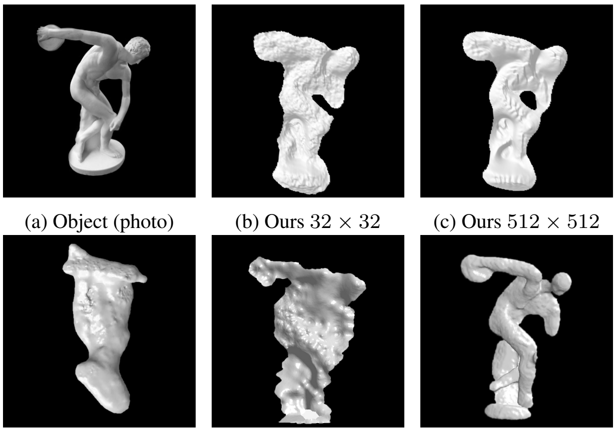
Method	Runtime	IoU	MAE	RMSE
FBP [35]	<0:01	0.738	4.86	5.03
f–k [20]	<0:01	0.659	3.81	4.86
Fermat [39]	0:12	0.730	1.05	1.58
D-LCT [41]	0:05	0.728	0.59	0.95
Tsai et al. [33]	102:06	0.730	0.28	1.03
Rbf	4:51	0.760	0.41	1.33
Rbf (Sinogram)	1:34	0.490	1.13	2.10
Depth Map	2:25	0.803	0.26	0.76

to the same interval and apply a scaling and translation to the reconstruction volume before the rendering. Hence, our loss function can be written as

$$L(c,d) = \|I - R(c,d)\|_2 + \lambda_d TV(d) + \lambda_c TV(c), \quad (12)$$

where TV is an isotropic total variation with $\epsilon = 0.001$ for smoothing with regularization weights λ_d and λ_c. We initialize with a coarse resolution depth map and double the resolution during the optimization.

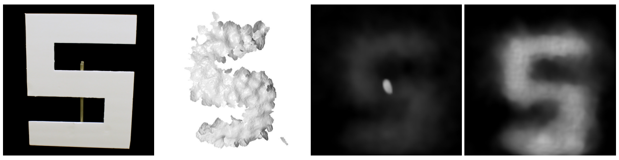
We also evaluate this representation on the synthetic bunny from [5] in Fig. 5. The reconstruction captures the fine details of the surface structure better than all other representations, resulting in the best metrics as listed in Table 2. While D-LCT [41] runs much faster, it lacks some details when compared to differentiable rendering based approaches. At the same time our method offers a significant runtime improvement over the method of Tsai et al. [33].



(a) Object (photo) (b) Ours 32 × 32 (c) Ours 512 × 512

(d) Fermat [39] (e) Tsai et al. [33] (f) D-LCT [41]

Figure 7. Reconstruction of the "Statue" dataset photo shown in (a) [20]. (d)–(f), three reconstructions from recent literature (adapted from [41]). (b) and (c) show reconstructions obtained from our framework using a depth map representation for different input resolutions.



(a) Object (b) Tsai et al. (c) Ours raw (d) Ours flat-field

Figure 8. Reconstructions of the "Diffuse S" dataset [27]. From left to right: photo of the object [27] (a); reconstruction by Tsai et al. [33] (b); reconstructions using our method as depth map with varying albedo: (c), raw dataset; (d), flat-field corrected dataset.

We show the application of this approach on measurement data of a statue [20] in Fig. 7 and the diffuse S [27] in Fig. 8. The quality of the reconstructions of the statue is on par with the reconstruction of D-LCT from [41]. Even after reducing the resolution down to 32 × 32 the quality stays consistent with a reconstruction time of only 39 seconds. For higher resolutions, we switch to a stochastic gradient descent optimization with batch size of 4096 scan points. Therefore, the reconstruction time does not increase beyond a resolution of 64 × 64 and keeps below three minutes.

The reconstruction of the diffuse S shows a failure case of our background network, which cannot deal with the large amounts of spatially varying background present in the dataset. We clean the data up by applying a semi-automatic flat field correction that estimates a static background component from the signal-less portion of the dataset (before the first transient onset). The resulting reconstruction is similar to the one of Tsai et al. [33], but runs in under three minutes.

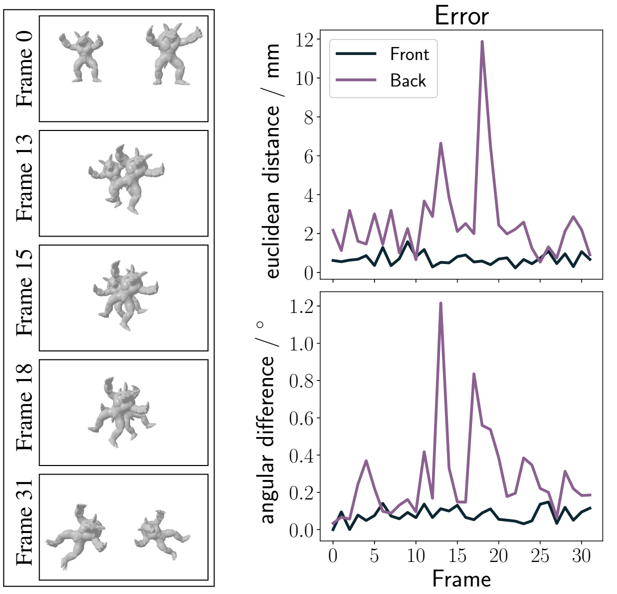


Figure 9. The two armadillos are positioned with 1 m and 1.5 m distance to the wall and perform a linear motion and rotation as indicated by the key frames in the first column. The transient input has a PSNR of 28.4. The plots on the right show the position and rotation error in millimeters and degrees, respectively.

4.3. Tracking

This application takes as input one or more meshes of hidden objects and a transient image of these objects at unknown positions. The aim is to infer the hidden object's spatial position and orientation. To this end, we optimize the position vector and the orientation quaternion of each object to match the given transients.

We demonstrate the tracking of two armadillo meshes over a video in Fig. 9. The first frame is initialized to the correct position and rotation and we iteratively optimize the transformation of both objects for each frame using the results of the previous frame as an initialization.

The positions and rotations are matched with negligible errors for both objects. The accuracy of the armadillo in the back is slightly lower because of the reduced light intensity reaching the wall, and it degrades during the middle of the video where most of the object is occluded by the armadillo in the foreground. The estimation quality is, however, still reasonable even though our method only approximates the full visibility of the triangles and does not compute gradients for the visibility term. The optimization of a single transform with more translation and rotation is shown in the supplemental document.

4.4. Proof of Concept: Self-Supervised Learning

Finally, we demonstrate the flexibility of our differentiable renderer by using it to train a reconstruction network in a purely self-supervised manner. We generate synthetic data from random sets of gaussian blobs similar to Section 4.1. The convolutional network takes the transient image as input and outputs a density volume that is converted into a mesh using marching cubes. We pass this mesh through our differentiable renderer and compute the L2 loss between the resulting transient image and the network input, which can be backpropagated through all steps to update the network parameters.

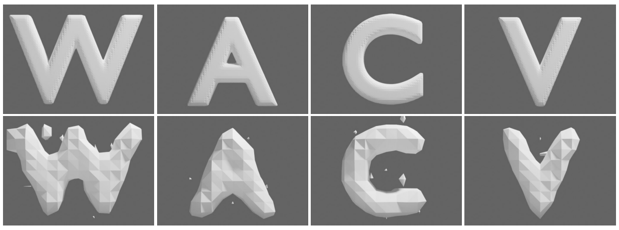


Figure 10. Ground truth models (top) and their reconstructions (bottom) using a network trained in a self-supervised regime with synthetic data generated from volumetric blobs.

We train the network for 500000 iterations using Adam [16] with a batch size of 32. The volume and scan point resolution is set to 16. Additionally, we add a small L2 regularization of the gradients of the volumetric output for smoothness. Results are shown in Fig. 10.

5. Conclusion

We have demonstrated that an efficient computation of the gradients for differentiable transient rendering greatly improves the reconstruction speed compared to other rendering based NLoS reconstructions. Our implementation is general enough to handle many cases and yields reconstructions quantitatively better than other approaches. Paired with a background network we were able to show results on a large range of simulated and real measurements. As the implementation is integrated into the PyTorch environment, it offers great flexibility and we have demonstrated its use in a self-supervised learning application. Furthermore, it may serve as a building block for future end-to-end training approaches or methods that also make use of the latest neural scene representations.

A major limitation of our method is its restriction to three-bounce, pulse-based setups, a necessity to achieve the highest possible performance for non-line-of-sight problems. As future work, we can imagine to extend the software by implementing gradients with respect to scan positions to allow for calibration similar to [17], but using more complex targets.

Acknowledgements. This work was supported by the European Research Council under ERC Starting Grant "ECHO" (802192).

References

[1] Nils Abramson. Light-in-flight recording by holography. *Optics letters*, 3(4):121–123, 1978.

[2] Victor Arellano, Diego Gutierrez, and Adrian Jarabo. Fast back-projection for non-line of sight reconstruction. *Optics express*, 25(10):11574–11583, 2017.

[3] Mauro Buttafava, Jessica Zeman, Alberto Tosi, Kevin Eliceiri, and Andreas Velten. Non-line-of-sight imaging using a time-gated single photon avalanche diode. *Optics express*, 23(16):20997–21011, 2015.

[4] Wenzheng Chen, Fangyin Wei, Kiriakos N Kutulakos, Szymon Rusinkiewicz, and Felix Heide. Learned feature embeddings for non-line-of-sight imaging and recognition. *ACM Transactions on Graphics (TOG)*, 39(6):1–18, 2020.

[5] Miguel Galindo, Julio Marco, Matthew O'Toole, Gordon Wetzstein, Diego Gutierrez, and Adrian Jarabo. A dataset for benchmarking time-resolved non-line-of-sight imaging, 2019.

[6] Javier Grau, Markus Plack, Patrick Haehn, Michael Weinmann, and Matthias Hullin. Occlusion fields: An implicit representation for non-line-of-sight surface reconstruction. *arXiv preprint arXiv:2203.08657*, 2022.

[7] Javier Grau Chopite, Matthias B Hullin, Michael Wand, and Julian Iseringhausen. Deep non-line-of-sight reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 960–969, 2020.

[8] Felix Heide, Matthew O'Toole, Kai Zang, David B Lindell, Steven Diamond, and Gordon Wetzstein. Non-line-of-sight imaging with partial occluders and surface normals. *ACM Transactions on Graphics (ToG)*, 38(3):1–10, 2019.

[9] Felix Heide, Lei Xiao, Wolfgang Heidrich, and Matthias B Hullin. Diffuse mirrors: 3d reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3222–3229, 2014.

[10] Quercus Hernandez, Diego Gutierrez, and Adrian Jarabo. A computational model of a single-photon avalanche diode sensor for transient imaging. *arXiv preprint arXiv:1703.02635*, 2017.

[11] Julian Iseringhausen and Matthias B Hullin. Non-line-of-sight reconstruction using efficient transient rendering. *ACM Transactions on Graphics (TOG)*, 39(1):1–14, 2020.

[12] Mariko Isogawa, Dorian Chan, Ye Yuan, Kris Kitani, and Matthew O'Toole. Efficient non-line-of-sight imaging from transient sinograms. In *European Conference on Computer Vision*, pages 193–208. Springer, 2020.

[13] Adrian Jarabo, Julio Marco, Adolfo Munoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM Transactions on Graphics (ToG)*, 33(6):1–10, 2014.

[14] A. Jarabo, B. Masia, J. Marco, and D. Gutierrez. Recent Advances in Transient Imaging: A Computer Graphics and Vision Perspective. *ArXiv e-prints*, Nov. 2016.

[15] Achuta Kadambi, Hang Zhao, Boxin Shi, and Ramesh Raskar. Occluded imaging with time-of-flight sensors. *ACM Transactions on Graphics (ToG)*, 35(2):1–12, 2016.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Jonathan Klein, Martin Laurenzis, Matthias B Hullin, and Julian Iseringhausen. A calibration scheme for non-line-of-sight imaging setups. *Optics Express*, 28(19):28324–28342, 2020.

[18] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.

[19] David B Lindell, Gordon Wetzstein, and Vladlen Koltun. Acoustic non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6780–6789, 2019.

[20] David B Lindell, Gordon Wetzstein, and Matthew O'Toole. Wave-based non-line-of-sight imaging using fast fk migration. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[21] Xiaochun Liu, Sebastian Bauer, and Andreas Velten. Analysis of feature visibility in non-line-of-sight measurements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10140–10148, 2019.

[22] Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. Non-line-of-sight imaging using phasor-field virtual wave optics. *Nature*, 572(7771):620–623, Aug. 2019.

[23] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[24] Julio Marco, Wojciech Jarosz, Diego Gutierrez, and Adrian Jarabo. Transient photon beams. In *ACM SIGGRAPH 2017 Posters*, pages 1–2. 2017.

[25] Fangzhou Mu, Sicheng Mo, Jiayong Peng, Xiaochun Liu, Ji Hyun Nam, Siddeshwar Raghavan, Andreas Velten, and Yin Li. Physics to the rescue: Deep non-line-of-sight reconstruction for high-speed imaging. *arXiv preprint arXiv:2205.01679*, 2022.

[26] Ji Hyun Nam, Eric Brandt, Sebastian Bauer, Xiaochun Liu, Marco Renna, Alberto Tosi, Eftychios Sifakis, and Andreas Velten. Low-latency time-of-flight non-line-of-sight imaging at 5 frames per second. *Nature communications*, 12(1):1–10, 2021.

[27] Matthew O'Toole, David B Lindell, and Gordon Wetzstein. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature*, 555(7696):338–341, 2018.

[28] Adithya Kumar Pediredla, Mauro Buttafava, Alberto Tosi, Oliver Cossairt, and Ashok Veeraraghavan. Reconstructing rooms using photon echoes: A plane based model and reconstruction algorithm for looking around the corner. In *2017 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2017.

[29] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[30] Siyuan Shen, Zi Wang, Ping Liu, Zhengqing Pan, Ruiqian Li, Tian Gao, Shiying Li, and Jingyi Yu. Non-line-of-sight imaging via neural transient fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2257–2268, 2021.

[31] Malcolm Slaney and Philip A Chou. Time of flight tracer. Technical report, Technical Report. Microsoft Research., 2014.

[32] Adam Smith, James Skorupski, and James Davis. Transient rendering. 2008.

[33] Chia-Yin Tsai, Aswin C Sankaranarayanan, and Ioannis Gkioulekas. Beyond volumetric albedo–a surface optimization framework for non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1545–1555, 2019.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G Bawendi, and Ramesh Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3(1):1–8, 2012.

[36] Bin Wang, Ming-Yang Zheng, Jin-Jian Han, Xin Huang, Xiu-Ping Xie, Feihu Xu, Qiang Zhang, and Jian-Wei Pan. Non-line-of-sight imaging with picosecond temporal resolution. *Physical Review Letters*, 127(5):053602, 2021.

[37] Cheng Wu, Jianjiang Liu, Xin Huang, Zheng-Ping Li, Chao Yu, Jun-Tian Ye, Jun Zhang, Qiang Zhang, Xiankang Dou, Vivek K Goyal, et al. Non–line-of-sight imaging over 1.43 km. *Proceedings of the National Academy of Sciences*, 118(10):e2024468118, 2021.

[38] Lifan Wu, Guangyan Cai, Ravi Ramamoorthi, and Shuang Zhao. Differentiable time-gated rendering. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021.

[39] Shumian Xin, Sotiris Nousias, Kiriakos N Kutulakos, Aswin C Sankaranarayanan, Srinivasa G Narasimhan, and Ioannis Gkioulekas. A theory of fermat paths for non-line-of-sight shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6800–6809, 2019.

[40] Shinyoung Yi, Donggun Kim, Kiseok Choi, Adrian Jarabo, Diego Gutierrez, and Min H Kim. Differentiable transient rendering. *ACM Transactions on Graphics (TOG)*, 40(6):1–11, 2021.

[41] Sean I. Young, David B. Lindell, Bernd Girod, David Taubman, and Gordon Wetzstein. Non-line-of-sight surface reconstruction using the directional light-cone transform. In *Proc. CVPR*, 2020.

[42] Cheng Zhang, Bailey Miller, Kan Yan, Ioannis Gkioulekas, and Shuang Zhao. Path-space differentiable rendering. *ACM transactions on graphics*, 39(4), 2020.