2-13-2020

# Novel View Synthesis in Time and Space

Simon Niklaus
*Portland State University*



Recommended Citation
Niklaus, Simon, "Novel View Synthesis in Time and Space" (2020). *Dissertations and Theses.* Paper 5421.
https://doi.org/10.15760/etd.7294

# Novel View Synthesis in Time and Space

by

Simon Niklaus

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Computer Science

Dissertation Committee:
Feng Liu, Chair
Bart Massey
Melanie Mitchell
Christof Teuscher

Portland State University
2020

## Abstract

Novel view synthesis is a classic problem in computer vision. It refers to the generation of previously unseen views of a scene from a set of sparse input images taken from different viewpoints. One example of novel view synthesis is the interpolation of views in between the two images of a stereo camera. Another classic problem in computer vision is video frame interpolation, which is important for video processing. It refers to the generation of video frames in between existing ones and is commonly used to increase the frame rate of a video or to match the frame rate to the refresh rate of the monitor that the video is being displayed on. Interestingly, off-the-shelf video frame interpolation can directly be employed to successfully perform view interpolation to address the aforementioned stereo view interpolation problem.

Video frame interpolation can be seen as temporal novel view synthesis. However, this perspective is usually not considered and novel view synthesis generally concerns generating unseen views in space rather than time. For this reason, the set of sparse input images that is used for spatial novel view synthesis is commonly either captured at the same time, or it is assumed that the scene is static. This paradigm limits the applicability of novel view synthesis in real-world scenarios though.

This thesis addresses three applications of novel view synthesis and provides practical solutions that do not require difficult-to-acquire multi-view imagery: video frame interpolation which performs temporal video-to-video synthesis, synthesizing

the 3D Ken Burns effect from a single image which performs spatial image-to-video synthesis, synthesizing video action shots which performs spatiotemporal video-to-video and video-to-image synthesis. These applications not only explore different dimensions of time and space, they also perform novel view synthesis on everyday image and video footage. This is in stark contrast to the large body of existing work which focuses on spatial novel view synthesis while requiring multiple input views that were either captured at the same time or under the assumption of a static scene.

## Acknowledgements

I would not be where I am today without the following individuals. Please note that this is by no means a complete list and any omissions have been accidental.

*Feng Liu.* Thank you for giving me the opportunity to be your student. You taught me how to be a scholar and I am and forever will be grateful for it.

*Aleena Watson.* Thank you for your loving support throughout this journey. You allowed me to let my mind wander, and helped me focus on what's important.

*Long Mai.* Thank you for all the work that we have done together. I consider myself to have been successful and you are a driving force behind this success.

*Bart Massey.* Thank you for seeing something in me that others did not. Without you, I would have not considered pursuing a doctoral degree in the first place.

*Hoang Le.* Thank you for always being there for me when I needed you. I am fortunate to have you as a friend and I still owe you several dinners.

*Cuong Nguyen.* Thank you for having an ear for what I had to say. The conversations with you greatly contributed to my well-being in difficult times.

*Karin Niklaus.* Thank you for giving me the space I needed. It must be difficult to have me gone, but you are where my home is and I will always come back.

*Sven Holzheid.* Thank you for being a livelong friend that has always been there for me. Albeit being afar in space, you have always been close to my heart.

*Wu-chi Feng.* Thank you for always being on my side and letting me teach my

own classes. I like to believe that it allowed me to leave a legacy behind.

*Rebecca Sexton-Lee.* Thank you for being a fantastic graduate advisor. You have saved me more than once by addressing my issues in a timely manner.

*Christine Igarta.* Thank you for being an outstanding international advisor. Anything visa-related can be pretty stressful but you were always here to help.

*Oliver Wang.* Thank you for being my manager at Adobe. I am excited to be working with you and am grateful for your unconditional support.

*Tianfan Xue.* Thank you for having me as your intern at Google. Without your support, the project would not have been as successful as it ended up being.

*Jon Barron.* Thank you for being a mentor. You are the reason that I applied for an internship at Google, and I am grateful for all the advise you have given me.

*Melanie Mitchell.* Thank you for following me throughout my studies. It has always been a major milestone for me when I emailed you about a new paper.

*Jonathan Walpole.* Thank you for all the thoughtful conversations that we have had. I have always been looking forward to our weekly lunch meetings.

*Jess Millar.* Thank you for being an exceptional friend throughout. I am fondly looking back at our shared time in and out of the student dormitory.

*Bo Chen.* Thank you for helping me to get used to being a teaching assistant and for all the conversations that we have had during our weekend lunches.

*Bob Yakas.* Thank you for showing me that youth is ageless. We sadly did not find much time for each other, but I am cherishing the moments we shared.

*Alvin P. Adams.* Thank you for sharing your history with me, I learned a lot from you. It is with a heavy heart that I had to let you go, rest in peace old friend.

## Table of Contents

## List of Tables

## List of Figures

# 1 Introduction

Computer vision is an area within computer science that strives to find computational ways to gain a high-level understanding from image and video footage. It is a growing area and one of its prime conferences, the IEEE Conference on Computer Vision and Pattern Recognition, has recently reached an h5-index of 240 which makes it the most prominent conference within computer science. Nowadays, one can simply upload an image to a cloud provider which will then return a high-level description of the image in the form of tags and object annotations [1] . This technology is based on decades of research in image recognition. Many other research efforts within computer vision, like novel view synthesis, are less easy to make use of.

Novel view synthesis focuses on generating previously unseen views of scenes or 3D objects from a sparse set of input images taken at different viewpoints. This set of images is commonly either captured at the same time, or it is assumed that the scene is static and does not change between individual captures. This simplified setting makes novel view synthesis more tractable, but it significantly limits its applications. This limiting factor is further amplified by existing research that predominately focuses on novel view synthesis in space, with little exploration of the time domain.

The emphasis of this thesis is to analyze novel view synthesis in the context of time and space by exploring three different applications. In doing so, each application has a

[1] `https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/`

strong emphasis on practicability and hence performs novel view synthesis on everyday image and video footage without constraining how the material was captured.

> *Thesis Statement.* Novel view synthesis in time and space can be performed in a practical manner without requiring difficult-to-acquire multi-view imagery.

This is in stark contrast to the large body of existing work in novel view synthesis which focuses on the generation of spatially novel views from a sparse set of images that were either captured at the same time or under the assumption of a static scene. The three applications that this thesis explores are as follows.

1. *Video Frame Interpolation.* Given a video, video frame interpolation performs novel view synthesis in time to generate new video frames in-between existing ones in order to change the frame rate of the input video (Figure 1.1).

2. *3D Ken Burns Effect from a Single Image.* Given an image, the 3D Ken Burns effect depicts the image as a video where a virtual camera is moving through the scene in 3D such that the resulting animation correctly depicts motion parallax and conveys the perception of depth (Figure 1.2).

3. *Video Action Shot Synthesis.* Given a video, for example of a person riding a bicycle, an action shot visually summarizes the motion of the person. This can either be done in the form of an image or of a video that depicts multiple past and future snapshots of the person at the same time (Figure 1.3).

When not only operating in space but also in time and when not constraining how the input imagery was captured, novel view synthesis becomes increasingly difficult. The following outlines a key challenge for each of the three applications.

1. Video frame interpolation is subject to non-rigid deformations between individual frames. In contrast, when performing view interpolation between stereo images,

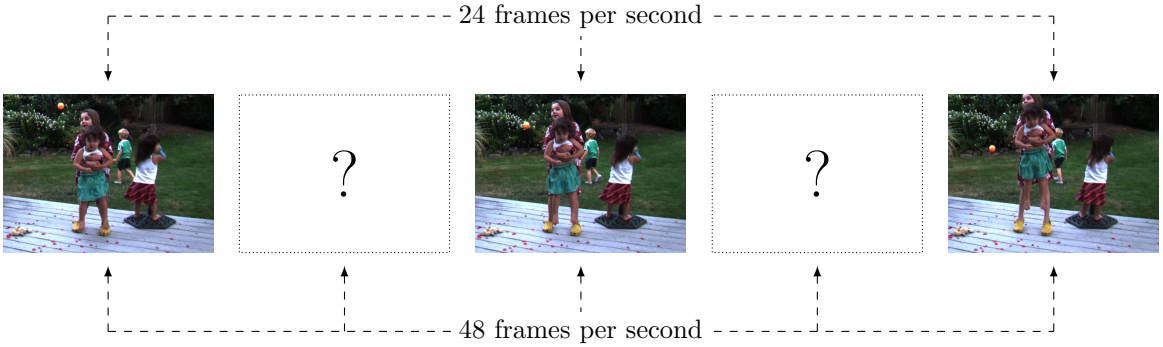



Figure 1.1: An illustration of frame interpolation to double the frame rate of a video.

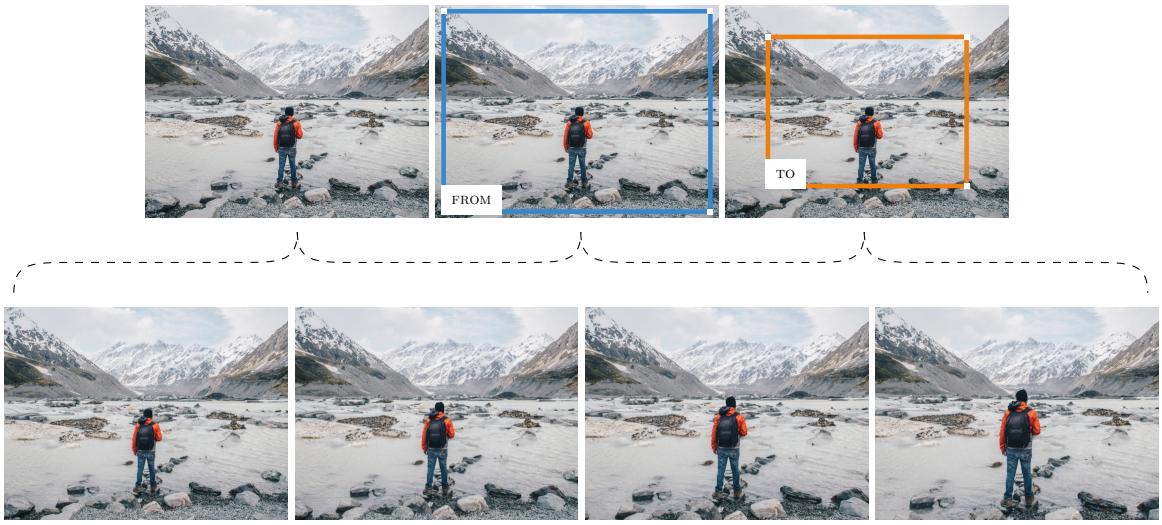



Figure 1.2: An example 3D Ken Burns effect, zooming with accurate motion parallax.

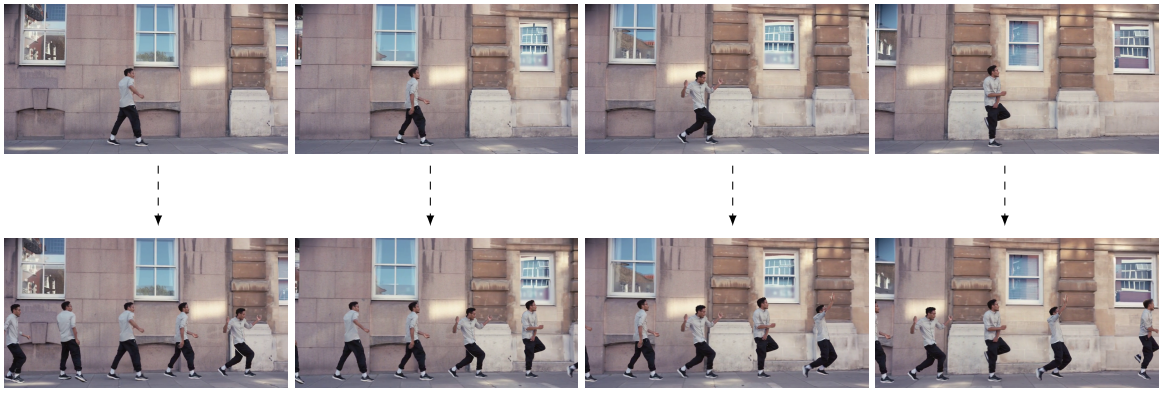


Figure 1.3: An example video action shot, which illustrates the motion of a subject.

one can establish geometrically sound correspondences between the input views. To deal with these non-rigid deformations, this thesis makes use of a learning-based approach that compensates for inter-frame motion without making explicit assumptions about the scene geometry. Further, traditional frame interpolation techniques employ forward warping but there is no definitive way to implement this operation in a differentiable manner as required when employing deep learning. This thesis proposes softmax splatting to account for this.

2. Synthesizing a 3D effect from a single image is inherently challenging and existing methods are often only applicable to specific scene types, 3D object models, or domain-specific light field imagery. To make synthesizing the 3D effect more applicable, this thesis identifies and addresses several key problems with the common way of estimating the scene geometry as image depth by tailoring the depth estimate to the task of novel view synthesis.

3. Transferring the appearance of an object in motion from a source frame to a target frame requires knowing how the camera moves and where the object is within the world geometry. The former is challenging due to the moving object since many approaches for visual odometry and structure from motion assume a static scene. The latter is challenging since the transferred object in the target frame may be depicted from an angle that differs from the source frame while simultaneously being subject to non-rigid deformations. This thesis makes use of human priors to better reason about these aspects, thus making it possible to effectively synthesize video action shots.

This thesis follows a multi-paper format of the electronic thesis and dissertation formatting requirements of Portland State University [2] . It is separated into a related

[2] `https://www.pdx.edu/ogs/etd-formatting`

work section and three main chapters. The related work section summarizes relevant literature for all main chapters and each main chapter corresponds to an academic paper that has either already been published, is in the process of being published, or is being prepared for submission as outlined below.

1. The chapter on frame interpolation has been submitted to the IEEE Conference on Computer Vision and Pattern Recognition.

2. The chapter on synthesizing the 3D Ken Burns effect from a single image has been published in the ACM Transactions on Graphics.

3. The chapter on video action shot synthesis will be submitted to a conference such as the IEEE Conference on Computer Vision and Pattern Recognition.

Please also consider reviewing the supplementary video demos, which accompany the individual papers. While showing pictures may in many cases be sufficient to convey the point at hand, motion pictures are at the heart of this thesis and it is thus important to examine them in their original form.

# 2 Related Work

Novel view synthesis focuses on generating novel views of scenes or 3D objects from input images taken from a sparse set of viewpoints. It is important for a wide range of applications, including virtual and augmented reality [46, 54, 126], 3D display technologies [26, 65, 72, 121, 161], and image- or video-manipulation [67, 69, 75, 83, 119, 180]. Novel view synthesis is typically solved using image based rendering techniques [64], with recent approaches allowing for high-quality view synthesis results [20, 21, 46, 47, 48, 113]. With the emergence of deep neural networks, learning-based techniques have become an increasingly popular tool for novel view synthesis [33, 59, 61, 94, 98, 135, 138, 146, 147, 164, 176]. As such, the research on novel view synthesis has made and is continuing to make great strides towards enabling more applications while striving for high-quality results.

The following section reviews the related work concerning three different applications of novel view synthesis: video frame interpolation (temporal, video-to-video), synthesizing the 3D Ken Burns effect (spatial, image-to-video), and synthesizing video action shots (spatiotemporal, video-to-video or video-to-image).

## 2.1 Video Frame Interpolation

Video frame interpolation concerns the synthesis of frames in between existing frames of a video. Research on frame interpolation has seen a recent resurgence, with

multiple papers proposing various flow-based [9, 10, 60, 85, 86, 106, 120, 125, 165], kernel-based [9, 10, 107, 108], and phase-based [96, 97] approaches. This thesis builds upon the approach from Niklaus *et al.* [106] who estimate optical flow between two input images in both directions, extract generic contextual information from the input images using pre-trained filters, forward-warp the images together with their context maps according to optical flow, and finally employ a synthesis network to obtain the interpolation result. This thesis extends their approach and proposes softmax splatting, which enables warping task-specific feature pyramids for image synthesis in an end-to-end manner. This includes fine-tuning the off-the-shelf optical flow estimator for the task of frame interpolation and supervising the metric that is used to disambiguate cases where multiple pixels map to the same location.

*Differentiable Image Sampling.* With the introduction of spatial transformer networks, Jaderberg *et al.* [57] proposed differentiable image sampling. Since then, this technique has found broad adoption in the form of backward warping to synthesize an image $I_A$ from an image $I_B$ given a correspondence $F_{A \to B}$ for each pixel in $I_A$ to its location in $I_B$. Prominent examples where this approach has been used include unsupervised depth estimation [39, 90, 175], unsupervised optical flow prediction [93, 154, 169], optical flow prediction [55, 122, 141], novel view synthesis [23, 84, 177], video frame interpolation [9, 60, 85, 86], and video enhancement [17, 143, 165].

*Differentiable Forward Warping.* In contrast to differentiable image sampling, performing forward warping to synthesize $I_B$ from $I_A$ based on $F_{A \to B}$ has seen less adoption with deep learning, partly due to additional challenges such as multiple source pixels in $I_A$ possibly being mapped to the same target location in $I_B$. For optical flow estimation, Wang *et al.* [154] forward-warp an image filled with ones to obtain an occlusion mask. However, they sum up contributions of all the pixels that are mapped to the same output pixel without a mechanism to remove possible

outliers, which limits the applicability of this technique for image synthesis. For frame interpolation, Niklaus *et al.* [106] use the equivalent of z-buffering which is well motivated but not differentiable [104]. Bao *et al.* [9] linearly weight the optical flow according to a depth estimate as an approach for dealing with multiple source pixels mapping to the same target location. However, adding a bias to the depth estimation affects the result of this linearly weighted warping and leads to negative side effects. In contrast, the proposed softmax splatting is not subject to these concerns.

*Feature Pyramids for Image Synthesis.* For image synthesis, Niklaus *et al.* [106] warp context information from a pre-trained feature extractor that a synthesis network can use to make better predictions. Bao *et al.* [9] refined this approach through end-to-end supervision of the feature extractor. In contrast, this thesis proposes extracting and warping a feature pyramid which allows the synthesis network to make use of a multi-scale representation for better interpolation results. The proposed use of feature pyramids for image synthesis is inspired by recent work on video analysis. For video semantic segmentation, Gadde *et al.* [36] warp features that were obtained when processing the preceding frame in order to support the segmentation of the current frame. For optical flow estimation, Hui *et al.* [55] and Sun *et al.* [141] extend this idea of warping features and employ it across multiple scales in the form of feature pyramids. These approaches do not target image synthesis though.

*Task-specific Optical Flow.* Learning feature pyramids for image synthesis is not the only type of task-specific supervision within the proposed frame interpolation pipeline. Recently, Xue *et al.* [165] demonstrated the benefit of fine-tuning an optical flow predictor for the task that the estimated optical flow is being used for. This thesis follows this principle and fine-tunes the utilized off-the-shelf optical flow predictor for the task of video frame interpolation. This step requires end-to-end supervision, which is seamlessly supported by the proposed softmax splatting.

*Temporally Consistent Synthesis.* Temporal consistency is a common concern when synthesizing images in time [7, 52, 53, 73]. For frame interpolation, Jiang *et al.* [60] collect a specialized training dataset with frame-nonuples and supervise their network on seven intermediate frames at a time in order to ensure temporally consistent results. In the same vein, Liu *et al.* [85] and Reda *et al.* [125] utilize cycle consistency to better supervise their model. In comparison, the proposed softmax splatting leads to temporally consistent results without requiring a specialized multi-frame training dataset or additional cycle-consistency during training.

## 2.2 3D Ken Burns Effect Synthesis

The traditional Ken Burns effect animates images with a 2D scan and zoom, its 3D counterpart augments this paradigm by introducing motion parallax and thus providing a more compelling result. Creating such an effect from a single image is painstakingly difficult. The photo must be manually separated into different segments, which then have to carefully be arranged in the virtual 3D space, and inpainting needs to be performed to avoid holes when the virtual camera moves away from its origin. This thesis targets the problem of automatically synthesizing the 3D Ken Burns effect from a single input image, which is an extreme form of novel view synthesis.

*Learning-based View Synthesis from a Single Image.* Recent novel view synthesis methods approach the single-image setting using deep learning [144, 177]. Synthesizing novel views from a single image is inherently challenging and existing methods are often only applicable to specific scene types [43, 84, 105], 3D object models [111, 112, 127, 166, 167], or domain-specific light field imagery [139]. Most relevant to this thesis are methods that estimate the scene geometry of the input image via depth [23, 84], normal maps [84], or layered depth [148]. To synthesize the 3D Ken Burns effect, the approach proposed in this thesis likewise first predicts the depth of the input image and

then performs depth-based view synthesis. This two-step approach explicitly captures the scene geometry within the depth map, which makes it possible to directly improve the estimated scene geometry to suppress artifacts such as geometric distortions and to tailor the depth prediction to the task of view synthesis.

*Single-image Depth Estimation.* Single-image depth estimation has gained a lot of research interest over the past decades [68]. Recent advances in deep neural networks along with the introduction of annotated depth image datasets [1, 22, 74, 80, 129, 133, 159, 173] enabled large improvements in monocular depth estimation. Another promising direction is the use of spatial or temporal pixel-correspondence to train for depth estimation in a self-supervised manner [37, 39, 41, 79, 89, 149, 175]. However, depth estimation from a single image remains an open research problem. The quality of the predicted depth maps varies depending on the image type and the depth maps from existing methods are in many scenarios not suitable for generating high-quality novel view synthesis results due to geometric and semantic distortions as well as inaccurate depth boundaries. To support the 3D Ken Burns effect synthesis, this thesis introduces a pipeline consisting of depth prediction, adjustment, and refinement to specifically address those issues.

*Creative Effect Synthesis.* With 3D scene information such as depth or scene layouts, a range of creative camera effects can be produced from the input image, such as depth-of-field synthesis [152, 153], 2D-to-3D conversion [161], and photo pop-up [49, 140]. This thesis focuses on synthesizing the 3D Ken Burns effect which is a camera motion effect. The desired output is a whole video corresponding to a given camera path. A number of methods have been proposed in the past to enable camera fly-through effects from a single image. Horry *et al.* [50] present a semi-automatic system that lets users represent the scene with a simplified spidery mesh after a manual foreground segmentation process. The image is then projected onto that simplified

scene geometry which allows flying a camera through it to obtain certain 3D illusions. Based on a similar idea, follow-up work enriches the scene representation to handle scenes with more than one vanishing point and more diverse camera motions [63, 77]. While realistic effects can be achieved for certain types of images, the simplified scene representation is often too simplistic to handle general types of images and still requires manual segmentation which demands significant user effort. Most related to this thesis is the system from Zheng *et al.* [172] which synthesizes a video with realistic motion parallax from still images. This method, however, requires multiple images as input whereas this thesis focuses on a more challenging problem of synthesizing the desired effect from a single input image.

*Image-to-Video Generation.* The intended output of the 3D Ken Burns effect synthesis is a video that is subject to motion parallax. This thesis is thus also related to image-to-video generation, an increasingly popular research topic in computer vision. Existing work in this area focuses on developing generative models to predict motions in video frames given one or a few starting frames [51, 76, 81, 91, 124, 151, 162]. While promising results have been achieved for synthesizing object motion in videos with static background, they are often not suitable for the task of synthesizing realistic camera motion effects as required to produce the 3D Ken Burns effect.

## 2.3 Video Action Shot Synthesis

Summarizing the motion of an object within a single image is a common objective in photography, for example in the form of stroboscopic photography or artistic motion blur [15, 27]. The task that this thesis is addressing, video action shot synthesis, is likewise concerned with the depiction of motion. However, instead of summarizing the motion of a video in an image, video action shot synthesis focuses on visualizing motion within the video itself by projecting past and future occurrences of the main

subject into each frame. As such, each frame within a video action shot can already be considered as a summarization.

*Summarizing Videos as Images.* Summarizing videos is a classic problem in computer vision, with early work focusing on aspects like how to incorporate depth to deal with occlusion [35], how to address redundancies in the motion [18], or how to represent human motion in particular [6, 171]. With the rise of surveillance cameras, video summarization further gained traction in the form of video synopsis which supports people when reviewing hour-long videos [92, 114, 115, 123]. An inherent aspect of video summarization is that it requires the sampling of moments in time and not all moments are suitable to create a compelling result. Dedicated efforts thus analyze interactive video summarization techniques [103, 145]. Most relevant to this thesis is the work from Klose *et al.* [67] who show examples of video action shots. However, their approach requires a camera that also captures scene depth.

*Camera Motion Estimation.* When analyzing the motion of a subject in a video in which camera motion is present, one needs to account for the path of the camera in order to correctly align the main subject across video frames. Visual odometry as well as simultaneous location and mapping are popular techniques from the robotics community that estimate the camera motion [30, 31, 100, 101]. Similarly, structure-from-motion is a common technique within the computer vision community to estimate the camera extrinsics as well as intrinsics [25, 130, 131, 136]. However, these approaches assume that the scene is static, which is not the case with video action shot synthesis. To address this issue, this thesis focuses on video action shots with humans as the main subject. This makes it possible to easily identify the main subject and to exclude it from the estimation of the camera motion, for example by employing COLMAP [130, 131] and removing feature points that belong to the human.

*Modeling the Subject.* In videos with present camera motion, it is not sufficient

to align the main subject across video frames in pixel space. It is also necessary to determine the 3D location as well as the shape of the main subject. While it is possible to estimate the camera motion and the scene geometry at the same time, current approaches to do so assume that the scene is static and are hence not applicable for video action shot synthesis [149, 168, 90, 175]. On the other hand, estimating the scene geometry from a single image is highly challenging and thus subject to inaccurate predictions [22, 37, 39, 79, 80, 118]. By focusing on humans, this thesis allows for a better representation of the main subject thanks to the large body of work on human modeling [4, 5, 13, 42, 62, 88, 117, 142, 150, 174].

*Synthesizing Human Avatars.* Even with a known camera path as well as the shape and 3D location of the main subject, projecting the subject from one frame to another is not as straightforward as it seems. Since the camera path may significantly alter the perspective of the main subject, one cannot simply copy the pixels of the human subject from one frame to another since it may lead to an incomplete representation. Even when focusing on humans as the main subject, most approaches for human modeling either only recover a textureless shape [13, 62, 142, 150, 174], assume the camera to be static [11, 19, 178], assume the human to be static [3], require the human to be depicted in a canonical pose [4], are trained on single-domain data and hence lack generalizability [128, 132], or rely on clever heuristics [156]. It is also not immediately possible to combine information from the main subject across multiple frames in order to complete its structural representation since it may be subject to non-rigid deformations. This thesis focuses on synthesizing humans from point clouds while minimizing noticeable artifacts through rendering heuristics.

# 3 Novel View Synthesis in Time

This chapter has been adapted from a pending CVPR submission. All uses of "we" or "our" refer to the authors of this paper (Simon Niklaus and Feng Liu). Simon's contributions are: forward warping via softmax splatting, feature pyramids for image synthesis, the architecture details such as using PWC-Net for optical flow prediction and a GridNet for image synthesis, all implementation aspects, the design and execution of the evaluation, and the paper writing.

## 3.1 Abstract

Differentiable image sampling in the form of backward warping has seen broad adoption in tasks like depth estimation and optical flow prediction. In contrast, how to perform forward warping has seen less attention, partly due to additional challenges such as resolving the conflict of mapping multiple pixels to the same target location in a differentiable way. We propose softmax splatting to address this paradigm shift and show its effectiveness on the application of frame interpolation. Specifically, given two input frames, we forward-warp the frames and their feature pyramid representations based on an optical flow estimate using softmax splatting. In doing so, the softmax splatting seamlessly handles cases where multiple source pixels map to the same target location. We then use a synthesis network to predict the interpolation result from the warped representations. Our softmax splatting allows us to not only interpolate frames at an arbitrary time but also to fine tune the feature pyramid and optical

flow. We show that our frame synthesis approach, empowered by softmax splatting, achieves new state-of-the-art results for video frame interpolation.

## 3.2 Introduction

Video frame interpolation is a classic problem in computer vision with many practical applications. It can, for example, be used to convert the frame rate of a video to the refresh rate of the monitor that is used for playback, which is beneficial for human perception [70, 71]. Frame interpolation can also help in video editing tasks, such as temporally consistent color modifications, by propagating the changes that were made in a few keyframes to other frames [95]. Frame interpolation can also support inter-frame compression for videos [157], serve as an auxiliary task for optical flow estimation [87, 158], or generate training data to learn how to synthesize motion blur [15]. While these applications employ frame interpolation in time, it can also be used to synthesize novel views in space by interpolating between viewpoints [61].

Approaches for video frame interpolation can be categorized as flow-based, kernel-based, and phase-based. We adopt the flow-based paradigm since it has proven to work well in quantitative benchmarks [8]. One common approach for these methods is to estimate the optical flow $F_{t\to 0}$ and $F_{t\to 1}$ between two input frames $I_0$ and $I_1$ from the perspective of the frame $I_t$ that is ought to be synthesized. The interpolation result can then be obtained by backward warping $I_0$ according to $F_{t\to 0}$ and $I_1$ according to $F_{t\to 1}$ [57]. While it is intuitive, this approach makes it difficult to use an off-the-shelf optical flow estimator and prevents synthesizing frames at an arbitrary $t$ in a natural manner. To address these concerns, Jiang *et al.* [60] and Bao *et al.* [9] approximate $F_{t\to 0}$ and $F_{t\to 1}$ from $F_{0\to 1}$ and $F_{1\to 0}$.

Different from backward warping, Niklaus *et al.* [106] directly forward-warp $I_0$ according to $t \cdot F_{0\to 1}$ and $I_1$ according to $(1-t) \cdot F_{1\to 0}$, which avoids having to

SepConv - $\mathcal{L}_F$ Niklaus *et al.* [108] | ToFlow Xue *et al.* [165] | CyclicGen Liu *et al.* [85] | CtxSyn - $\mathcal{L}_F$ Niklaus *et al.* [106] | DAIN Bao *et al.* [9] | Ours - $\mathcal{L}_F$ (proposed)

Figure 3.1: A difficult example for frame interpolation. Our approach produces a high-quality result in spite of the flamingo leg that is subject to large motion.

approximate $F_{t\to 0}$ and $F_{t\to 1}$. Another aspect of their approach is to warp not only the images but also the corresponding context information, which a synthesis network can use to make better predictions. However, their forward warping uses the equivalent of z-buffering in order to handle cases where multiple source pixels map to the same target location. It is thus unclear how to fully differentiate this operation due to the z-buffering [104]. We propose softmax splatting to address this limitation, which allows us to jointly supervise all inputs to the forward warping. As a consequence, we are able to extend the idea of warping a generic context map to learning and warping a task-specific feature pyramid. Furthermore, we are able to supervise not only the optical flow estimator but also the metric that weights the importance of different pixels when they are warped to the same location. This approach, which is enabled by our proposed softmax splatting, achieves new state-of-the-art results and ranks first in the Middlebury benchmark for frame interpolation.

In short, we propose softmax splatting to perform differentiable forward warping and show its effectiveness on the application of frame interpolation. An interesting research question that softmax splatting addresses is how to handle different source pixels that map to the same target location in a differentiable way. Softmax splatting

enables us to train and use task-specific feature pyramids for image synthesis. Furthermore, softmax splatting not only allows us to fine-tune an off-the-shelf optical flow estimator for the task of video frame interpolation, it also enables us to supervise the metric that is used to disambiguate cases where multiple source pixels forward-warp to the same target location.

### 3.3 Video Frame Interpolation

Given two frames $I_0$ and $I_1$, frame interpolation aims to synthesize an intermediate frame $I_t$ where $t \in (0, 1)$ defines the desired temporal position. To address this problem, we first use an off-the-shelf optical flow method to estimate the optical flow $F_{0\to1}$ and $F_{1\to0}$ between the input frames in both directions. We then use forward warping in the form of softmax splatting $\overrightarrow{\sigma}$ to warp $I_0$ according to $F_{0\to t} = t \cdot F_{0\to1}$ and $I_1$ according to $F_{1\to t} = (1-t) \cdot F_{1\to0}$ as follows.

$$I_t \approx \overrightarrow{\sigma}\left(I_0, F_{0\to t}\right) = \overrightarrow{\sigma}\left(I_0, t \cdot F_{0\to1}\right) \tag{3.1}$$

$$I_t \approx \overrightarrow{\sigma}\left(I_1, F_{1\to t}\right) = \overrightarrow{\sigma}\left(I_1, (1-t) \cdot F_{1\to0}\right) \tag{3.2}$$

This is in contrast to backward warping $\overleftarrow{\omega}$, which would require $F_{t\to0}$ and $F_{t\to1}$ but computing this $t$-centric optical flow from $F_{0\to1}$ and $F_{1\to0}$ is complicated and subject to approximations [9]. We then combine these intermediate results to obtain $I_t$ using a synthesis network. More specifically, we not only warp the input frame in color- but also feature-space across multiple resolutions which enables the synthesis network to make better predictions.

We subsequently first introduce forward warping via softmax splatting and then show how it enables us to establish new state-of-the-art results for frame interpolation.

### 3.3.1 Forward Warping via Softmax Splatting

Backward warping is a common technique that has found broad adoption in tasks like unsupervised depth estimation or optical flow prediction [57]. It is well supported by many deep learning frameworks. In contrast, forward warping an image $I_0$ to $I_t$ according to $F_{0\to t}$ is not supported by these frameworks. We attribute this lack of support to the fact that there is no absolute way of performing forward warping. Forward warping is subject to multiple pixels in $I_0$ being able to map to the same target pixel in $I_t$ and there are various possibilities to address this ambiguity. We thus subsequently introduce common approaches to handle this mapping-ambiguity and discuss their limitations. We then propose softmax splatting which addresses these inherent limitations. Please note that we use the terms "forward warping" and "splatting" interchangeably.

**Summation splatting.** A straightforward approach of handling the aforementioned mapping-ambiguity is to sum all contributions. We define this summation splatting $\overrightarrow{\Sigma}$ as follows, where $I_t^{\Sigma}$ is the sum of all contributions from $I_0$ to $I_t$ according to $F_{0\to t}$ subject to the bilinear kernel $b$.

$$\text{let } \boldsymbol{u} = \boldsymbol{p} - \big(\boldsymbol{q} + F_{0\to t}[\boldsymbol{q}]\big) \tag{3.3}$$

$$b(\boldsymbol{u}) = \max(0, 1 - |\boldsymbol{u}_x|) \cdot \max(0, 1 - |\boldsymbol{u}_y|) \tag{3.4}$$

$$I_t^{\Sigma}[\boldsymbol{p}] = \sum_{\forall \boldsymbol{q} \in I_0} b(\boldsymbol{u}) \cdot I_0[\boldsymbol{q}] \tag{3.5}$$

$$\overrightarrow{\Sigma}\,(I_0, F_{0\to t}) = I_t^{\Sigma} \tag{3.6}$$

As shown in Figure 3.2, this summation splatting leads to brightness inconsistencies in overlapping regions like the front of the car. Furthermore, the bilinear kernel $b$ leads to pixels in $I_t$ that only receive partial contributions from the pixels in $I_0$ which

yet again leads to brightness inconsistencies like on the street. However, we use this summation splatting as the basis of all subsequent forward warping approaches. Its partial derivatives are as follows.

$$\frac{\partial I_t[\boldsymbol{p}]}{\partial I_0[\boldsymbol{q}]} = \frac{\partial I_t[\boldsymbol{p}]}{\partial I_t^{\Sigma}[\boldsymbol{p}]} \frac{\partial I_t^{\Sigma}[\boldsymbol{p}]}{\partial I_0[\boldsymbol{q}]} \tag{3.7}$$

$$\frac{\partial I_t[\boldsymbol{p}]}{\partial F_{0\to t}^{x}[\boldsymbol{q}]} = \frac{\partial I_t[\boldsymbol{p}]}{\partial I_t^{\Sigma}[\boldsymbol{p}]} \frac{\partial I_t^{\Sigma}[\boldsymbol{p}]}{\partial F_{0\to t}^{x}[\boldsymbol{q}]} \tag{3.8}$$

and analogous for the $y$ component of $F_{0\to t}$. It is not easy to obtain these through automatic differentiation since few frameworks support the underlying `scatter_nd` function. We hence provide the relevant derivatives as follows.

$$\text{let } \boldsymbol{u} = \boldsymbol{p} - \big(\boldsymbol{q} + F_{0\to t}[\boldsymbol{q}]\big) \tag{3.9}$$

$$\frac{\partial I_t^{\Sigma}[\boldsymbol{p}]}{\partial I_0[\boldsymbol{q}]} = b(\boldsymbol{u}) \tag{3.10}$$

$$\frac{\partial I_t^{\Sigma}[\boldsymbol{p}]}{\partial F_{0\to t}^{x}[\boldsymbol{q}]} = \frac{\partial b(\boldsymbol{u})}{\partial F_{0\to t}^{x}} \cdot I_0[\boldsymbol{q}] \tag{3.11}$$

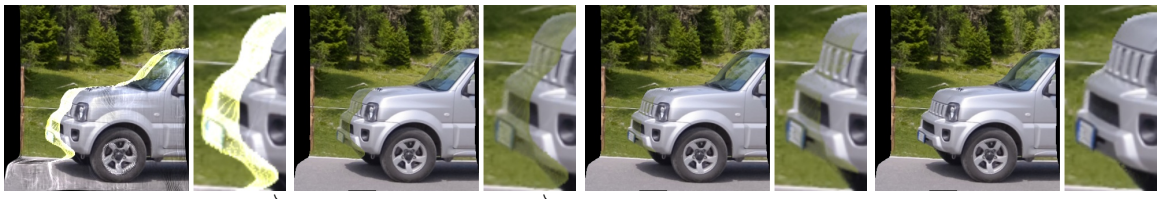
$$\frac{\partial b(\boldsymbol{u})}{\partial F_{0\to t}^{x}} = \max(0, 1 - |\boldsymbol{u}_y|) \cdot \begin{cases} 0, \text{ if } |\boldsymbol{u}_x| \geq 1 \\ -\operatorname{sgn}(\boldsymbol{u}_x), \text{ else} \end{cases} \tag{3.12}$$

and analogous for the $y$ component of $F_{0\to t}$. We will be providing a reference implementation of this summation splatting $\overrightarrow{\Sigma}$, which is written in CUDA for efficiency.

**Average splatting.** To address the brightness inconsistencies that occur with summation splatting, we need to normalize $I_t^{\Sigma}$. To do so, we can reuse the definition of $\overrightarrow{\Sigma}$ and determine average splatting $\overrightarrow{\Phi}$ as follows.

$$\overrightarrow{\Phi}\left(I_0, F_{0\to 1}\right) = \frac{\overrightarrow{\Sigma}\left(I_0, F_{0\to 1}\right)}{\overrightarrow{\Sigma}\left(\mathbf{1}, F_{0\to 1}\right)} \tag{3.13}$$

As shown in Figure 3.2, this approach handles the brightness inconsistencies and maintains the appearance of $I_0$. However, this technique averages overlapping regions like at the front of the car with the grass in the background.



summation splatting $\overrightarrow{\Sigma}$ average splatting $\overrightarrow{\Phi}$ linear splatting $\overrightarrow{*}$ softmax splatting $\overrightarrow{\sigma}$

Figure 3.2: Given two images $I_0$ and $I_1$ as well as an optical flow estimate $F_{0\to1}$, this figure shows an example of warping $I_0$ to $I_t$ according to $F_{0\to t} = t \cdot F_{0\to1}$ with four different forward warping approaches. The summation warping $\overrightarrow{\Sigma}$ handles cases where multiple pixels in $I_0$ map to the same target location in $I_t$ by taking their sum, which leads to brightness inconsistencies. The average warping $\overrightarrow{\Phi}$ takes their mean instead and is able to maintain the overall appearance of $I_0$ but blends overlapping regions. The linear splatting $\overrightarrow{*}$ weights the pixels in $I_0$ before warping them but still fails to clearly separate the front of the car from the grass in the background. In contrast, our proposed softmax splatting $\overrightarrow{\sigma}$ shows the expected behavior with the car correctly occluding the background.

**Linear splatting.** In an effort to separate overlapping regions, one could linearly weight $I_0$ by an importance mask $Z$ and define linear splatting $\overrightarrow{*}$ as follows.

$$\overrightarrow{*}\left(I_0, F_{0\to1}\right) = \frac{\overrightarrow{\Sigma}\left(Z \cdot I_0, F_{0\to1}\right)}{\overrightarrow{\Sigma}\left(Z, F_{0\to1}\right)} \tag{3.14}$$

where $Z$ could, for example, relate to the depth of each pixel [9]. As shown in Figure 3.2, this approach can better separate the front of the car from the grass in the background. It is not invariant to translations with respect to $Z$ though. If $Z$ represents the inverse depth then there will be a clear separation if the car is at $Z = 1/1$ and the grass in the background is at $Z = 1/10$. But, if the car is at $Z = 1/101$ and the grass in the background is at $Z = 1/110$ then they will be averaged again despite being equally far apart in terms of depth.

**Softmax splatting.** To clearly separate overlapping regions according to an importance mask $Z$ with translational invariance, we propose softmax splatting $\overrightarrow{\sigma}$.

$$\overrightarrow{\sigma}\left(I_0, F_{0\to1}\right) = \frac{\overrightarrow{\Sigma}\left(\exp(Z) \cdot I_0, F_{0\to1}\right)}{\overrightarrow{\Sigma}\left(\exp(Z), F_{0\to1}\right)} \tag{3.15}$$

where Z could, for example, relate to the depth of each pixel [9]. As shown in Figure 3.2,

this approach is able to clearly separate the front of the car from the background without any remaining traces of grass. Further, it shares resemblance to the softmax function and is hence invariant to translations $\beta$ with respect to $Z$ as follows.

$$\text{let } \boldsymbol{u} = \boldsymbol{p} - \big(\boldsymbol{q} + F_{0\to t}[\boldsymbol{q}]\big) \tag{3.16}$$

$$b(\boldsymbol{u}) = \max(0, 1 - |\boldsymbol{u}_x|) \cdot \max(0, 1 - |\boldsymbol{u}_y|) \tag{3.17}$$

$$I_t^{\sigma}[\boldsymbol{p}] = \frac{\sum_{\forall \boldsymbol{q} \in I_0} \exp(Z[\boldsymbol{q}] + \beta) \cdot b(\boldsymbol{u}) \cdot I_0[\boldsymbol{q}]}{\sum_{\forall \boldsymbol{q} \in I_0} \exp(Z[\boldsymbol{q}] + \beta) \cdot b(\boldsymbol{u})} \tag{3.18}$$

$$= \frac{\sum_{\forall \boldsymbol{q} \in I_0} \exp(Z[\boldsymbol{q}]) \cdot \exp(\beta) \cdot b(\boldsymbol{u}) \cdot I_0[\boldsymbol{q}]}{\sum_{\forall \boldsymbol{q} \in I_0} \exp(Z[\boldsymbol{q}]) \cdot \exp(\beta) \cdot b(\boldsymbol{u})} \tag{3.19}$$

$$= \frac{\sum_{\forall \boldsymbol{q} \in I_0} \exp(Z[\boldsymbol{q}]) \cdot b(\boldsymbol{u}) \cdot I_0[\boldsymbol{q}]}{\sum_{\forall \boldsymbol{q} \in I_0} \exp(Z[\boldsymbol{q}]) \cdot b(\boldsymbol{u})} \tag{3.20}$$

This property is important when mapping multiple pixels to the same location. If $Z$ represents depth, then the car and the grass in the background in Figure 3.2 are treated equally whether the car is at $Z = 1$ and the grass in the background is at $Z = 10$ or the car is at $Z = 101$ and the grass in the background is at $Z = 110$. It is not invariant to scale though and multiplying $Z$ by $\alpha$ will affect how well overlapping regions will be separated. This parameter can be learned via end-to-end training.

**Importance metric.** We use $Z$ to weight pixels in $I_0$ in order to resolve cases where multiple pixels from $I_0$ map to the same target pixel in $I_t$. This $Z$ could, for example, represent depth [9]. However, obtaining such a depth estimate is computationally expensive and inherently challenging which makes it prone to inaccuracies. We thus use brightness constancy as a measure of occlusion [8], which can be obtained via backward warping $\overleftarrow{\omega}$ as follows.

$$Z = \alpha \cdot \left\| I_0 - \overleftarrow{\omega}\left(I_1, F_{0\to 1}\right) \right\|_1 \tag{3.21}$$

Since our proposed softmax splatting is fully differentiable, we can not only learn $\alpha$

(initially set to $-1$) but also use a small neural network $\upsilon$ to further refine this metric.

$$Z = \upsilon\big(I_0, -\left\|I_0 - \overleftarrow{\omega}\left(I_1, F_{0\to1}\right)\right\|_1\big) \tag{3.22}$$

One could also obtain $Z$ directly from $\upsilon(I_0)$ but we were unable to make this $\upsilon$ converge. Lastly, when applying softmax splatting to tasks different from frame interpolation, the importance metric may be adjusted accordingly.

**Efficiency.** PyTorch's backward warping requires 1.1 ms to warp a full-HD image on a Titan X with a synthetic flow drawn from $\mathcal{N}(0, 10^2)$. In contrast, our implementation of softmax splatting requires 3.7 ms since we need to compute $Z$ and handle race conditions during warping.

### 3.3.2 Feature Pyramids for Image Synthesis

We adopt the video frame interpolation pipeline from Niklaus *et al.* [106] who, given two input frames $I_0$ and $I_1$, first estimate the inter-frame motion $F_{0\to1}$ and $F_{1\to0}$ using an off-the-shelf optical flow method. They then extract generic contextual information from the input images using a pre-defined filter $\psi$ and forward-warp $\overrightarrow{\omega}$ the images together with their context maps according to $t \cdot F_{0\to1} = F_{0\to t}$ and $(1-t) \cdot F_{1\to0} = F_{1\to t}$, before employing a synthesis network $\phi$ to obtain the interpolation result $I_t$.

$$I_t = \phi\Big(\overrightarrow{\omega}\left(\{I_0, \psi\left(I_0\right)\}, F_{0\to t}\right), \overrightarrow{\omega}\left(\{I_1, \psi\left(I_1\right)\}, F_{1\to t}\right)\Big)$$

This approach is conceptually simple and has been proven to work well. However, Niklaus *et al.* [106] were not able to supervise the context extractor $\psi$ and instead used `conv1` of ResNet-18 [45] due to the limitations of their forward warping $\overrightarrow{\omega}$ approach. This particular limitation makes it an ideal candidate to show the benefits of our proposed softmax splatting.

Our proposed softmax splatting allows us to supervise $\psi$, enabling it to learn to extract features that are important for image synthesis. Furthermore, we extend

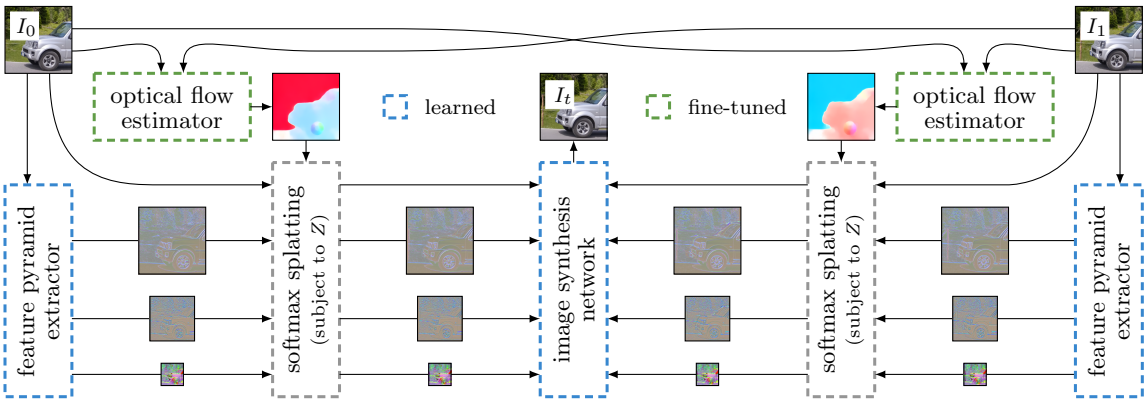



Figure 3.3: An overview of our frame interpolation framework. Given two input frames $I_0$ and $I_1$, we first estimate the bidirectional optical flow between them. We then extract their feature pyramids and forward-warp them together with the input frames to the target temporal position $t \in (0, 1)$ according to the optical flow. Using softmax splatting enables end-to-end training and thus allows the feature pyramid extractor to learn to gather features that are important for image synthesis. The warped input frames and feature pyramids are then fed to a synthesis network to generate the interpolation result $I_t$.

this idea by extracting and warping features at multiple scales in the form of feature pyramids. This allows the synthesis network $\phi$ to further improve its predictions. Please see Figure 3.3 for an overview of our video frame interpolation framework.

**Optical flow estimator.** We use an off-the-shelf optical flow method to make use of the ongoing achievements in research on correspondence estimation. Specifically, we use PWC-Net [141] and show that FlowNet2 [56] and LiteFlowNet [55] perform equally well within our evaluation. In accordance with the findings of Xue *et al.* [165], we additionally fine-tune PWC-Net for frame interpolation.

**Feature pyramid extractor.** The architecture of our feature pyramid extractor is shown in Figure 3.4. Our proposed softmax splatting enables us to supervise this feature pyramid extractor in an end-to-end manner, allowing it to learn to extract features that are useful for the subsequent image synthesis. As shown in our evaluation, this approach leads to significant improvements in the quality of the interpolation result. We also show that the interpolation quality degrades if we use fewer levels.

| type | features | kernel | stride | padding |
|---|---|---|---|---|
| Input | – | – | – | – |
| Conv2d | $3 \rightarrow 32$ | $3 \times 3$ | $1 \times 1$ | $1 \times 1$ |
| PReLU | – | – | – | – |
| Conv2d | $32 \rightarrow 32$ | $3 \times 3$ | $1 \times 1$ | $1 \times 1$ |
| PReLU | – | – | – | – |
| Conv2d | $32 \rightarrow 64$ | $3 \times 3$ | $2 \times 2$ | $1 \times 1$ |
| PReLU | – | – | – | – |
| Conv2d | $64 \rightarrow 64$ | $3 \times 3$ | $1 \times 1$ | $1 \times 1$ |
| PReLU | – | – | – | – |
| Conv2d | $64 \rightarrow 96$ | $3 \times 3$ | $2 \times 2$ | $1 \times 1$ |
| PReLU | – | – | – | – |
| Conv2d | $96 \rightarrow 96$ | $3 \times 3$ | $1 \times 1$ | $1 \times 1$ |
| PReLU | – | – | – | – |

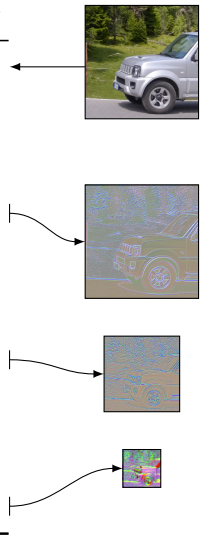


Figure 3.4: The architecture of our feature pyramid extractor. The feature visualization was obtained using PCA and is only serving an aesthetic purpose. See our evaluation for an analysis of the feature pyramid space for image synthesis.

**Image synthesis network.** The synthesis network generates the interpolation result guided by the warped input images and their corresponding feature pyramids. We employ a GridNet [34] architecture with three rows and six columns for this task. To avoid checkerboard artifacts [110], we adopt the modifications proposed by Niklaus *et al.* [106]. The GridNet architecture is a generalization of U-Nets and is thus well suited for the task of image synthesis.

**Importance metric.** Our proposed softmax splatting uses an importance metric $Z$ which is used to resolve cases where multiple pixels forward-warp to the same target location. We use brightness constancy to compute this metric as outlined in Section 3.3.1. Furthermore, we refine this occlusion estimate using a small U-Net consisting of three levels, which is trained end-to-end with the feature pyramid extractor and the image synthesis network.

**Training.** We adopt the training from Niklaus *et al.* [106]. We thus train two versions of our model to account for the perception-distortion tradeoff [12], one trained on color

loss $\mathcal{L}_{Lap}$ which performs well in standard benchmarks and one trained on perceptual loss $\mathcal{L}_F$ which retains more details in difficult cases. However, instead of using a proprietary training dataset, we use frame-triples from the training portion of the publicly available Vimeo-90k dataset [165].

**Efficiency.** With an Nvidia Titan X, we are able to synthesize a 720p frame in 0.357 seconds as well as a 1080p frame in 0.807 seconds. The parameters of our entire pipeline amount to 31 megabytes when stored.

### 3.4 Experiments

We evaluate our method, which utilizes softmax splatting to improve an existing frame interpolation approach, and compare it to state-of-the-art methods quantitatively and qualitatively on publicly available datasets. To support examining the visual quality of the frame interpolation results, we also provide a supplementary video.

**Methods.** We compare our approach to several state-of-the-art frame interpolation methods for which open source implementations from the respective authors are publicly available. This includes SepConv [108], ToFlow [165], CyclicGen [85], and DAIN [9]. We also include the closed source CtxSyn [106] approach wherever possible, with the results kindly being provided by its authors.

**Datasets.** We perform the quantitative evaluation on common datasets for frame interpolation. This includes the Vimeo-90k [165] test dataset as well as the samples from the Middlebury benchmark with publicly-available ground truth interpolation results [8]. When comparing our approach to other state-of-the-art methods, we additionally incorporate samples from UCF101 [86, 137] and Xiph [1] .

**Metrics.** We follow recent work on frame interpolation and use PSNR and SSIM [155]

[1] `https://media.xiph.org/video/derf/`

for all quantitative comparisons. We also incorporate the LPIPS [170] metric which strives to measure perceptual similarity. While higher values indicate better results with PSNR and SSIM, lower values indicate better results with the LPIPS metric.

### 3.4.1 Ablation Experiments

We show the effectiveness of our proposed softmax splatting by improving the context-aware frame interpolation from Niklaus *et al.* [106]. We thus not only need to compare softmax splatting to alternative ways of performing differentiable forward warping, we also need to analyze the improvements that softmax splatting enabled.

**Context-aware synthesis.** Since we adopt the framework of Niklaus *et al.* [106], we first need to verify that we can match their performance. We thus replace our feature pyramid extractor with the `conv1` layer of ResNet-18 [45] and we do not fine-tune the utilized PWC-Net for frame interpolation. This leaves the training dataset as well as the softmax splatting as the only significant differences. As shown in Table 3.1 (first section), our implementation performs slightly better in terms of PSNR on the Middlebury examples. It is significantly better in terms of PSNR on the Vimeo-90k test data though, but this is expected since we supervise on the Vimeo-90k training data. We can thus confirm that the basis for our approach replicates CtxSyn.

**Softmax splatting for frame interpolation.** We discussed various ways of performing differentiable forward warping in Section 3.3.1 and outlined their limitations. We then proposed softmax splatting to address these limitations. To analyze the effectiveness of softmax splatting, we train four versions of our approach, each one using a different forward warping technique. As shown in Table 3.1 (second section), summation splatting performs worst and softmax splatting performs best in terms of PSNR. Notice that the PSNR of average splatting is better than linear splatting

| | Vimeo-90k [165] | | | Middlebury [8] | | |
|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| CtxSyn | 34.39 | 0.961 | 0.024 | 36.93 | 0.964 | 0.016 |
| Ours - CtxSyn-like | 34.85 | 0.963 | 0.025 | 37.02 | 0.966 | 0.018 |
| Ours - summation splatting | 35.09 | 0.965 | 0.024 | 37.47 | 0.968 | 0.018 |
| Ours - average splatting | 35.29 | 0.966 | 0.023 | 37.53 | 0.969 | 0.017 |
| Ours - linear splatting | 35.26 | 0.966 | 0.024 | 37.73 | 0.968 | 0.017 |
| Ours - softmax splatting | 35.54 | 0.967 | 0.024 | 37.81 | 0.969 | 0.017 |
| Ours - pre-defined $Z$ | 35.54 | 0.967 | 0.024 | 37.81 | 0.969 | 0.017 |
| Ours - fine-tuned $Z$ | 35.59 | 0.967 | 0.024 | 37.97 | 0.970 | 0.017 |
| Ours - 1 feature level | 35.08 | 0.965 | 0.024 | 37.32 | 0.968 | 0.018 |
| Ours - 2 feature levels | 35.37 | 0.966 | 0.024 | 37.79 | 0.970 | 0.016 |
| Ours - 3 feature levels | 35.59 | 0.967 | 0.024 | 37.97 | 0.970 | 0.017 |
| Ours - 4 feature levels | 35.69 | 0.968 | 0.023 | 37.99 | 0.971 | 0.016 |
| Ours - FlowNet2 | 35.83 | 0.969 | 0.022 | 37.67 | 0.970 | 0.016 |
| Ours - LiteFlowNet | 35.59 | 0.968 | 0.024 | 37.83 | 0.970 | 0.017 |
| Ours - PWC-Net | 35.59 | 0.967 | 0.024 | 37.97 | 0.970 | 0.017 |
| Ours - PWC-Net-ft | 36.10 | 0.970 | 0.021 | 38.42 | 0.971 | 0.016 |
| Ours - $\mathcal{L}_{Lap}$ | 36.10 | 0.970 | 0.021 | 38.42 | 0.971 | 0.016 |
| Ours - $\mathcal{L}_{F}$ | 35.48 | 0.964 | 0.013 | 37.55 | 0.965 | 0.008 |

Table 3.1: Ablation experiments to quantitatively analyze the effect of the different components of our approach.

on the Middlebury examples but worse on the Vimeo-90k test data. We attribute this erratic behavior of linear splatting to its lack of translational invariance. These findings support the motivations behind our proposed softmax splatting.

**Importance metric.** Our proposed softmax splatting uses an importance metric $Z$ to resolve cases where multiple pixels forward-warp to the same target location. We use brightness constancy [8] to obtain this metric. Since softmax splatting is fully differentiable, we can use a small U-Net to fine-tune this metric which, as shown in Table 3.1 (third section), leads to slight improvements in terms of PSNR. This

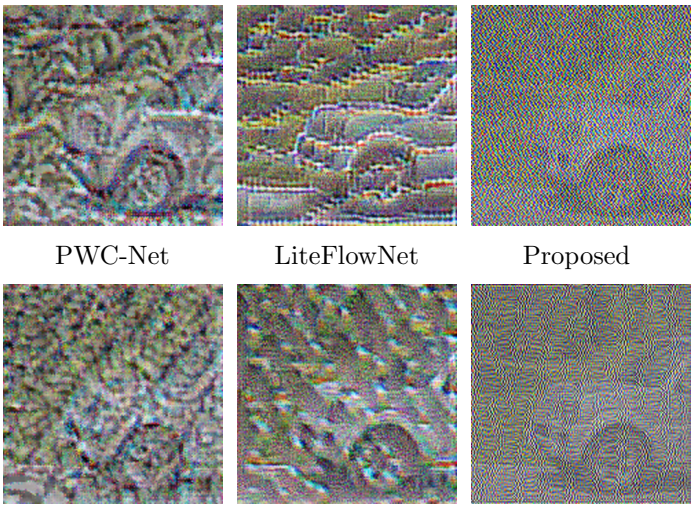



Figure 3.5: Feature response visualization for different task-specific feature pyramids using the image from Figure 3.2.

demonstrates that softmax splatting can effectively supervise $Z$ and that brightness constancy works well as the importance metric for video frame interpolation.

**Feature pyramids for image synthesis.** Softmax splatting enables us to synthesize images from warped feature pyramids, effectively extending the interpolation framework from Niklaus *et al.* [106]. In doing so, the softmax splatting enables end-to-end training of the feature pyramid extractor, allowing it to learn to gather features that are important for image synthesis. As shown in Table 3.1 (fourth section), the quality of the interpolation results improves when using more feature levels. Notice the diminishing returns when using more feature levels, with four levels of features overfitting on the Vimeo-90k dataset. We thus use three levels of features for our approach. To examine the difference between feature pyramids for frame interpolation and those for motion estimation by visualizing their feature responses [32]. Specifically, we maximize the activations of the last layer of our feature pyramid extractor as well as equivalent layers of PWC-Net [141] and LiteFlowNet [55] by altering the input image. Figure 3.5 shows feature activations, indicating that our feature pyramid focuses on

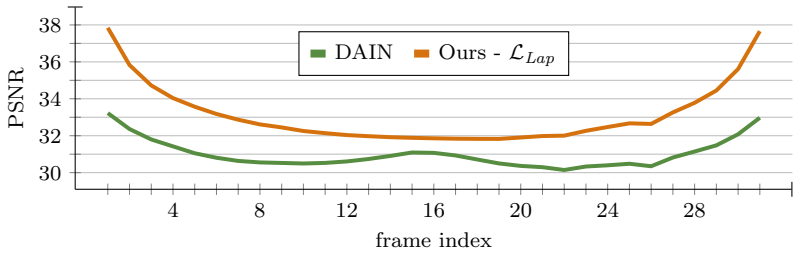



Figure 3.6: Assessment of the temporal consistency of our approach on the high frame-rate Sintel dataset [58].

fine details which are important to synthesize high-quality results while the feature pyramids for optical flow exhibit large patterns to account for large displacements.

**Optical flow estimation.** To analyze how well our approach performs with different optical flow methods, we consider three diverse state-of-the-art optical flow methods [55, 56, 141], each trained on FlyingChairs [28]. As shown in Table 3.1 (fifth section), they all perform similarly well. Due to softmax splatting being fully differentiable, we are further able to fine-tune the optical flow estimation for the task of frame interpolation [165]. Specifically, we fine-tune PWC-Net and see additional improvements with this PWC-Net-ft that has been optimized for the task of frame interpolation. We thus use PWC-Net-ft for our approach.

**Perception-distortion tradeoff.** We train two versions of our model, one trained on color loss and one trained on perceptual loss, in order to account for the perception-distortion tradeoff [12]. As shown in Table 3.1 (sixth section), the model trained using color loss $\mathcal{L}_{Lap}$ performs best in terms of PSNR and SSIM whereas the one trained using perceptual loss $\mathcal{L}_F$ performs best in terms of LPIPS. We note that the $\mathcal{L}_F$-trained model better recovers details in challenging cases, making it more practical.

**Temporal consistency.** Since we use forward warping to compensate for motion, we can interpolate frames at an arbitrary temporal position despite only supervising our model at $t = 0.5$. To analyze the temporal consistency of this approach, we perform

| | Vimeo-90k [165] | | | Middlebury [8] | | | UCF101 - DVF [86] | | | Xiph - 2K (4K resized to 2K) | | | Xiph - "4K" (2K crop from 4K) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| SepConv - $\mathcal{L}_1$ | 33.80 | 0.956 | 0.027 | 35.73 | 0.959 | 0.017 | 34.79 | 0.947 | 0.029 | 34.77 | 0.929 | 0.067 | 32.06 | 0.880 | 0.169 |
| SepConv - $\mathcal{L}_F$ | 33.45 | 0.951 | 0.019 | 35.03 | 0.954 | 0.013 | 34.69 | 0.945 | 0.024 | 34.47 | 0.921 | 0.041 | 31.68 | 0.863 | 0.097 |
| ToFlow | 33.73 | 0.952 | 0.027 | 35.29 | 0.956 | 0.024 | 34.58 | 0.947 | 0.027 | 33.93 | 0.922 | 0.061 | 30.74 | 0.856 | 0.132 |
| CyclicGen | 32.10 | 0.923 | 0.058 | 33.46 | 0.931 | 0.046 | 35.11 | 0.950 | 0.030 | 33.00 | 0.901 | 0.083 | 30.26 | 0.836 | 0.142 |
| CtxSyn - $\mathcal{L}_{Lap}$ | 34.39 | 0.961 | 0.024 | 36.93 | 0.964 | 0.016 | 34.62 | 0.949 | 0.031 | 35.71 | 0.936 | 0.073 | 32.98 | 0.890 | 0.175 |
| CtxSyn - $\mathcal{L}_F$ | 33.76 | 0.955 | 0.017 | 35.95 | 0.959 | 0.013 | 34.01 | 0.941 | 0.024 | 35.16 | 0.921 | 0.035 | 32.36 | 0.857 | 0.081 |
| DAIN | 34.70 | 0.964 | 0.022 | 36.70 | 0.965 | 0.017 | 35.00 | 0.950 | 0.028 | 35.95 | 0.940 | 0.084 | 33.49 | 0.895 | 0.170 |
| Ours - $\mathcal{L}_{Lap}$ | 36.10 | 0.970 | 0.021 | 38.42 | 0.971 | 0.016 | 35.39 | 0.952 | 0.033 | 36.62 | 0.944 | 0.107 | 33.60 | 0.901 | 0.234 |
| Ours - $\mathcal{L}_F$ | 35.48 | 0.964 | 0.013 | 37.55 | 0.965 | 0.008 | 35.10 | 0.948 | 0.022 | 35.74 | 0.921 | 0.029 | 32.50 | 0.856 | 0.071 |

Table 3.2: Quantitative comparison of various state-of-the-art methods across multiple datasets for video frame interpolation.

a benchmark on a high frame-rate version of the Sintel dataset [58]. Specifically, we interpolate frames 1 through 31 from frame 0 and frame 32 on all of its 13 scenes. We include DAIN for reference since it is also able to interpolate frames at an arbitrary $t$. As shown in Figure 3.6, DAIN degrades around frame 8 and frame 24 whereas our approach via softmax splatting does not.

### 3.4.2 Quantitative Evaluation

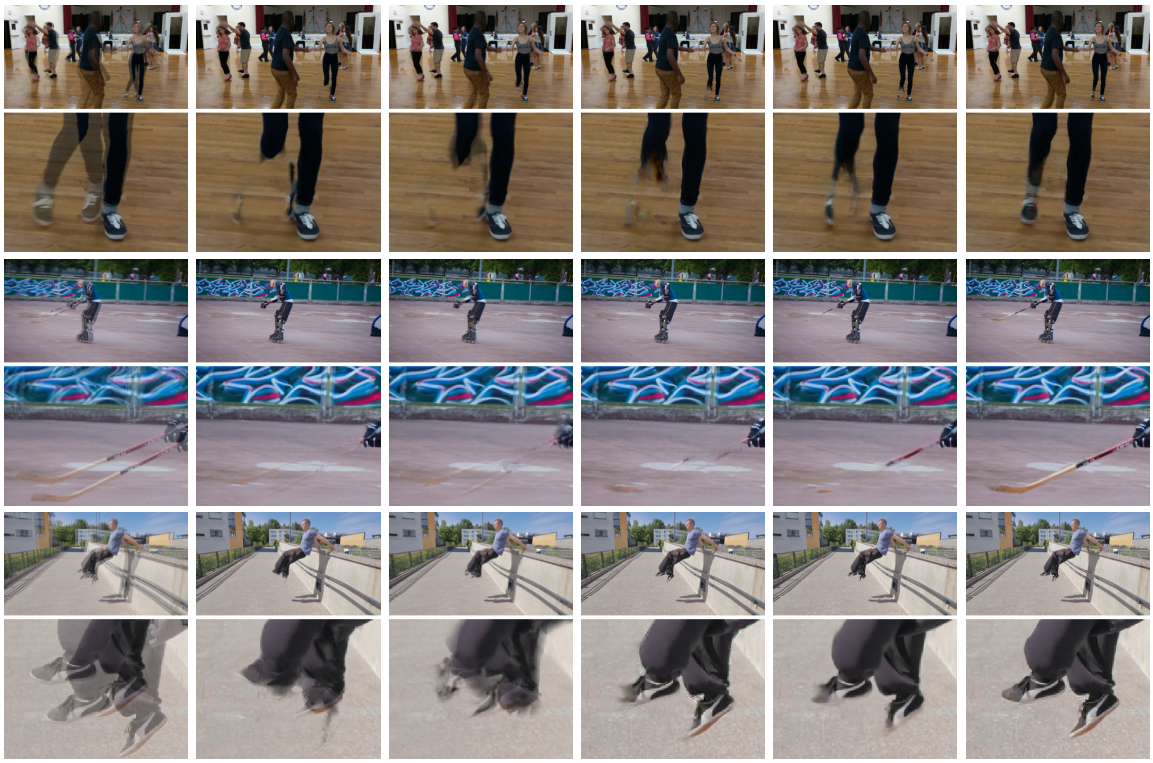
We compare our approach to state-of-the-art frame interpolation methods on common datasets. Since these datasets are all low resolution, we also incorporate 4K video clips from Xiph which are commonly used to assess video compression. Specifically, we selected the eight 4K clips with the most amount of inter-frame motion and extracted the first 100 frames from each clip. We then either resized the 4K frames to 2K or took a 2K center crop from them before interpolating the even frames from the odd ones. Since cropping preserves the inter-frame per-pixel motion, this "4K" approach allows us to approximate interpolating at 4K while actually interpolating

| | Boxing | | Crosswalk | | Driving | | Market-1 | | Market-2 | | Ritual | | Square | | Tango | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2K | "4K" | 2K | "4K" | 2K | "4K" | 2K | "4K" | 2K | "4K" | 2K | "4K" | 2K | "4K" | 2K | "4K" |
| | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ | PSNR ↑ |
| ToFlow | 36.75 | 33.52 | 33.54 | 31.42 | 34.84 | 33.38 | 30.87 | 29.41 | 34.24 | 30.30 | 28.11 | 22.61 | 38.87 | 36.59 | 34.24 | 28.67 |
| Sep. - $\mathcal{L}_1$ | 36.75 | 33.35 | 36.34 | 32.79 | 34.95 | 33.32 | 32.03 | 31.21 | 36.49 | 34.62 | 28.15 | 23.12 | 38.63 | 36.16 | 34.85 | 31.90 |
| Sep. - $\mathcal{L}_F$ | 36.54 | 33.15 | 35.69 | 32.10 | 34.84 | 33.24 | 31.69 | 30.68 | 36.26 | 34.27 | 27.95 | 23.06 | 38.41 | 35.84 | 34.43 | 31.08 |
| CyclicGen | 36.51 | 32.95 | 33.73 | 31.37 | 34.74 | 33.46 | 30.02 | 28.69 | 29.89 | 27.91 | 28.21 | 22.95 | 37.44 | 35.13 | 33.45 | 29.60 |
| DAIN | 37.74 | 34.75 | 38.81 | <u>35.90</u> | 35.14 | 33.60 | 33.06 | 31.99 | 38.03 | 36.49 | 29.16 | <u>23.91</u> | 39.50 | 37.00 | 36.14 | 34.28 |
| Ctx. - $\mathcal{L}_{L_1}$ | 37.41 | 33.55 | 38.14 | 34.03 | 34.92 | 32.71 | 32.93 | 31.97 | 38.33 | 37.18 | 28.47 | 23.08 | 39.35 | 37.09 | 36.16 | 34.21 |
| Ctx. - $\mathcal{L}_F$ | 36.68 | 32.88 | 37.40 | 33.01 | 34.56 | 32.45 | 32.20 | 31.10 | 37.94 | 36.62 | 28.24 | 23.10 | 38.87 | 36.61 | 35.36 | 33.10 |
| Ours - $\mathcal{L}_{L_1}$ | <u>38.44</u> | <u>35.44</u> | <u>38.93</u> | 34.33 | <u>35.69</u> | <u>33.82</u> | <u>33.31</u> | <u>32.37</u> | <u>39.58</u> | <u>38.02</u> | <u>29.43</u> | 23.83 | <u>40.90</u> | <u>37.96</u> | <u>36.86</u> | <u>34.58</u> |
| Ours - $\mathcal{L}_F$ | 37.48 | 34.40 | 37.82 | 33.47 | 35.14 | 33.27 | 32.20 | 31.18 | 39.07 | 37.01 | 29.15 | 23.72 | 40.35 | 37.13 | 35.89 | 33.27 |

Table 3.3: Quantitative comparison on the eight 4K clips from Xiph with the most amount of inter-frame motion.

at 2K instead. Directly processing 4K frames would have been unreasonable since DAIN, for example, already requires 16.7 gigabytes of memory to process 2K frames. In comparison, our approach only requires 5.9 gigabytes to process 2K frames which can be halved by using half-precision floating point operations.

As shown in Table 3.2, our $\mathcal{L}_{Lap}$-trained model outperforms all other methods in terms of PSNR and SSIM whereas our $\mathcal{L}_F$-trained model performs best in terms of LPIPS. Please note that on the Xiph dataset, all methods are subject to a significant degradation across all metrics when interpolating the "4K" frames instead of the ones that were resized to 2K. This shows that frame interpolation at high resolution remains a challenging problem. For completeness, we also show the per-clip metrics for the samples from Xiph in the supplementary material. We also submitted the results of our $\mathcal{L}_{Lap}$-trained model to the Middlebury benchmark [8]. Our approach currently ranks first in this benchmark and we provide the relevant results and accompanying screenshots in the supplementary material.



Overlayed ToFlow CyclicGen CtxSyn - $\mathcal{L}_F$ DAIN Ours - $\mathcal{L}_F$

Figure 3.7: Interpolation results for three difficult examples, comparing our approach with several state-of-the-art methods.

### 3.4.3 Qualitative Evaluation

Since videos are at the heart of our approach, we provide a qualitative comparison in the supplementary video. We additionally provide still results in Figure 3.7. These support the findings of the quantitative evaluation and show difficult examples where our proposed approach produces high-quality results whereas competing techniques are subject to visual artifacts.

### 3.4.4 Discussion

Our proposed softmax splatting enables us to extend and significantly improve the approach from Niklaus *et al.* [106]. Specifically, softmax splatting enables end-to-end training which allows us to not only employ and optimize feature pyramids for image

synthesis but also to fine-tune the optical flow estimator [165]. Our evaluation shows that these changes significantly improve the interpolation quality.

Another relevant approach is the one from Bao *et al.* [9]. They forward-warp the optical flow and subsequently backward warp the input images to the target location according to the warped optical flow. However, they use linear splatting and nearest neighbor interpolation. In comparison, our approach employs softmax splatting which is translational invariant and yields better results than linear splatting. Our approach is also conceptually simpler due to not warping the flow and not incorporating depth- or kernel-estimates. In spite of it simplicity, our approach compared favorably in the benchmark and is temporally consistent whereas DAIN was subject to degradations at $t = 0.25$ and $t = 0.75$.

The success of adversarial training as well as cycle consistency in image generation shows that more advanced ways of supervision can lead to high-quality synthesis results [40, 85, 125, 179]. While we consider such improvements orthogonal to this paper, we plan to explore these ideas to better supervise our model in future work.

## 3.5 Conclusion

In this paper, we presented softmax splatting for differentiable forward warping and demonstrated its effectiveness on the application of frame interpolation. The key research question that softmax splatting addresses is how to handle cases where different source pixels forward-warp to the same target location in a differentiable way. Further, we show that feature pyramids can successfully be employed for high-quality image synthesis, which is an aspect of feature pyramids that has not been explored yet. Our proposed frame interpolation pipeline, which is enabled by softmax splatting and conceptually simple, compares favorably in benchmarks and achieves new state-of-the-art results.

# 4 Novel View Synthesis in Space

This chapter has been adapted from a SIGGRAPH Asia paper. All uses of "we" or "our" refer to the authors of this paper (Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu). Simon's contributions are: depth estimation for novel view synthesis, context-aware color- and depth-inpainting, the design of the user interface, the computer-generated dataset, the architecture details such as using the GridNet architecture, all implementation aspects, the design and execution of the evaluation except the usability study, and the paper writing except the related work section.

## 4.1 Abstract

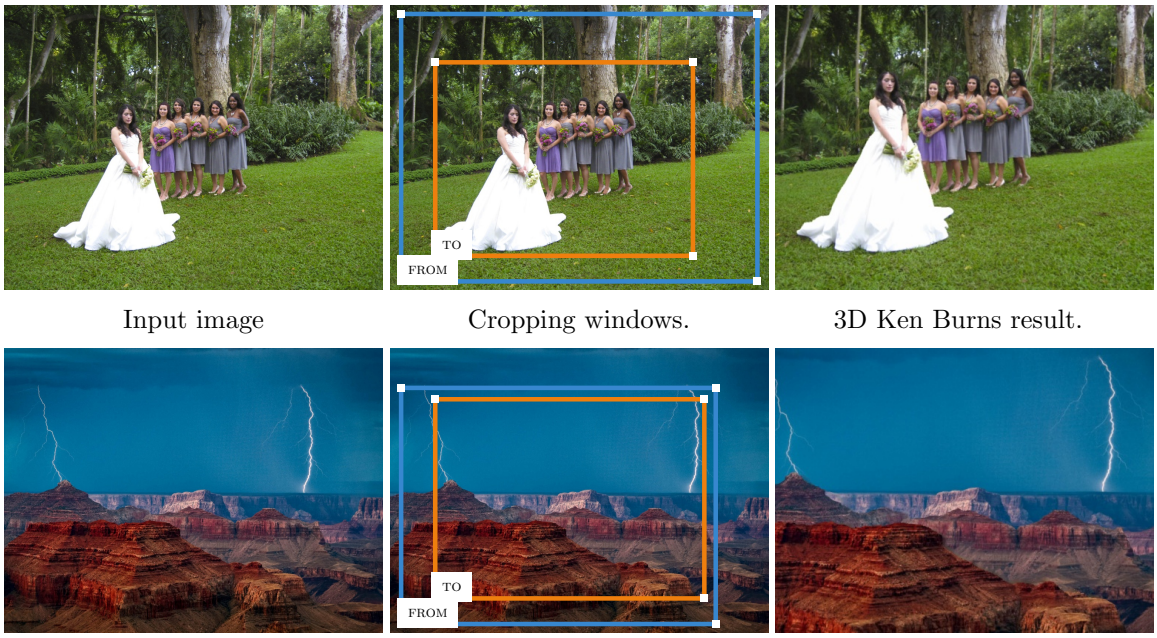
The Ken Burns effect allows animating still images with a virtual camera scan and zoom. Adding parallax, which results in the 3D Ken Burns effect, enables significantly more compelling results. Creating such effects manually is time-consuming and demands sophisticated editing skills. Existing automatic methods, however, require multiple input images from varying viewpoints. In this paper, we introduce a framework that synthesizes the 3D Ken Burns effect from a single image, supporting both a fully automatic mode and an interactive mode with the user controlling the camera. Our framework first leverages a depth prediction pipeline, which estimates scene depth that is suitable for view synthesis tasks. To address the limitations of existing depth estimation methods such as geometric distortions, semantic distortions, and inaccurate depth boundaries, we develop a semantic-aware neural network for depth

prediction, couple its estimate with a segmentation-based depth adjustment process, and employ a refinement neural network that facilitates accurate depth predictions at object boundaries. According to this depth estimate, our framework then maps the input image to a point cloud and synthesizes the resulting video frames by rendering the point cloud from the corresponding camera positions. To address disocclusions while maintaining geometrically and temporally coherent synthesis results, we utilize context-aware color- and depth-inpainting to fill in the missing information in the extreme views of the camera path, thus extending the scene geometry of the point cloud. Experiments with a wide variety of image content show that our method enables realistic synthesis results. Our study demonstrates that our system allows users to achieve better results while requiring little effort compared to existing solutions for the 3D Ken Burns effect creation.

## 4.2 Introduction

Advanced image- and video-editing tools allow artists to freely augment photos with depth information and to animate virtual cameras, enabling motion parallax as the camera scans over a still scene. This cinematic effect, which we refer to as 3D Ken Burns effect, has become increasingly popular in documentaries, commercials, and other media. Compared to the traditional Ken Burns effect which animates images with 2D scan and zoom [1] , this 3D counterpart enables much more compelling experiences. However, creating such effects from a single image is painstakingly difficult: The photo must be manually separated into different segments, which then have to carefully be arranged in the virtual 3D space, and inpainting needs to be performed to avoid holes when the virtual camera moves away from its origin. In this paper, we target the problem of automatically synthesizing the 3D Ken Burns

[1] `http://en.wikipedia.org/wiki/Ken_Burns_effect`




Input image | Cropping windows. | 3D Ken Burns result.

Figure 4.1: 3D Ken Burns effect from a single image. Given a single input image and optional user annotations in form of two cropping windows, our framework animates the input image while adding parallax to synthesize a 3D Ken Burns effect. Our method works well for a wide variety of content, including portrait (top) and landscape (bottom) photos. Please refer to our video demo to examine these examples.

effect from a single image. We further optionally incorporate simple user-specified camera paths, parameterized by the desired start- and end-view, to grant the user more control over the resulting effect as shown in Figure 4.1.

This problem of synthesizing realistic moving-camera effects from a single image is highly challenging. Two fundamental concerns need to be addressed. First, to synthesize a new view from a novel camera position, the scene geometry of the original view needs to be recovered accurately. Second, from the predicted scene geometry, a temporally consistent sequence of novel views has to be synthesized which requires dealing with disocclusion. We address both challenges and provide a complete system that enables synthesizing the 3D Ken Burns effect from a single image.

To synthesize the 3D Ken Burns effect, our method first estimates the depth map from the input image. While existing depth prediction methods have rapidly improved

over the past few years, monocular depth estimation remains an open problem. We observed that existing depth prediction methods are not particularly suitable for view synthesis tasks such as ours. Specifically, we identified three critical issues of existing depth prediction methods that need to be addressed to make them applicable to 3D Ken Burns synthesis: geometric distortions, semantic distortions, and inaccurate depth boundaries. Based on this observation, we designed a depth estimation pipeline along with the training framework dedicated to addressing these issues. To this end, we developed a semantic-aware neural network for depth estimation and train the network on our newly constructed large-scale synthetic dataset which contains accurate ground truth depth of various photo-realistic scenes.

From the input image and the associated depth map, a sequence of novel views has to be synthesized to produce an output video for the 3D Ken Burns effect. The synthesis process needs to handle three requirements. First, as the camera moves away from its original position, disocclusion necessarily happens. The missing information needs to be filled-in with geometrically consistent content. Second, the novel view renderings need to be synthesized in a temporally consistent manner. The straightforward approach of filling-in the missing information and synthesizing each view independently is not only computationally inefficient but also temporally unstable. Third, we have found that professional artists that use our system manually produce the most compelling effects when they are able to immediately perceive the result of their interaction. The synthesis thus needs to be real-time in order to best support such users. To address these requirements, we propose a simple yet effective solution: We map the input image to points in a point cloud according to the estimated depth. We then perform color- and depth-inpainting of novel view renderings at extreme views like at the beginning and at the end of the virtual camera path. This allows us to extend the point cloud with geometrically sound information. The extended

point cloud can then be used to synthesize all novel view renderings in an efficient and temporally consistent manner.

Together, our depth prediction pipeline and novel view synthesis approach provide a complete system for generating the 3D Ken Burns effect from a single image. This system provides a fully automatic solution where the start- and end-view of the virtual camera path are automatically determined so as to minimize the amount of disocclusion. In addition to the fully automatic mode, our system also provides an interactive mode in which users can control the start- and end-view through an intuitive user interface. This allows a more fine-grained control over the resulting 3D Ken Burns effect, thus supporting users in their artistic freedom.

The key contributions of this paper are as follows. We introduce the problem of 3D Ken Burns synthesis from a single image which enables automatic video generation in the form of a moving-camera effect. We leverage existing computer vision technologies and augment them to achieve plausible synthesis results. Our system offers a fully automatic mode which generates a convincing effect without any user feedback, and a view control mode which allows users to control the effect with simple interactions. Experiments on a wide range of real-world imagery demonstrate the effectiveness of our system. Our study shows that our system enables users to achieve better results while requiring little effort compared to existing solutions.

## 4.3 3D Ken Burns Effect Synthesis

Our framework consists of two main components, namely the depth estimation pipeline (Figure 4.3), and the novel view synthesis pipeline (Figure 4.7). In this section, we describe each component in detail.

### 4.3.1 Semantic-aware Depth Estimation

To synthesize the 3D Ken Burns effect, our method first estimates the depth of the input image. While recent advanced methods for monocular depth estimation have shown good performance on public benchmarks, we observed that their predictions are at times not suitable to produce high-quality view synthesis results. In particular, there are at least three major issues when applying existing depth estimation methods to generate the 3D Ken Burns effect:

1. *Geometric distortions.* While state-of-the-art depth estimation methods can generate reasonable depth orderings, they often have difficulty in capturing geometric relations such as planarity. Geometric distortion, such as bending planes, thus often appear in the synthesis results (Figure 4.2, top row).

2. *Semantic distortions.* Existing depth estimation methods predict the depth maps without explicitly taking the semantics of objects into account. Therefore, in many cases the depth values are assigned inconsistently inside regions of the same object, resulting in unnatural synthesis results such as objects sticking to the ground plane or different parts of an object being torn apart (Figure 4.2, bottom row).

3. *Inaccurate depth boundaries.* Current state-of-the-art methods for single-image depth estimation process the input image at a low resolution and utilize bilinear interpolation to obtain the full-resolution depth estimate. They are thus unable to accurately capture depth boundaries, resulting in artifacts in the novel view renderings (Figure 4.5).

In this paper, we design a semantic-aware depth estimation dedicated to addressing these issues. To do so, we separate the depth estimation into three steps. First,

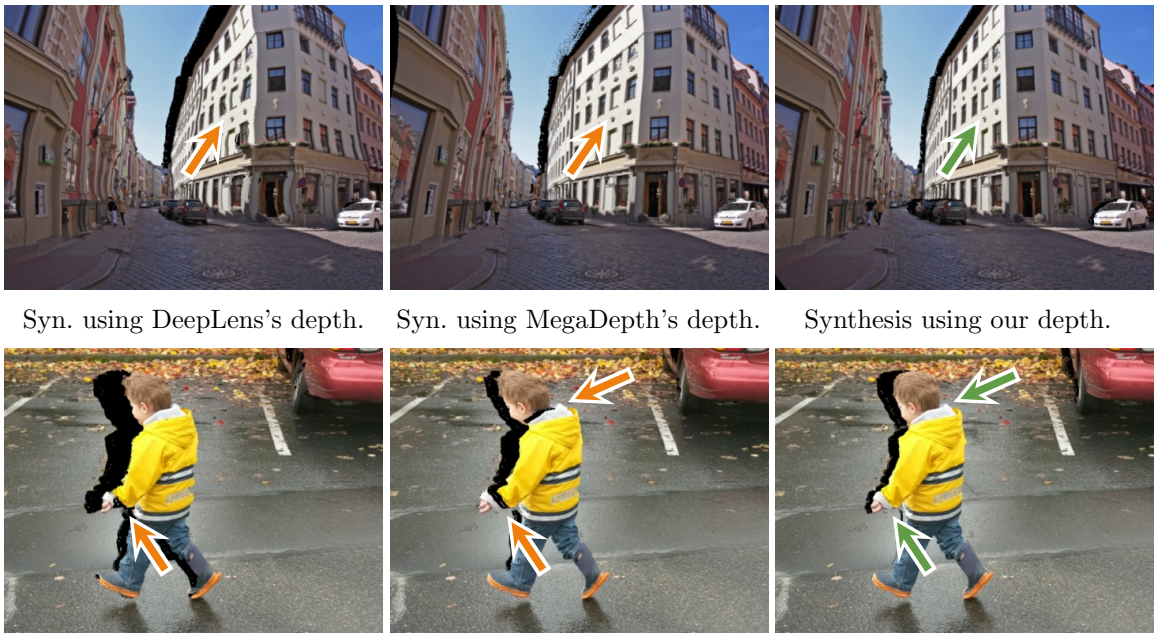



Figure 4.2: Geometric- and semantic-distortion examples resulting from off-the-shelf depth estimation methods. These images were synthesized by moving a virtual camera left and right. To focus the comparison on the depth estimate quality, we do not show our final synthesis result and instead only show the intermediate point-cloud rendering that are subject to disocclusion. In the first row, DeepLens and MegaDepth are subject to geometric distortions in the white building. In the second row, DeepLens and MegaDepth are subject to semantic distortions and are inconsistent with respect to the hand of the boy. Furthermore, MegaDepth's depth prediction also separates the head of the boy from the rest of the body.

estimating coarse depth using a low-resolution image while relying on semantic information extracted using VGG-19 [134] to facilitate generalizability. Second, adjusting the depth map according to the instance-level segmentation of Mask R-CNN [44] to ensure consistent depth values for salient objects. Third, refining the depth boundaries guided by the input image while upsampling the low-resolution depth estimate. Our depth estimation pipeline is shown in Figure 4.3 and subsequently elaborated.

#### 4.3.1.1 Depth Estimation

Following existing work on monocular depth estimation, we leverage a neural network to predict a coarse depth map. To facilitate a semantic-aware depth prediction, we

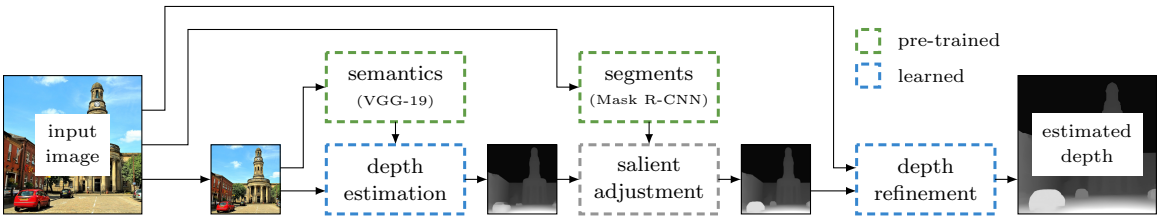



Figure 4.3: Overview of our depth estimation pipeline. Given a high-resolution image, we start by estimating a coarse depth based on a low-resolution input image. This depth estimation network is guided by semantic information extracted using VGG-19 [134] and supervised on a computer-generated dataset with accurate ground truth depth in order to facilitate geometrically sound predictions. To avoid semantic distortions, we then adjust the depth map according to the segmentation of Mask R-CNN [44] and make sure that each salient object is mapped to a coherent plane. Lastly, we utilize a depth refinement network that, guided by the input image, upsamples the coarse depth and ensures accurate depth boundaries.

further provide semantic guidance by augmenting the input of our network with the feature maps extracted from the `pool_4` layer of VGG-19 [134]. We found that granting explicit access to this semantic information encourages the network to better capture the geometry of large scene structures, thus addressing the concern of geometric distortions. Different from existing work, we do not resize the input image to a fixed resolution when providing it to the network and instead resize it such that its largest dimension is 512 pixels while preserving its aspect ratio.

*Architecture.* We employ a GridNet [34] architecture with the modifications proposed by Niklaus *et al.* [106] to prevent checkerboard artifacts [110]. We incorporate this grid architecture with a configuration of six rows and four columns, where the first two columns perform downsampling and the last two columns perform upsampling. This multi-path GridNet architecture allows the network to effectively combine feature representations from multiple scales. We feed the input image into the first row, while inserting the semantic features from VGG-19 into the fourth row of the grid. We explicitly encourage the network to focus more on the semantic features and less on the input image by letting the first three rows of the grid (corresponding to the input

image) have a channel size of 32, 48, and 64 respectively while the fourth through sixth row (corresponding to the semantic features) have 512 channels each. As such, a majority of the parameters reside in the bottom half of the network, forcing it to heavily make use of semantic features and in-turn supporting the generalization capability of our depth estimation network.

*Loss Functions.* To train our depth estimation network, we adopt the pixel-wise $\ell_1$ as well as the scale invariant gradient loss proposed by Ummenhofer *et al.* [149] to emphasize depth discontinuities. Specifically, given the ground truth inverse depth $\hat{\xi}$, we supervise the estimated inverse depth $\xi$ using the $\ell_1$-based loss as

$$\mathcal{L}_{\text{ord}} = \sum_{i,j} \left\| \xi(i,j) - \hat{\xi}(i,j) \right\|_1 \tag{4.1}$$

Similar to Ummenhofer *et al.* [149], we encourage more pronounced depth discontinuities and stimulate smoothness in homogeneous regions by incorporating a scale invariant gradient loss as

$$\mathcal{L}_{\text{grad}} = \sum_{h \in \{1,2,4,8,16\}} \sum_{i,j} \left\| \mathbf{g}_{\mathbf{h}}[\xi](i,j) - \mathbf{g}_{\mathbf{h}}[\hat{\xi}](i,j) \right\|_2 \tag{4.2}$$

where the discrete scale invariant gradient $\mathbf{g}$ is defined as

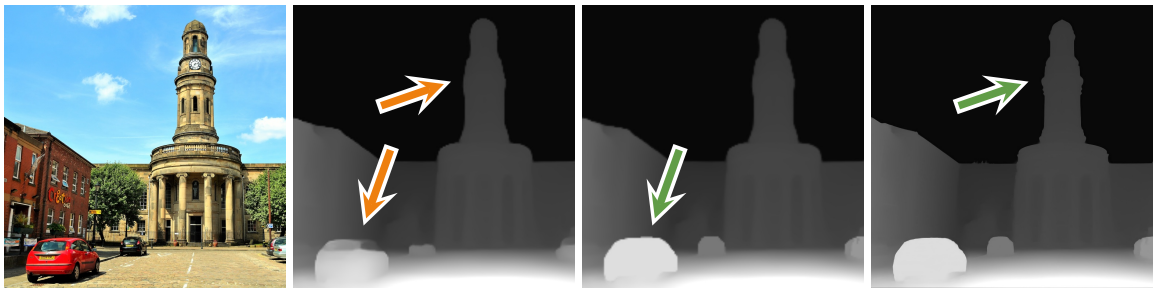
$$\mathbf{g}_{\mathbf{h}}[f](i,j) = \left( \frac{f(i+h,j) - f(i,j)}{|f(i+h,j)| + |f(i,j)|}, \frac{f(i,j+h) - f(i,j)}{|f(i,j+h)| + |f(i,j)|} \right)^\top \tag{4.3}$$

We emphasize the scale invariant gradient loss when training our depth estimation network and combine the two losses as

$$\mathcal{L}_{\text{depth}} = 0.0001 \cdot \mathcal{L}_{\text{ord}} + \mathcal{L}_{\text{grad}} \tag{4.4}$$

As such, we encourage accurate depth boundaries which are important for the resulting quality when synthesizing the 3D Ken Burns effect.

*Training.* We utilize Adam [66] with $\alpha = 0.0001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ and train our depth estimation network for $3 \cdot 10^6$ iterations. We incorporate 13017



Input image. Initial estimate. Adjusted depth. Refined depth.

Figure 4.4: Intermediate depth estimation results. This example demonstrates the contribution of each stage in our depth estimation pipeline. The initially estimated depth is subject to semantic distortion with respect to the red car and has inaccurate depth boundaries, for example, at the masonry of the tower. The depth adjustment addresses the semantic distortion of the red car, while the depth refinement addresses the fine details at object boundaries.

samples from the raw dataset of NYU v2 [133] together with 8685 samples from MegaDepth [80]. Since these datasets are subject to noise and an inaccurate depth at object boundaries, we also leverage our own dataset which is described in Section 4.3.4. Our dataset consists of realistic renderings which provide high-quality depth maps with clear discontinuities at object boundaries.

#### 4.3.1.2 Depth Adjustment

We have found that our depth prediction network augmented with semantic features and trained using our high-quality dataset significantly improves the scene geometry represented by the estimate depth. However, semantic distortions have not been entirely resolved. It is extremely challenging to obtain accurate object-level depth predictions as the neural network not only needs to reason about the boundary of each object but also needs to determine the geometric relationship between different parts of an object. One approach to address this problem is to either provide semantic labels as input to the depth estimation network, or to train the depth estimation network in a multi-task setting to jointly predict segmentation masks [29, 82, 99, 102]

Cropped input image. Using initial depth. Using refined depth. Refined and z-filtered.

Figure 4.5: Example of our point cloud rendering. Using the point cloud of the initial depth estimate exemplifies the importance of our depth refinement, as objects may otherwise be torn apart at the object boundaries. We further note that moving the virtual camera forward may lead to cracks through which occluded background points may erroneously become visible (note the blue grid pattern on the tower), which we successfully address through z-filtering.

which would encourage the network to reason about object boundaries.

In contrast, we borrow a technique frequently employed by artists when creating the 3D Ken Burns effect manually: Identify the object segments and approximate each object with a frontal plane positioned upright on the ground plane. We mimic this practice and utilize instance-level segmentation masks from Mask R-CNN [44] for this purpose. Specifically, we select the masks of semantically important objects such as humans, cars, and animals and adjust the estimated depth values by assigning the smallest depth value from the bottom of the salient object to the entire mask. We note that this approximation is not physically correct. However, it is effective in producing perceptually plausible results for a majority of content as demonstrated by many artist-created results.

#### 4.3.1.3 Depth Refinement

So far, our depth estimation network is designed to reduce geometric distortions with the depth adjustment addressing semantic distortions. However, the resulting depth estimate is of low resolution and may be erroneous at boundary regions. One possible

solution to this problem is to apply joint bilateral filtering to upsample the depth map. However, this does not work well in our case. As also observed in previous work [78], we found that the texture of the guiding image tends to be transferred to the upsampled depth. In this work, we thus instead employ a neural network that, guided by a high-resolution image, learns how to perform depth upsampling that is subject to erroneous estimates at object boundaries. During inference, this model predicts the refined depth map at an aspect-dependent resolution with the largest dimension being 1024 pixels. This upscaling factor can further be increased by modifying the neural network accordingly.

*Architecture.* We insert the input image into a U-Net with three downsampling blocks which use strided convolutions and three corresponding upsampling blocks which use convolutions and bilinear upsampling. We insert the estimated depth at the bottom of the U-Net, allowing the network to learn how to downsample the input image in order to guide the depth during upsampling.

*Loss Functions.* Like with our depth estimation network, we encourage accurate predictions at object boundaries as well as smoothness in homogeneous regions and employ the same $\mathcal{L}_{\text{depth}}$ loss when training our refinement network.

*Training.* We utilize Adam [66] with $\alpha = 0.0001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ and train our depth refinement network for $1 \cdot 10^6$ iterations. Since accurate ground truth depth boundaries are crucial for training this network, we only use our computer-generated dataset which is described in Section 4.3.4. Specifically, we downsample and distort the ground truth depth to simulate the coarse depth prediction and use it, together with the high-resolution image, as inputs to the depth refinement network.

#### 4.3.1.4 Summary

Our depth estimation pipeline is designed to address each of the identified issues that are important when using depth estimation methods to create the 3D Ken Burns effect: geometric distortions, semantic distortions, and inaccurate depth boundaries. Please see Figure 4.4 which demonstrates the contribution of each step in our pipeline to the final depth estimate.

### 4.3.2 Context-aware Inpainting for View Synthesis

To synthesize the 3D Ken Burns effect from the estimated depth, our method first maps the input image to points in a point cloud. Each frame of the resulting video can then be synthesized by rendering the point cloud from the corresponding camera position along a pre-determined camera path. The point cloud, however, is only a partial view of the world geometry as seen from the input image. Therefore, the resulting novel view renderings are incomplete with holes caused by disocclusion. One possible solution is to utilize off-the-shelf image inpainting methods to fill-in the missing areas in each synthesized video frame. This approach, however, fails to satisfy the following requirements:

1. *Geometrically consistent inpainting.* Due to the nature of disocclusion, the filled-in area should resemble the background with a clear separation of the foreground object. Existing off-the-shelf inpainting methods do not explicitly reason about the geometry of the inpainting result though, which is why they are unable to satisfy this requirement (Figure 4.6).

2. *Temporal consistency.* When rendering multiple novel views to generate a moving-camera effect, the result needs to be temporally consistent. The traditional inpainting formulation does not consider our given scenario, which is why

Without inpainting. DeepFill inpainting. EdgeConnect inpainting. Proposed inpainting.

Figure 4.6: Example synthesis results, comparing two popular off-the-shelf inpainting methods with our approach. DeepFill fails to inpaint a plausible result due to the non-rectangular nature of the area that is ought to be inpainted. EdgeConnect inpaints a more plausible result but is not temporally consistent and fails to preserve the object boundary. In contrast, our inpainting approach is both temporally consistent and maintains a clear object boundary.

independently applying an existing off-the-shelf inpainting method is subject to temporal inconsistencies (Figure 4.6).

3. *Real-time synthesis.* When manually specifying the camera path for the 3D Ken Burns effect, we found that the best user experience is achieved when users can immediately perceive the result and make adjustments accordingly. Applying off-the-shelf inpainting methods in a frame-by-frame manner would be too slow to adequately support this use case scenario (Section 4.3.3).

In this paper, we design a dedicated view synthesis pipeline to address these requirements as illustrated in Figure 4.7. Given the point cloud obtained from the input image and its depth estimate, we perform joint color- and depth-inpainting to fill-in missing areas in incomplete novel view renderings. Having the inpainting method also incorporate depth enables geometrically consistent inpainting. The inpainted depth can then be used to map the inpainted color to new points in the existing point cloud, addressing the problem of disocclusion. To synthesize the 3D Ken Burns effect along a pre-determined camera path, it is in this regard sufficient to perform the color-

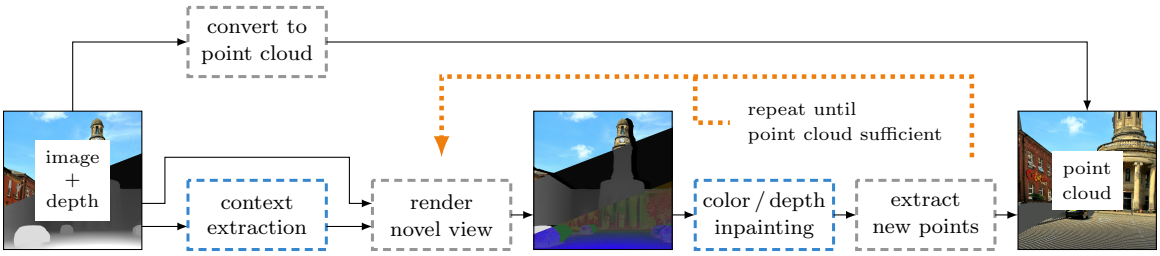



Figure 4.7: Overview of our novel view synthesis approach. From the point cloud obtained from the input image and the estimated depth map, we render consecutive novel views from new camera positions. This point cloud is only a partial view of the world geometry though, which is why novel view renderings will be subject to disocclusion. To address this issue, we perform geometrically consistent color- and depth-inpainting to recover a complete novel view from an incomplete render where each pixel contains color-, depth-, and context-information. The inpainted depth can then be used to map the inpainted color to new points in the existing point cloud. By repeating this procedure until the point cloud has been extended sufficiently, it is possible to render complete and temporally consistent novel views in real time. To synthesize the 3D Ken Burns effect along a camera path, it is in this regard sufficient to perform the color- and depth-inpainting only at extreme views.

and depth-inpainting only at extreme views like at the beginning and at the end. Rendering this extended point cloud preserves temporal consistency and can be done in real-time. To enable real-time synthesis when having an artist specify an arbitrary camera path, we repeat this procedure at extreme views to the left, right, top, and bottom. Our synthesis approach is illustrated in Figure 4.7 and we subsequently elaborate the involved steps.

#### 4.3.2.1 Point Cloud Rendering

We obtain novel view renderings by projecting the point cloud to an image plane subject to the pinhole camera model. In doing so, we utilize a z-buffer to correctly address occlusion. When moving the virtual camera forward, the point cloud rendering may, however, suffer from shine-through artifacts in which occluded background points becomes visible in foreground regions. Tulsiani *et al.* [148] address these artifacts by rendering the point cloud at half the input resolution. In order to preserve the image

resolution, we instead heuristically filter the z-buffer before projecting the points to the image plane. Specifically, we identify shined-through artifact regions by identifying pixels for which two adjacently opposing neighbors are significantly closer to the virtual camera. We then fill the cracks in the z-buffer with the average depth of the neighboring foreground pixels.

#### 4.3.2.2 Context Extraction

Niklaus *et al.* [106] observed that incorporating contextual information is beneficial for generating high-quality novel view synthesis results. Specifically, each point in the point cloud can be extended with contextual information that describes the neighborhood of where the corresponding pixel used to be in the input image. This augments the point cloud with rich information that can, for example, be leveraged for computer graphics in the form of neural rendering [3, 16, 94]. To make use of this technique, we leverage a neural network with two convolutional layers to extract 64 channels of context information from the input image. We train this context extractor jointly with the inpainting network, which allows the extractor to learn how to gather information that is useful when inpainting incomplete novel view renderings.

#### 4.3.2.3 Color- and Depth-inpainting

Different from existing image inpainting methods, our inpainting network accepts color-, depth-, and context-information as input and performs joint color- and depth-inpainting. The additional context provides rich information that is beneficial for high-quality image synthesis while the depth enables geometrically consistent inpainting results with foreground objects clearly being separated from the background. Specifically, we render the color-, depth-, and context-information of the input image to a novel view that is incomplete due to disocclusion. We then use our color- and

depth-inpainting network to fill-in missing areas. The inpainted depth allows us to map the inpainted color to new points in the existing point cloud, effectively extending the world geometry that the point cloud represents.

*Architecture.* Similarly to our depth estimation network, we employ a GridNet [34] architecture for our inpainting network due to its ability to learn how to combine representations at multiple scales. Specifically, we utilize a grid with four rows and four columns with a per-row channel size of 32, 64, 128, and 256 respectively. It accepts the color, depth, and context of the incomplete novel view rendering and returns the inpainted color and depth.

*Loss Functions.* We adopt a pixel-wise $\ell_1$ loss as well as a perceptual loss based on deep image features to supervise the color inpainting. Specifically, given a ground truth novel view $I_{gt}$, we supervise the inpainted color $I$ using the $\ell_1$-based loss as

$$\mathcal{L}_{\text{color}} = \|I - I_{gt}\|_1 \tag{4.5}$$

For the perceptual loss, we employ a content loss based on the difference between deep image features as

$$\mathcal{L}_{\text{percep}} = \|\phi(I) - \phi(I_{gt})\|_2^2 \tag{4.6}$$

where $\phi$ represents feature activations from a generic image classification network. Specifically, we use the activations of the `relu4_4` layer from VGG-19 [134]. To supervise the depth-inpainting, we use the $\ell_1$-based loss $\mathcal{L}_{\text{ord}}$ as well as the scale invariant gradient loss $\mathcal{L}_{\text{grad}}$, thus yielding

$$\mathcal{L}_{\text{inpaint}} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{percep}} + 0.0001 \cdot \mathcal{L}_{\text{ord}} + \mathcal{L}_{\text{grad}} \tag{4.7}$$

as the combination of loss functions that we use to supervise the training of our color- and depth-inpainting network.

*Training.* We utilize Adam [66] with $\alpha = 0.0001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and train our inpainting network for $2 \cdot 10^6$ iterations. Given an input image, we require

ground truth novel views to supervise the training of the inpainting network. To this end, we extended our synthetic dataset and collected multiple views as described in Section 4.3.4 and shown in Figure 4.10.

#### 4.3.2.4 Summary

Our novel view synthesis approach is designed to address each of the identified requirements that are important when synthesizing the 3D Ken Burns effect: geometrically consistent inpainting, temporal consistency, and real-time synthesis. Please consider our supplementary video demo to further examine our synthesis results. This video demo also contains an example interaction with our user interface which exemplifies why real-time synthesis is a key feature when manually specifying the camera path.

### 4.3.3 User Interface

Given an input image, our system synthesizes the 3D Ken Burns effect from a virtual camera path parameterized by a start- and end-position. We obtain a sequence of frames by uniformly sampling novel view renderings across the linear path between the two positions. Here we describe how to derive camera positions from cropping windows placed on the input image, how to automatically select suitable cropping windows, and how to support the artist in using our system interactively.

#### 4.3.3.1 Camera Parametrization

When synthesizing the 2D Ken Burns effect, it is common practice to specify a source- and a target-crop within the input image. This approach provides an intuitive way to manually define the 2D scan and zoom. We adopt this paradigm of parameterizing the start- and end-view for our 3D Ken Burns effect. It is not trivial to match a cropping window in the 2D image space to a virtual camera position in 3D space. In

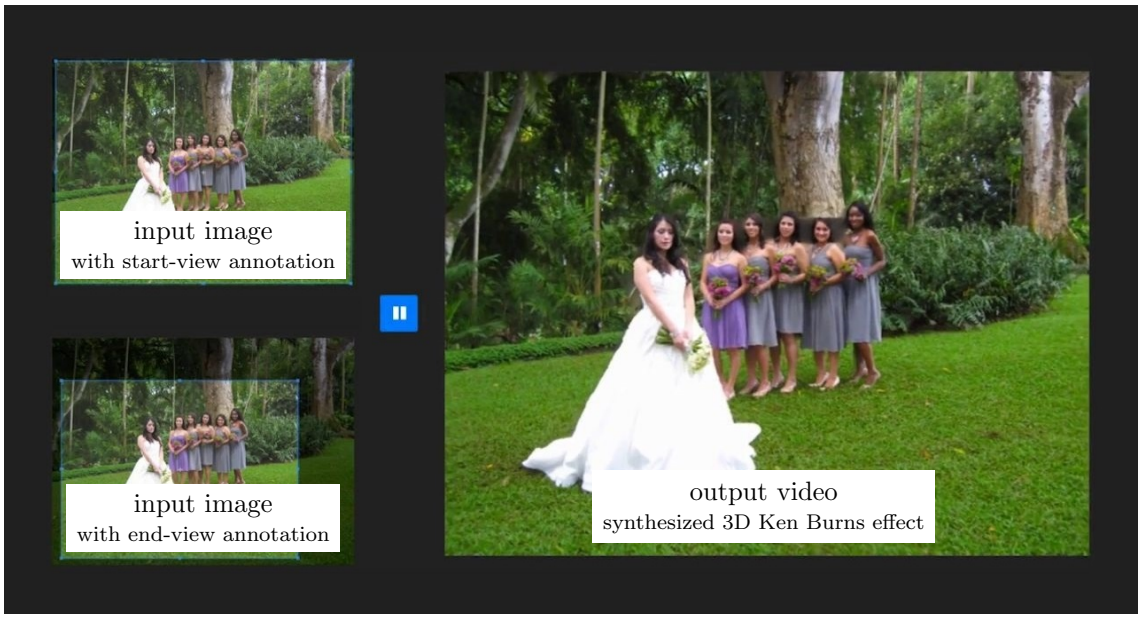



Figure 4.8: Example screenshot from the user interface. It allows users to manipulate the start- and end-view windows while perceiving the resulting effect in real time.

our method, we choose the XY-coordinate of the two virtual cameras such that the foreground object within the scene moves in accordance with the cropping windows. That is, if the source- and target-crop are 100 pixels apart then the foreground object should move by 100 pixels in the synthesized 3D Ken Burns result. Lastly, we use the size of the cropping windows in relation to the input image to determine the Z-coordinate of the corresponding virtual cameras.

#### 4.3.3.2 Automatic Mode

In the fully automatic mode, we let the algorithm automatically determine the start- and end-view such that the amount of disocclusion is minimized. Specifically, we treat the entire input image as the start-view and employ a uniform sampling grid to find the cropping window corresponding to the end-view that results in the minimum amount of disocclusion. In the resulting 3D Ken Burns effect, the virtual camera naturally approaches the the dominant salient foreground object and emphasizes it

Sample from NYU v2. Sample from MegaDepth.

Figure 4.9: Examples from the NYU v2 and the MegaDepth dataset, which provide sparse annotations that are subject to inaccurate depth boundaries.

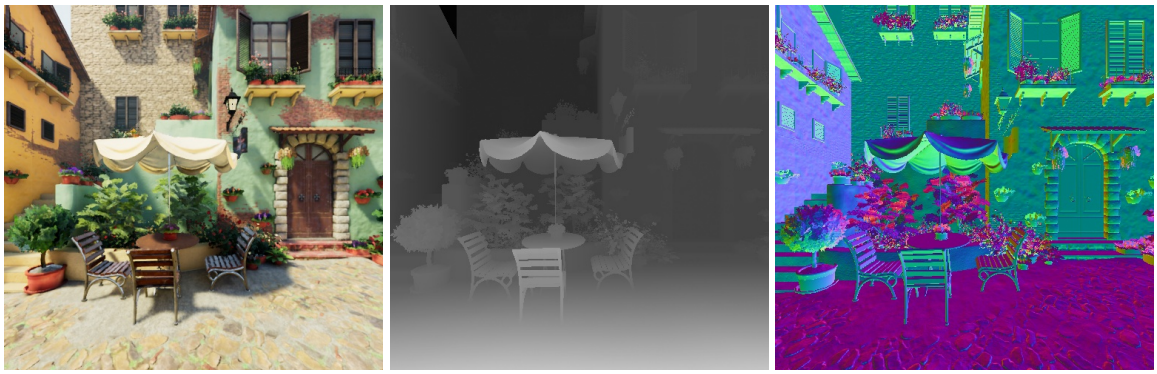
through motion parallax. An example result that we obtained using the automatic mode can be found at the top of Figure 4.1.

#### 4.3.3.3 Interactive Mode

Some users may desire a more fine-grained control over the synthesized 3D Ken Burns effect. To support this use case, we provide an interactive mode in which users determine the two cropping windows which represent the start- and end-view. Thanks to our efficient novel view rendering pipeline, our system can provide real-time feedback when manipulating the start- and end-view windows, which allows users to immediately perceive the effect of their actions. A screenshot is shown in Figure 4.8, please refer to our supplementary video for an example of our system in action.

### 4.3.4 Training Data

We evaluated several datasets that provide ground truth depth information to supervise the training of our depth estimation pipeline, including the MegaDepth [80]



Sample from the dataset. Corresponding depth. Corresponding normal.

Figure 4.10: Example sequence of four neighboring views from our training dataset. It is computer generated and consists of 134041 scene captures with 4 views each from 32 photo-realistic environments.

as well as the NYU v2 [133] dataset. However, as shown in Figure 4.9, these datasets only provide sparse annotations that are subject to inaccurate depth boundaries. We also examined the KITTI dataset [38], which also provides multi-view data and thus would be useful to supervise the training of our color- and depth-inpainting network. However, it is sparse and subject to inaccuracies as well and particularly limited in terms of scene types and content. As previously shown in Figure 4.5, accurate depth boundaries are crucial for novel view synthesis.

We thus created our own computer-generated dataset from 32 virtual environments, which enables us to extract accurate ground truth depth information. Those virtual environments were collected from the UE4 Marketplace [2] . We intentionally collected highly realistic environments covering a wide range of scene types such as indoor scenes, urban scenes, rural scenes, and nature scenes. More specifically, we use the Unreal Engine to create a virtual camera rig to capture 134041 scenes from 32 environments where each scene consists of 4 views. Each view contains color-, depth-, and normal-maps at a resolution of $512 \times 512$ pixels. Please see Figure 4.10 for an example from our dataset. While we did not use any normal-maps, we collected them regardless

[2] `http://www.unrealengine.com/marketplace/en-US/store`

such that other researchers can make better use of our dataset in the future. Note that, while training our depth estimation network, we randomly crop either the top and bottom or the left and right of each sample in order to facilitate invariance to the aspect ratio of the input image.

## 4.4 Experiments

### 4.4.1 Usability Study

We conduct an informal user study to evaluate the usability of our system in supporting the creation of the 3D Ken Burns effect. In particular, we are interested in investigating how easy it is for non-expert users to achieve desirable results for images with different content. To simulate a plausible scenario, we collected 3D Ken Burns videos created by artists. Specifically, we searched for phrases like "3D Ken Burns effect" or "Parallax Effect" on YouTube and selected 30 representative results from tutorial videos. We then only further considered those results that do not contain additional artistic effects such as compositing, artificial lighting, and particle effects. We categorize the remaining videos into four groups according to the scene types of the input image, namely "landscape", "portrait", "indoor", "man-made outdoor environment" and randomly selected three videos in each category. We thus conduct our informal user study on those 12 examples, for which we have the input image as well as reference 3D Ken Burns effect results.

We recruit 8 participants for our study. In each session, the participant is assigned one image along with the reference result created by an artist. The participant is asked to use our as well as two other systems to create a similar effect from the provided image. The order in which the systems are used is randomized. The usability and quality of each tool is rated by the participant at the end of the session.


Figure 4.11: Usability study results. Our study shows that our system enables users to achieve good results while requiring much less effort.

We compare our framework with existing solutions for creating the 3D Ken Burns effect. We consider two commercial systems. The first is the Photo Motion software package [3] which is implemented as a template for Adobe After Effects [4]. This package provides a commercial implementation for the framework introduced by Horry *et al.* [50] which is one of the most well-known frameworks for interactive camera fly-through synthesis. The second baseline system we consider is the mobile app Viewmee [5] that has been developed to allow non-expert users to easily create the 3D Ken Burns effect. This is one of very few systems that support simple interactions targeting casual users with limited image- or video-editing experience.

At the end of each session, the participant is asked to rate the three systems in terms of two criteria: system usability and result quality. For system usability, the participant rates each system with a score from one to five, with one indicating the lowest usability (i.e. the tool is too difficult to use to obtain acceptable results within the allocated 30 minutes) and five indicating the best usability (i.e. the tool is easy to

[3] `http://www.videohive.net/item/photo-motion-pro/13922688`

[4] `http://www.adobe.com/products/aftereffects.html`

[5] `http://itunes.apple.com/us/app/id1222280873`

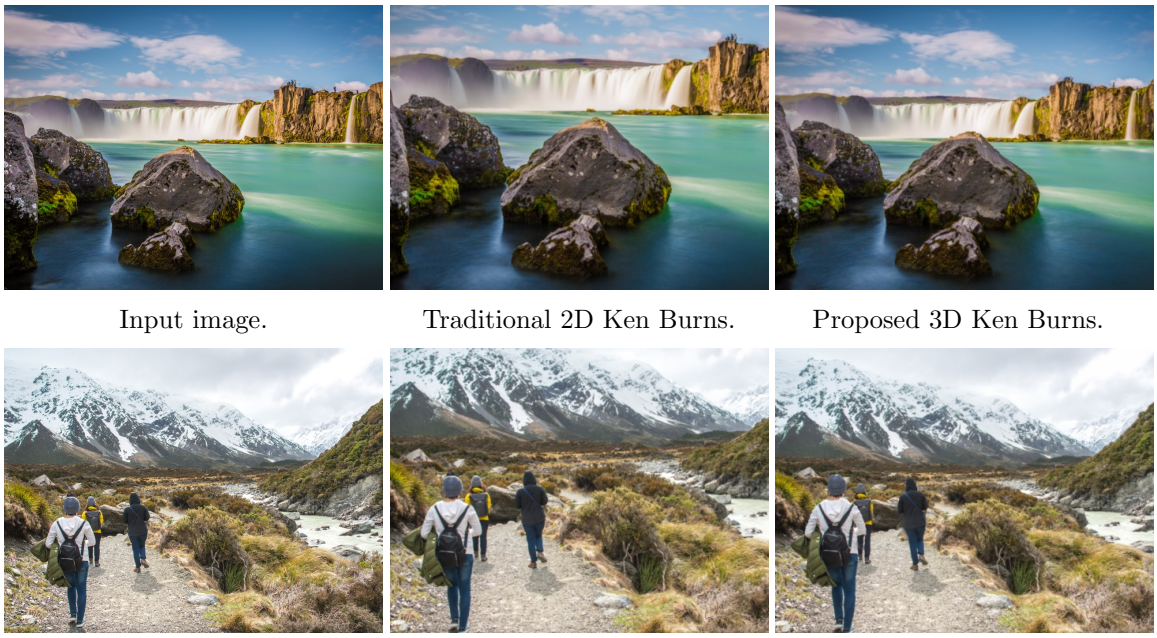
use to create good results). For the result quality, the participant is shown the three results that he or she created and asked to score each result from one to five, with one indicating the lowest quality and five indicating the highest quality.

We compare the user-provided usability scores as well as the per-system time for each of the 8 participants in Figure 4.11. The results show that using our system, the participants can obtain better results with much less effort compared to the other systems. Viewmee only seems to work for cases with a distinct foreground object in front of a distant background. Photo Motion Pro can model the scene depth for scenes with clear perspective but requires a lot of effort for manual segmentation and scene arrangement. It also is extremely difficult to use in scenes with many different depth layers. Please refer to our supplementary material for more visual examples shown in form of a video demo.

### 4.4.2 Automatic Mode Evaluation

As discussed in Section 4.3.3.2, our system provides an automatic mode that requires no user interaction. We investigate the effectiveness of our method in generating 3D Ken Burns effects from the input images automatically. In this experiment, we collect images from Flickr using different keywords, including "indoor", "landscape", "outdoor", and "portrait" to cover images of different scene types. We collect 12 images in total, with three images with different level of scene complexity in each category. We then use our automatic mode to generate one result for each image. For comparison, for each of our 3D Ken Burns effect result, we also generate a 2D Ken Burns effect result corresponding to the same camera path.

We evaluate the quality of our results with a subjective human evaluation procedure. We recruit 21 participants to subjectively compare the quality of our 3D Ken Burns synthesis results and the 2D counterparts. Each participant performs 12 comparison



Input image. Traditional 2D Ken Burns. Proposed 3D Ken Burns.

Figure 4.12: Example results comparing the 2D Ken Burns with our 3D Ken Burns. Please consider the supplementary video to examine the motion parallax.

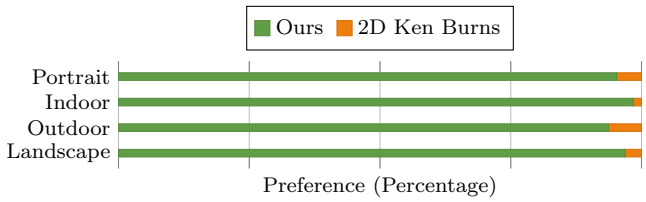



Figure 4.13: Results from a subjective user study comparing our 3D Ken Burns synthesis to a 2D baseline, indicating a strong preference for our system.

sessions corresponding to our 12 test images. Each session consists of a pair-wise comparison test presenting both the 3D and 2D Ken Burns synthesis results from an image in our test set. The participant is then asked to determine the result with better quality in terms of both 3D perception and overall visual quality.

Figure 4.13 shows average user preference percentage for our 3D Ken Burns effect results and those from the baseline 2D version for images in each category. The result indicates that our 3D Ken Burns synthesis results are preferred by the users in a majority of cases, which demonstrates the usefulness and effectiveness of our system. Please refer to our supplementary video for more visual examples of the comparison.

| Method | Training Data | Standard Metrics ($\sigma_i = 1.25^i$) rel ↓ | log10 ↓ | RMS ↓ | $\sigma_1$ ↑ | $\sigma_2$ ↑ | $\sigma_3$ ↑ |
|---|---|---|---|---|---|---|---|
| DIW | DIW | 0.25 | 0.10 | 0.76 | 0.62 | 0.88 | 0.96 |
| DIW | DIW + NYU | 0.19 | 0.08 | 0.60 | 0.73 | 0.93 | 0.98 |
| DeepLens | iPhone | 0.27 | 0.10 | 0.82 | 0.58 | 0.86 | 0.95 |
| MegaDepth | Mega | 0.24 | 0.09 | 0.72 | 0.63 | 0.88 | 0.96 |
| MegaDepth | Mega + DIW | 0.21 | 0.08 | 0.65 | 0.68 | 0.91 | 0.97 |
| Ours | Mega + NYU + Ours | 0.08 | 0.03 | 0.30 | 0.94 | 0.99 | 1.00 |
| Ours + Refinement | Mega + NYU + Ours | 0.08 | 0.03 | 0.30 | 0.94 | 0.99 | 1.00 |
| Ours w/ DIW arch | Mega + NYU + Ours | 0.18 | 0.07 | 0.56 | 0.76 | 0.94 | 0.98 |
| Ours w/o our data | Mega + NYU | 0.10 | 0.04 | 0.36 | 0.90 | 0.98 | 0.99 |

Table 4.1: Depth prediction quality on NYU v2. Our method compares favorably to state-of-the-art depth prediction methods in all depth quality metrics.

Figure 4.12 shows two examples comparing our generated 3D Ken Burns effect with the 2D version resulting from the same start- and end-view cropping windows. The 2D results show a typical zooming effect with no parallax. Our results, on the other hand, contain realistic motion parallax with strong depth perception, leading to a much more desirable effect.

### 4.4.3 Depth Prediction Quality

We now evaluate the effectiveness of our depth prediction module. We compare our depth prediction results with those from three state-of-the-art monocular depth prediction methods, including MegaDepth [80], DeepLens [153], and DIW [22]. For each method, we use the publicly available implementations provided by the authors. We evaluate the depth prediction quality using two public benchmarks on single-image depth estimation. We report the performance of MegaDepth, DeepLens, and DIW with their models trained on their proposed datasets. To address the scale-ambiguity of depth estimation, we scale and shift each depth prediction to minimize the absolute error between it and the ground truth.

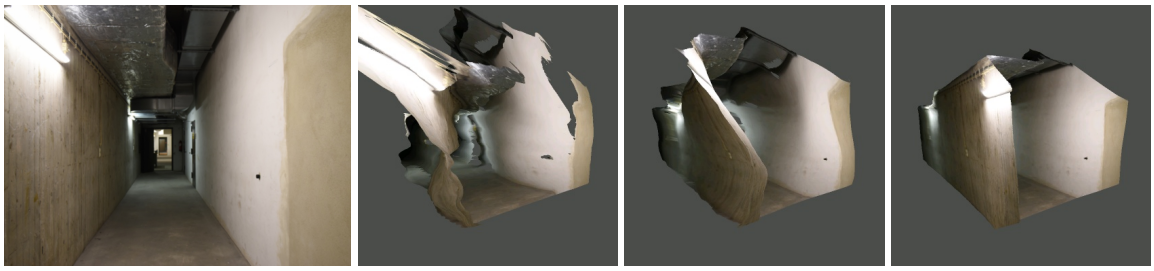
| Method | Training Data | Standard Metrics ($\sigma_i = 1.25^i$) rel ↓ | log10 ↓ | RMS ↓ | $\sigma_1$ ↑ | $\sigma_2$ ↑ | $\sigma_3$ ↑ | PE (cm / deg) $\varepsilon_{\text{PE}}^{\text{plan}}$ ↓ | $\varepsilon_{\text{PE}}^{\text{orie}}$ ↓ | DBE (px) $\varepsilon_{\text{DBE}}^{\text{acc}}$ ↓ | $\varepsilon_{\text{DBE}}^{\text{comp}}$ ↓ | DDE (% for $d = 3$ m) $\varepsilon_{\text{DDE}}^{0}$ ↑ | $\varepsilon_{\text{DDE}}^{+}$ ↓ | $\varepsilon_{\text{DDE}}^{-}$ ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIW | DIW | 0.25 | 0.10 | 1.00 | 0.61 | 0.86 | 0.95 | 4.55 | 41.46 | 10.00 | 10.00 | 81.17 | 8.76 | 10.08 |
| DIW | DIW + NYU | 0.19 | 0.08 | 0.80 | 0.72 | 0.91 | 0.97 | 6.16 | 30.30 | 7.93 | 9.41 | 85.68 | 7.25 | 7.07 |
| DeepLens | iPhone | 0.26 | 0.09 | 1.00 | 0.61 | 0.86 | 0.96 | 7.20 | 43.33 | 7.48 | 9.72 | 80.77 | 8.59 | 10.64 |
| MegaDepth | Mega | 0.23 | 0.09 | 0.83 | 0.67 | 0.89 | 0.96 | 7.62 | 35.51 | 5.40 | 8.61 | 83.11 | 9.05 | 7.84 |
| MegaDepth | Mega + DIW | 0.20 | 0.08 | 0.78 | 0.70 | 0.91 | 0.97 | 7.04 | 33.03 | 4.09 | 8.28 | 83.74 | 8.75 | 7.51 |
| Ours | Mega + NYU + Ours | 0.10 | 0.04 | 0.47 | 0.90 | 0.97 | 0.99 | 2.17 | 10.25 | 2.40 | 5.80 | 93.48 | 2.84 | 3.68 |
| Ours + Refinement | Mega + NYU + Ours | 0.10 | 0.04 | 0.47 | 0.90 | 0.97 | 0.99 | 2.19 | 10.24 | 2.02 | 5.44 | 93.49 | 2.83 | 3.68 |
| Ours w/ DIW arch | Mega + NYU + Ours | 0.15 | 0.06 | 0.62 | 0.80 | 0.95 | 0.99 | 6.31 | 19.49 | 3.12 | 8.04 | 89.10 | 5.68 | 5.22 |
| Ours w/o our data | Mega + NYU | 0.12 | 0.05 | 0.56 | 0.88 | 0.97 | 0.99 | 3.67 | 16.03 | 2.82 | 6.30 | 92.41 | 3.46 | 4.13 |

Table 4.2: Depth prediction quality on IBims-1. Our method compares favorably to state-of-the-art depth prediction methods in all depth quality metrics.

*NYU v2.* Silberman *et al.* [133] created one of the most well-known a benchmarks and datasets for single-image-depth estimation, consisting of 464 indoor scenes. Each scene contains aligned RGB and depth images, acquired from a Microsoft Kinect sensor. Following previous works on single-image depth estimation [22, 118, 181], we use the standard training-testing split and evaluate our method on the 654 image-depth pairs from the testing set.

*IBims-1.* Recently Koch *et al.* [68] introduced a new benchmark aiming for a more holistic evaluation of the depth prediction quality. This benchmark consists of 100 images with high-quality ground-truth depth maps. These images cover a wide variety of indoor scenes and the benchmark provides a comprehensive set of quality metrics to quantify different desired properties of a well-predicted depth map such as depth boundary quality, planarity, depth consistency, and absolute distance accuracy.

Table 4.1 and 4.2 (top) compare the depth prediction quality of different methods according to various quantitative metrics defined by each benchmark. Our method compares favorably to state-of-the-art depth prediction methods in all depth quality metrics. In addition, the result demonstrates that our depth prediction pipeline



Input image. DeepLens render. MegaDepth render. Rendered our depth.

Figure 4.14: Depth-based scene rendering. Compared to off-the-shelf methods, our depth prediction pipeline often better preserves the scene geometry.

improves significantly over off-the-shelf methods in terms of the Planarity Error (PE) and Depth Boundary Error (DBE) metrics on the iBims-1 benchmark. Those metrics are particularly designed to assess the quality in planarity and depth boundary preservation, respectively, which are particularly important for our synthesis task.

Table 4.1 and 4.2 (bottom) list two additional variations of our approach to better analyze the effect of our depth estimation network as well as our training dataset. Specifically, we supervised the network architecture from DIW [22] with all available training data to compare this architecture to ours. Furthermore, we supervised our depth estimation network only on the training data from MegaDepth and NYU v2 without incorporating our computer-generated dataset. Both variants lead to significantly worse depth quality metrics in the benchmark, which exemplifies the importance of all individual components of our proposed approach. Interestingly, both variants compare favorably to state-of-the-art depth prediction models.

Figure 4.14 compares the three-dimensional renderings with respect to different depth predictions. We can observe better preservation of the scene structure such as the planarity in our result compared to off-the-shelf depth prediction methods.

Input image. Artist 3D Ken Burns. Proposed 3D Ken Burns.

Figure 4.15: Example result comparing the 3D Ken Burns effect created by a professional artist with our automatic 3D Ken Burns synthesis. Please consider the supplementary video to examine the motion parallax.

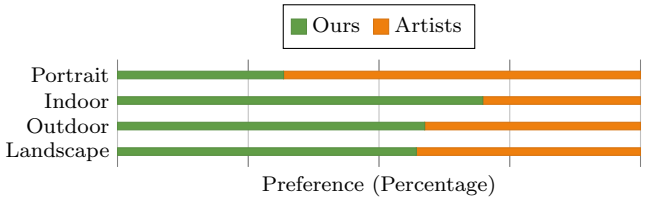



Figure 4.16: Results from a subjective user study comparing our 3D Ken Burns synthesis to results from artists, indicating no clear preference.

### 4.4.4 Discussion

Our previous experiment in Section 4.4.2 shows that users prefer our 3D Ken Burns effects in favor of the traditional 2D Ken Burns technique. It is also interesting to investigate how the effects created by our method compare to the ones made by skilled professional artists through laborious manual processing.

We conduct an additional subjective evaluation test. For each of the 12 artist-generated 3D Ken Burns results that we collected in Section 4.4.1, we use our system to create similar 3D Ken Burns effects using the corresponding input image. For each of the 12 test examples, we thus have a reference result generated by an artist and our result created by our proposed system. Please see Figure 4.15 for an example. We follow the same procedure as in Section 4.4.2. We ask the same set of 21 participants to perform 12 additional pair-wise comparison tests, comparing the results created by our system with the original artist-generated ones.

Figure 4.16 shows user preference percentage averaged over test cases in each category. Interestingly, our results are rated on-par with the ones from professional artists. Looking closely into each individual category, we observe that our results are slightly preferred compared to the artist's results in the indoor category. These scenes typically have a complicated depth distribution with many objects, which makes it extremely tedious to manually achieve the 3D Ken Burns effect. Our method can rely on a good depth prediction to handle those complicated scenes. The artist-created results, however, are more preferred in the portrait category. Looking into the results, we observe that portrait images often have simpler scene layouts which makes it easier to manually achieve good results. More importantly, we found that artists often intentionally exaggerate the parallax effect in portrait photos to make the effect much more dramatic to an extent that is not possible with physically-correct depth. This artistic emphasis is often preferred by viewers. Our method is limited by the parallax enabled by our depth prediction which is trained to match physically-correct depth and thus is not able to generate such dramatic effects.

We hope that our geometric- and semantic-aware depth prediction framework provides useful insights for future research in developing a more effective depth prediction tailored to view synthesis tasks. We would in this regard like to emphasize that the 3D Ken Burns effect is an artistic effect. In certain scenarios, view synthesis results generated from a physically correct scene prediction may not be optimal in delivering the desired artistic impression. Allowing such artistic manipulation in the 3D Ken Burns effect synthesis is an interesting direction to extend our work.

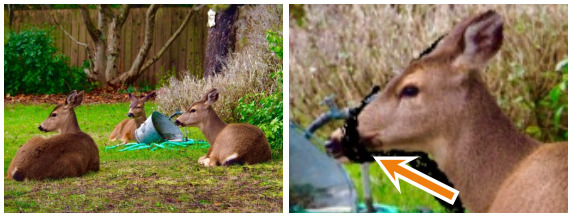
### 4.4.5 Limitations

While our method can generate a plausible 3D Ken Burns effect for images of different scene types, the results are not always perfect as shown in Figure 4.17. Single

a) Input image (left) and incorrectly estimated disparity at the reflection (right).



b) Input image (left) and estimated disparity with missing flagpole (right).



c) Input image (left) and magnified rendering with an inaccurate segmentation (right).

d) Rendering without (left) and with poorly inpainted point-cloud (right).

Figure 4.17: Examples of various commonly occurring issues with our proposed approach. Please see the limitations section for further details.

image depth estimation is highly challenging and our semantic-aware depth estimation network is not infallible. While our method can produce depth estimates subject to little or no distortion, we found that our results may still fail to predict accurate depth maps for challenging cases such as reflective surfaces (the reflection on the glossy poster in Fig. 4.17 (a)) or thin structures (the flagpole in Fig. 4.17 (b)). Object segmentation is challenging as well and the salient depth adjustment may fail due to erroneous masks. While our depth upsamling module can perform boundary-aware refinement to account for some mask inaccuracies, our result is affected when the error in the segmentation mask is significantly large. In Fig. 4.17 (c), the nose of the deer is cut off due to Mask R-CNN providing an inaccurate segmentation. Finally, we note that while our joint color- and depth-inpainting is an intuitive approach to extend the estimated scene geometry, it has only been supervised on our synthetic data and thus may sometimes generate artifacts when the input differs too much from the training data. In Fig. 4.17 (d), the inpainting result lacks texture and is darker than expected.

Training the color- and depth-inpainting model with real images and leveraging an adversarial supervision regime and a more sophisticated architecture, like one that uses partial convolutions, is an interesting direction to explore in future work.

## 4.5 Conclusion

In this paper, we developed a complete framework to produce the 3D Ken Burns effect from a single input image. Our method consists of a depth prediction model which predicts scene depth from the input image and a context-aware depth-based view synthesis model to generate the video results. To this end, we presented a semantically-guided training strategy along with high-quality synthetic data to train our depth prediction network. We couple its prediction with a semantics-based depth adjustment and a boundary-focused depth refinement process to enable an effective depth prediction for view synthesis. We subsequently proposed a depth-based synthesis model that jointly predicts the image and the depth map at the target view using a context-aware view synthesis framework. Using our synthesis model, the extreme views of the camera path are synthesized from the input image and the predicted depth map, which can be used to efficiently synthesize all intermediate views of the target video, resulting in the final 3D Ken Burns effect. Experiments with a wide variety of image content show that our method enables realistic synthesis results. Our study shows that our system enables users to achieve better results while requiring little effort compared to existing solutions for the 3D Ken Burns effect creation.

# 5 Novel View Synthesis in Time and Space

This chapter will be submitted to a conference such as CVPR or similar. All uses of "we" or "our" refer to the authors of this paper (Simon Niklaus, Long Mai, Oliver Wang, Dingzeyu Li, and Feng Liu). Simon's contributions are: the overall framework for the video action shot synthesis including the human-aware odometry estimation and the novel pipeline for reconstructing the protagonist, the human depth dataset, the architecture details such as using the GridNet architecture, all implementation aspects, the design and execution of the evaluation, and the paper writing.

## 5.1 Abstract

Action shots summarize the motion of an object in a video as a still image. In comparison, a video action shot not only depicts the motion trajectory as a still image, it augments the input video with past and future appearances of the main subject. Creating such effects manually is time-consuming and demands sophisticated editing stills. Existing automatic solutions, however, are limited to video footage from static cameras or requires videos from cameras with depth sensors. In this paper, we propose an automated framework for synthesizing video action shots from everyday video footage of human subjects. To achieve this, we perform human-aware odometry prediction, estimate the shape and location of the protagonist within the world geometry, and rerender the video while depicting past and future occurrences of

the human subject. Experiments on a wide variate of video footage show that our method enables synthesizing realistic video action shots.

## 5.2 Introduction

When displaying a collection of videos, it is common to use thumbnails to represent the videos. Automatically generating a representative thumbnail is challenging though and the state of the art for automatic thumbnail generation is based on selecting a representative frame from the video. Generating a meaningful thumbnail that summarizes the content of the video is a task that is still an open research question. Existing research typically focuses on domain specific problems, such as the summarization of surveillance footage [92, 114, 115, 123] or the visualization of human motion [6, 18, 35, 171]. This makes it possible to simplify the problem by, for example, assuming that the camera is static or by approximating the subject as a textureless parametric model [13]. However, these simplifications either limit the content for which this effect can occur, or they yield non-photorealistic renderings.

Summarizing a video of an object in motion is commonly referred to as an action shot. Given a video, for example of a running person, the summary depicts the motion trajectory through multiple occurrences of the runner as a still image. In comparison, a video action shot as shown in Figure 5.1 not only depicts the motion trajectory as a still image, it augments the input video with past and future appearances of the main subject. This augmented video makes it possible to carefully observe the motion of the subject in question. Conceptually, each frame in a video action shot can be considered as a traditional action shot. Creating such an effect is difficult though and it is currently predominantly achieved through laborious manual editing using specialized software [1] . And while Klose *et al.* [67] have shown how to generate

[1] `https://www.youtube.com/watch?v=lG4IBHPHZP0`

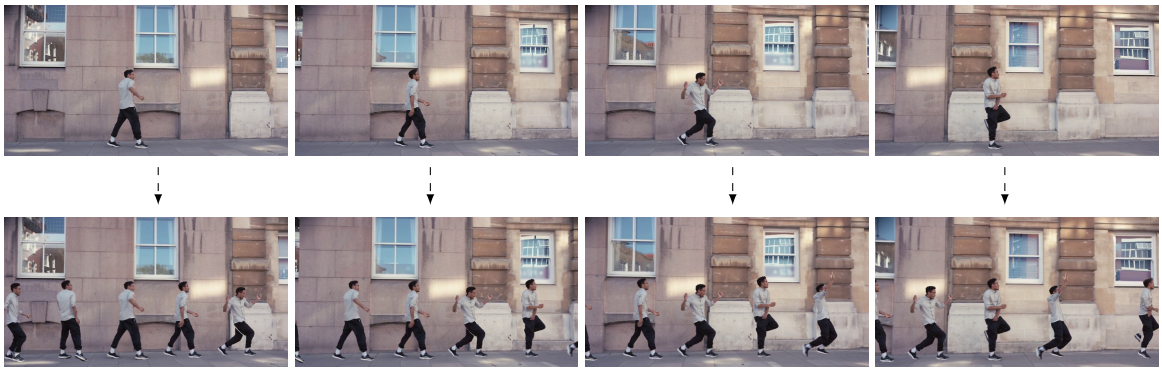


Figure 5.1: An example of a video action shot (bottom) from a sequence of input frames with a moving human subject (top). Each frame in the output video represents an action shot and extends the respective input frame with past and future appearances of the main subject.

this effect automatically through computer vision, their approach requires a camera that also captures scene depth. In contrast, this paper focuses on photorealistic video action shots on everyday video footage.

With unconstrained video footage, transferring the appearance of an object in motion from a source frame to a target frame requires knowing how the camera moves and which shape and location the object has within the world geometry. The former is challenging due to the moving object since many approaches for visual odometry and structure from motion assume a static scene [100, 101, 130, 131]. The latter is challenging since the transferred object in the target frame may be depicted from an angle that differs from the source frame while simultaneously being subject to non-rigid deformations [2, 14, 116, 160]. To make this problem tractable, we focus on humans as the main subject. This allows us to detect and exclude the non-static subject from the odometry estimation. Furthermore, it is possible to reconstruct a reasonably accurate shape of a human from a single image with current state-of-the-art technologies, which avoids having to deal with non-rigid deformations.

In short, we propose a framework for synthesizing video action shots from unconstrained video footage of human subjects. To achieve this, we perform human-aware

Input frame. Reconstruction from COLMAP, cameras shown in red.

Figure 5.2: Example structure-from-motion reconstruction, visualized as a point cloud with cameras shown in red. Notice that there are no points corresponding to the human from the input frame since the related feature points were removed during pre-processing. Even though the point cloud is noisy due to reflections in the glass windows, the camera parameters have been estimated sufficiently well.

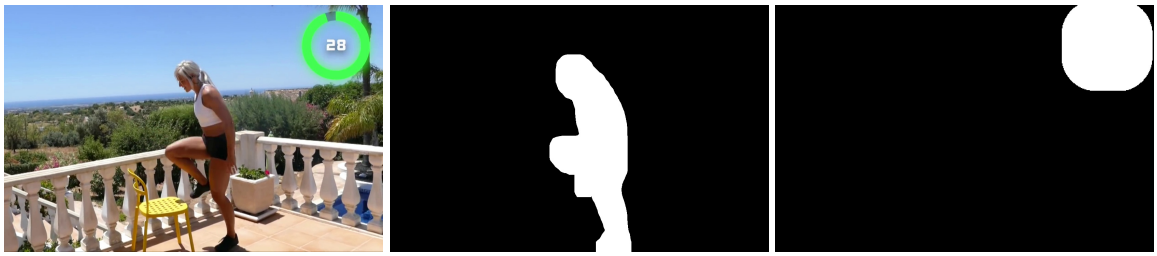
odometry prediction, estimate the three-dimensional position of the protagonist in each frame through depth-based localization, predict a per-frame high-resolution reconstruction of the human, and rerender the video while depicting past and future occurrences of the human subject.

## 5.3 Video Action Shot Synthesis

Our framework consists of four steps. First, estimating the camera extrinsics and intrinsics through human-aware odometry prediction. Second, localizing the protagonist within the world geometry. Third, predicting a per-frame high-resolution shape of the human. Fourth, rerendering the video while depicting past and future occurrences of the human subject. We subsequently describe these steps.

### 5.3.1 Human-aware Odometry Estimation

Since we do not constrain the input video and allow for camera motion, the applicability of traditional image-based rendering techniques is limited [171]. To be able to fully account for the camera motion, this includes not only camera extrinsics



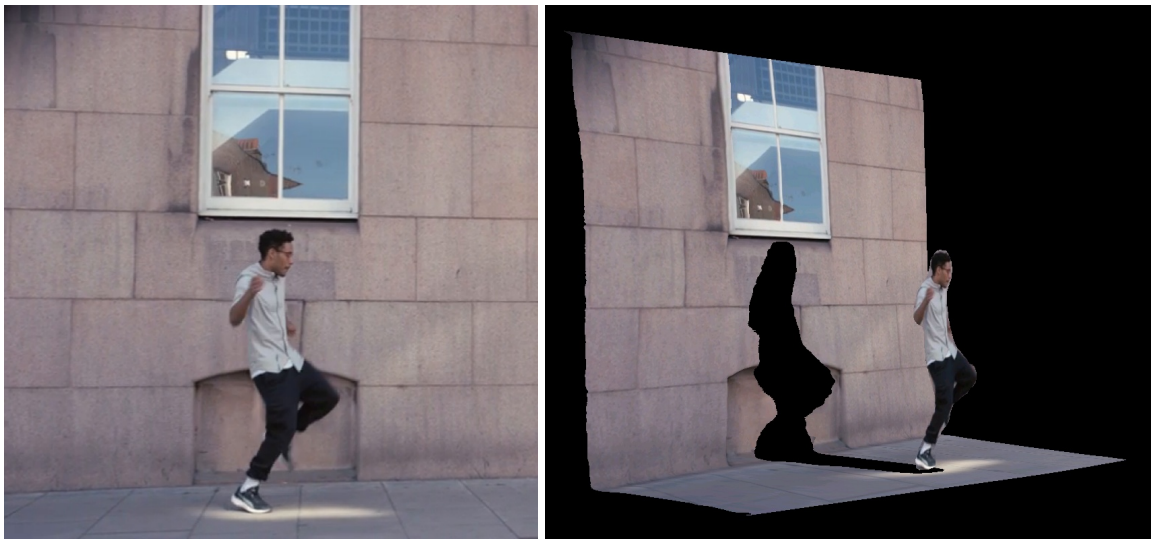
Input frame. Human estimate. Logo estimate.

Figure 5.3: Example masks for pre-processing an input frame. Odometry estimation typically assumes a static scene, we thus determine masks which violate this assumption and withhold these regions from the reconstruction process.

but also its intrinsics. We subsequently assume a pinhole camera model with radial distortions. Before recovering the camera parameters though, we can pre-process the video and exclude the human subject as well as any logos from the input.

**Mask estimation.** Odometry estimation typically assumes a static scene. This assumption does not hold true for the videos that we are targeting due to the apparent motion of the human protagonist. However, we can easily apply instance segmentation, for example using Mask R-CNN [44], to identify a human mask and withhold it from the odometry estimation. In doing so, we apply morphological dilation to increase the size of the human mask to account for errors introduced by the instance segmentation. Furthermore, we extend the masking and additionally include regions that contain image overlays such as logos or information displays that likewise violate the assumption of a static scene. We utilize a temporal gradient estimate to identify such regions [24]. Please see Figure 5.3 for an example of such masks.

**Odometry estimation.** There are a multitude of techniques to estimate the trajectory of a camera, including visual odometry, simultaneous location and mapping, and structure-from-motion. We evaluated several approaches for our given purpose, including COLMAP [130, 131], ORB-SLAM2 [100, 101], and DSO [30]. Note that we modified each of them to exclude the previously discussed masks. After evaluating



Input frame. Estimated human depth.

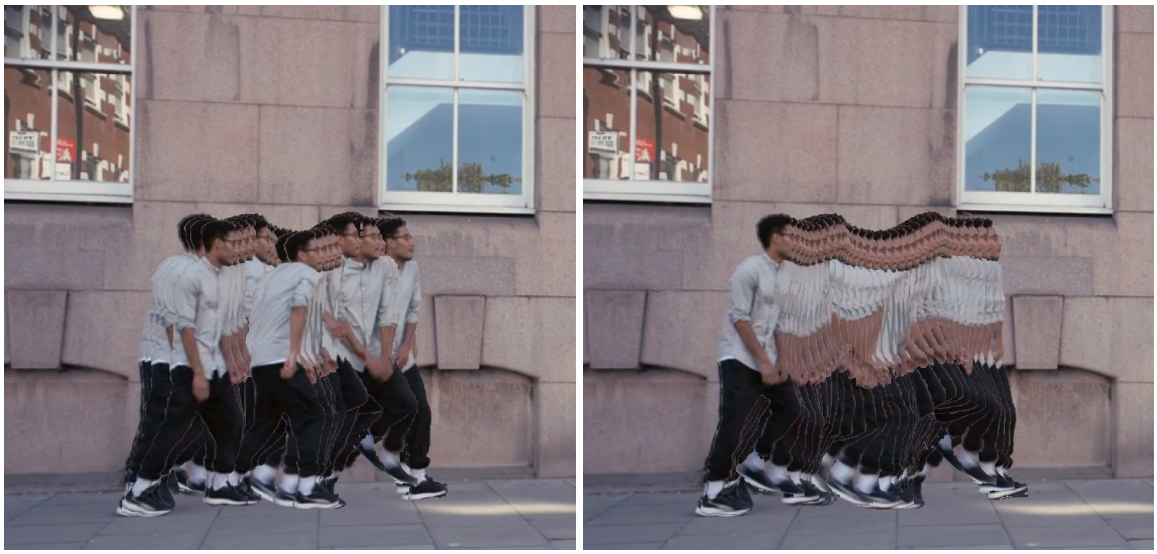
Figure 5.4: We employ single image depth estimation to obtain a dense geometry reconstruction for each frame in the input video and align this estimate to the sparse world reconstruction from COLMAP. We employ the approach from Niklaus *et al.* [109] to do so, which, for each frame in the input video, yields the three-dimensional location of the protagonist as a billboard within COLMAP's coordinate system.

these approaches on our test footage, we opted to use COLMAP and hence structure-from-motion for the given task. While structure-from-motion is computationally expensive, it consistently achieved accurate reconstructions in our tests.

### 5.3.2 Depth-based Human Localization

With video footage that is subject to camera motion, copying past and future occurrences of the human subject to a given output frame necessitates localizing the protagonist within the world geometry throughout time. Even though we use structure-from-motion in the previous step, which yields a point cloud reconstruction of the world geometry, we cannot directly use this sparse representation to locate the human subject since it has been excluded from the reconstruction.

**Depth-based registration.** The point cloud reconstruction obtained via structure-from-motion represents a sparse representation of the world geometry. We hence



Without temporal refinement. With temporal refinement.

Figure 5.5: The depth-based registration is subject to temporally inconsistent predictions which yields undesired results due to incorrect depth orderings (left). We employ a temporal filter to account for these inaccuracies (right).

employ single image depth estimation and obtain an additional dense reconstruction of the geometry depicted by each frame in the input video. We employ the approach from Niklaus *et al.* [109] to do so, which conveniently approximates humans as three-dimensional billboards. This dense reconstruction also includes the depth and hence the three-dimensional location of the human subject. However, these single image depth estimates are not aligned with COLMAP's coordinate system. We thus align each individual dense reconstruction to the global point cloud reconstruction by solving a least squares problem that recovers the otherwise unknown scale and bias. As demonstrated in Figure 5.4, this allows us to determine the three-dimensional location of the protagonist within COLMAP's coordinate system.

**Temporal refinement.** The per-frame dense reconstruction is subject to noise, for example due to erroneous single image depth estimates. As shown in Figure 5.5, this leads to slight inter-frame inconsistencies with respect to the estimated human location which yields undesired rendering results due to incorrect depth orderings. We

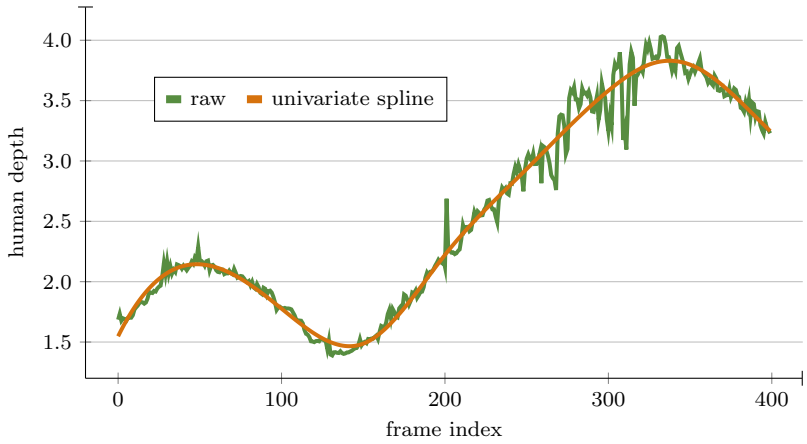



Figure 5.6: We smooth the per-frame human depth estimate using a univariate spline to account for temporally inconsistent single image depth estimates.

account for these by temporally smoothing the per-frame human depth estimate using a univariate spline as shown in Figure 5.6, which ensures consistent depth orderings.

### 5.3.3 Per-frame Human Reconstruction

At this point, we are able to estimate the camera trajectory as well as the location of the human subject throughout time within COLMAP's reconstruction of the world geometry. As shown in Figure 5.4, approximating the protagonist as a billboard does not yield satisfying results when rendering the subject at an angle. We thus augment the flat billboard with a three-dimensional shape as subsequently discussed.

**Human modeling.** Human reconstruction is a popular research topic within computer vision and there is a multitude of approaches for modelling humans. Some common examples are shown in Figure 5.7, which includes keypoints, parametric models, surface-based representations, and depth. While traditional skeleton keypoints are comparatively easy to estimate, they lack the structure that the given task necessitates. In comparison, SMPL [13] and DensePose [42] provide a more comprehensive approximation of the human shape. However, SMPL is lacking texture and DensePose is lacking shape. We thus opt to directly predict the depth of the human [142, 150]

OpenPose HMR / SMPL DensePose Depth

Figure 5.7: Example visualization of common approaches for modelling humans, which includes keypoints, parametric models, surface-based representations, and depth.

using a custom approach that, given an input image, estimates the human depth and a corresponding human mask at a high-resolution.

**Human shape estimation.** Our approach for estimating the shape of the human subject is shown in Figure 5.8, it illustrates our proposed pipeline for predicting the human depth and a human mask at a high-resolution from a given input image. Our shape estimation consists of three steps. First, estimating a rough human mask and the human keypoints through R-CNN [44]. Second, cropping the human according to the rough human mask and estimating the human depth and a human mask at a low-resolution using a neural network. This neural network takes not only the cropped input image as input, but it is also guided by the rough human mask as well as the human keypoints. Third, refining the low-resolution estimates guided by the high-resolution input image. This pipeline is loosely modeled after the depth estimation from Niklaus *et al.* [109], who identified and addressed several key issues when using depth for image synthesis. We accordingly also adopted their coarse estimation approach based on a GridNet [34] that receives VGG-19 [134] features, together with their proposed refinement network architecture and training regime.

**Loss functions.** While existing work on single image depth estimation has emphasized the importance of a scale-invariant loss [29, 149], we have found a simple $\ell_1$-based loss to be reasonably successful for our domain-specific human depth estimation task.

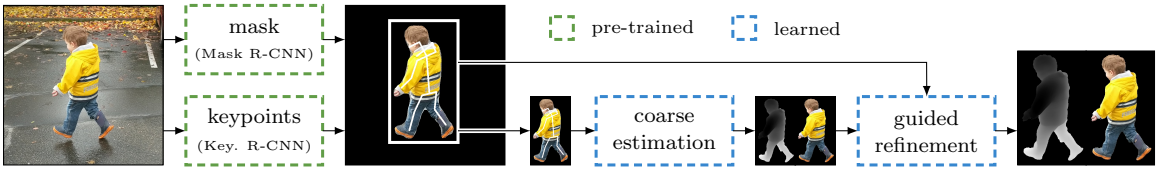



Figure 5.8: Overview of our human shape estimation approach that, given an input image, predicts the human depth and a human mask at a high-resolution. It consists of three steps. First, estimating a rough human mask and the human keypoints. Second, estimating the human depth and a human mask from a crop of the human at a low-resolution. This coarse estimation is additionally guided by the rough human mask and the human keypoints. Third, refining the low-resolution estimates guided by the high-resolution input.

Specifically, we minimize the difference between the ground truth human depth $Z_{gt}$ and the estimated human depth $Z$ using $\mathcal{L}_{depth}$ as follows.

$$\mathcal{L}_{depth} = \|Z - Z_{gt}\|_1 \tag{5.1}$$

Note that we only apply this loss term at pixel locations with known ground truth human depth. To encourage a spatially smooth human depth prediction, we additionally enforce $\mathcal{L}_{tv}$ to minimize the total depth variation as follows.

$$\mathcal{L}_{tv} = \|\nabla_x Z\| + \|\nabla_y Z\| \tag{5.2}$$

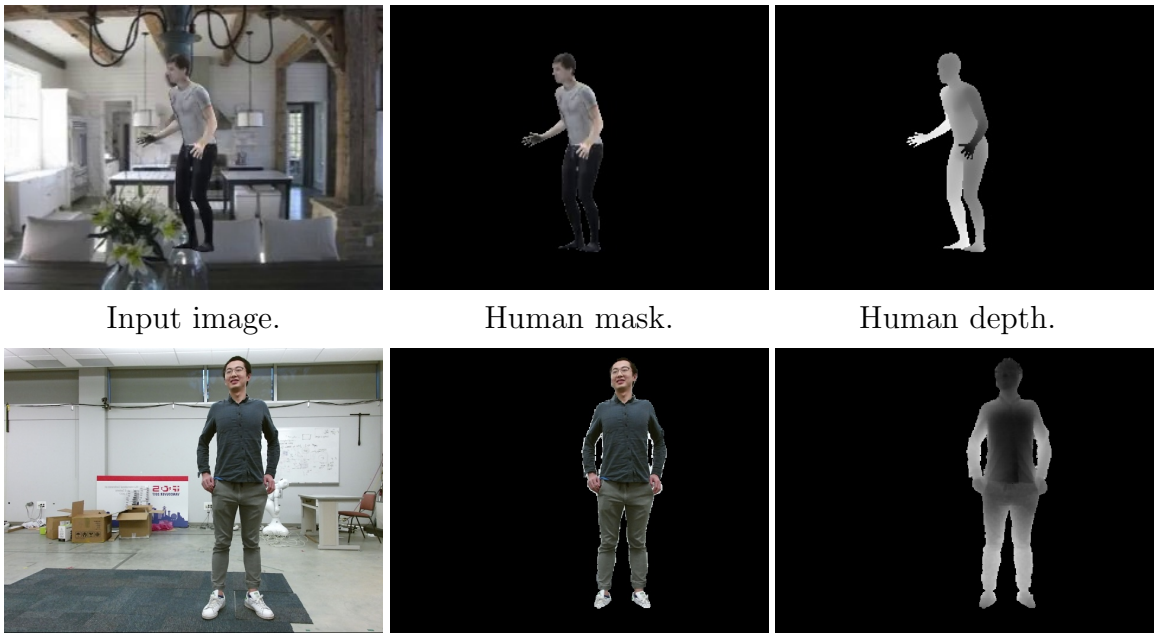
Estimating the human mask is a binary classification problem. As such, we minimize the difference between the ground truth human mask $H_{gt}$ and the estimated human mask $H$ using a loss $\mathcal{L}_{mask}$ based on binary cross entropy as follows.

$$\mathcal{L}_{mask} = -\Big(H_{gt} \cdot \log(H) + (1 - H_{gt}) \cdot \log(1 - H)\Big) \tag{5.3}$$

We train each of our two networks, the coarse estimation network and the guided refinement network, with the following combination of these three loss terms.

$$\mathcal{L}_{total} = \mathcal{L}_{depth} + 0.1 \cdot \mathcal{L}_{tv} + \mathcal{L}_{mask} \tag{5.4}$$

**Training data.** Unfortunately, training data with accurate ground truth human depth and human mask annotations is difficult to acquire. Varol *et al.* [150] used



Input image. Human mask. Human depth.

Figure 5.9: Examples from two existing dataset that contain ground truth human depth annotations. Varol *et al.* [150] (top) used computer graphics and rendered texturized SMPL [13] models on top of static images. Tang *et al.* [142] (bottom) used a Microsoft Kinect camera to collect RGB-D videos.

computer graphics and rendered texturized SMPL [13] models on top of static image backgrounds to acquire such data. In comparison, Tang *et al.* [142] used a Microsoft Kinect camera to collect RGB-D videos. As shown in Figure 5.9, however, the dataset from Varol *et al.* [150] looks unrealistic and the dataset from Tang *et al.* [142] is subject to significant inaccuracies around the boundary of the human. We have thus acquired a new computer-generated training dataset by capturing scenes with human subjects from within GTA 5 as shown in Figure 5.10, a video game that simulates a large virtual world. Specifically, we modified the `ClearDepthStencilView` function to acquire the depth buffer as well as the stencil buffer which includes information about the human mask. We then walked through the world of GTA 5 and captured scenes with human subjects. This task requires a significant amount of manual labor which is why we automatically capture four different views at two different illumination settings

Input image. Human Mask. Human depth.

Figure 5.10: Examples form our training dataset which consists of human subjects captured from within GTA 5, a video game that simulates a large virtual world.

per scene. We eventually collected 1000 scenes and thus 8000 samples in total. Since this is still a small size for a training dataset, we additionally incorporated the dataset from Varol *et al.* [150] which we rerendered at a higher resolution. The samples in our dataset as well as our rerendering of the data from Varol *et al.* [150] have a resolution of $1024 \times 1024$ pixels with accurate human depth and human mask annotations.

**Training scheme.** We train our coarse estimation and our guided refinement network in stages. That is, we first train the coarse estimation network until convergence before training the guided refinement network until convergence. We utilize Adam [66] with $\alpha = 0.00005$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ for this purpose. We used 4 samples per batch and it took $2 \cdot 10^6$ iterations for the coarse estimation network and $1 \cdot 10^6$ iterations for the refinement network to converge.

Input image. Coarse shape, points. Refined shape, mesh.

Figure 5.11: Depth-based rerendering of a human rotated by 40 degrees. The coarse shape estimate is subject to boundary artifacts as outlined by Niklaus *et al.* [109] whereas the rendering from the refined shape estimate yields realistic results. Please consider the supplementary video to see this example in motion.

### 5.3.4 Depth-based Human Rerendering

Given an image and the estimated human depth of the depicted protagonist, it is straightforward to map the pixels of the human to points within COLMAP's coordinate system in accordance with the estimated camera parameters. However, rendering depth-based point clouds requires special care [109]. We thus instead convert the estimated human shape to a vertex mesh within COLMAP's coordinate system and use an off-screen OpenGL context for rendering. We optionally also perform alpha blending at object boundaries to avoid aliasing artifacts. An example rerendering is shown in Figure 5.11, please consider the supplementary material to see this example in motion. Our depth-based vertex mesh rendering yields results that are free from artifacts that are common when performing depth-based rerendering, such as objects that are being torn apart at boundary regions [109]. However, the resulting rerendering

HMR / SMPL DensePose Depth

Figure 5.12: Example combination of SMPL and DensePose estimates, using the estimated SMPL model to augment the DensePose labels with depth information.

is subject to missing regions due to self-occlusion. This issue could be addressed by, for example, neural rendering techniques [3, 16, 94]. We consider such research efforts orthogonal to the work presented in this paper.

## 5.4 Experiments

We are, to the best of our knowledge, the first to propose a framework for synthesizing video action shots. As such, it is difficult to compare the results from our approach to those from other methods. We will thus resort to showing qualitative results of our video action shots in the supplementary video. However, there are various related approaches for estimating the human depth or a human mask from an input image. This section will thus focus on evaluating the human depth and the human mask prediction quality of our proposed human shape estimation.

**Methods.** As shown in Figure 5.4, it is possible to approximate a human as a billboard. We include this simple approach in our evaluation as a simple baseline for human depth estimation. Our human shape estimation is initialized with a rough human mask predicted by Mask R-CNN [44]. We include this method as a simple baseline for human mask estimation. Furthermore, we extend it by refining the

Figure 5.13: Representative examples from our evaluation dataset. It consists of high-quality human models which we rendered with HDRI backgrounds.

boundary through matting. Specifically, we use an off-the-shelf matting approach [163] to refine the boundary of the rough human mask. As shown in Figure 5.12, parametric models like SMPL [62] as estimated via HMR [62] may not always yield ideal results but we nevertheless include this approach for completeness. Furthermore, Figure 5.12 also shows that one can use an SMPL prediction to augment a DensePose [42] estimate with depth information which yields slightly better results. Lastly, we include the recent human depth estimation approach from Tang *et al.* [142] and use Mask R-CNN to initialize it as recommended by the authors.

**Metrics.** We include the mean absolute error (MAE) as well as the root mean square error (RMSE) for evaluating the human depth estimates. Since the human depth estimates may not have depth annotations for all pixels that belong to the human

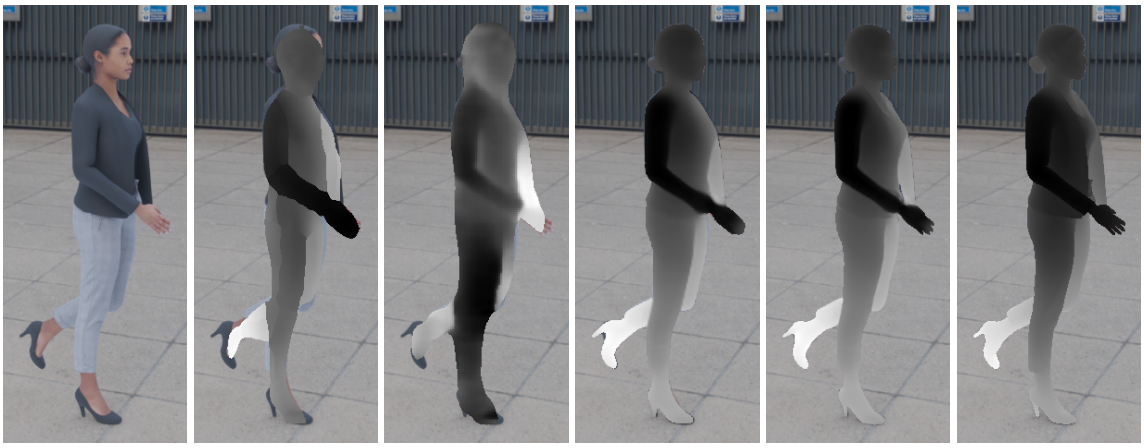
| Method | human depth | | | human mask | | |
|---|---|---|---|---|---|---|
| | MAE ↓ | RMSE ↓ | valid ↑ | preci. ↑ | recall ↑ | F1 ↑ |
| Billboard Baseline | 0.045 | 0.999 | *N/A* | *N/A* | *N/A* | 0*N/A* |
| Mask R-CNN [44] | *N/A* | *N/A* | *N/A* | 0.905 | 0.974 | 0.938 |
| Mask R-CNN [44] + Matting [163] | *N/A* | *N/A* | *N/A* | 0.958 | 0.976 | 0.966 |
| HMR [62] | 0.022 | 0.523 | 77.0% | 0.906 | 0.770 | 0.831 |
| HMR [62] + DensePose [42] | 0.023 | 0.516 | 87.4% | 0.964 | 0.874 | 0.916 |
| Tang *et al.* [142] + Mask R-CNN [44] | 0.034 | 0.783 | 97.4% | 0.905 | 0.974 | 0.938 |
| Ours w/o Refinement | 0.018 | 0.406 | 98.1% | 0.980 | 0.981 | 0.980 |
| Ours w/ Refinement | 0.018 | 0.410 | 98.8% | 0.989 | 0.988 | 0.989 |

Table 5.1: Human depth and human mask prediction quality on our dedicated evaluation dataset. Our proposed approach compares favorably in this benchmark.

subject, for example due to an inaccurate human mask, we additionally state how many valid predictions the human depth estimate contained. To address the scale-ambiguity of depth estimation, we additionally scale and shift each human depth estimate to minimize the absolute error between it and the ground truth. As for evaluating the human mask, we use the common metrics of precision, recall, and F1 score.

### 5.4.1 Evaluation Dataset

While we could have collected additional testing samples from GTA 5 as outlined in Section 5.3.3, we wanted to create an unbiased dataset that provides the basis for a fair comparison. We thus created another computer-generated dataset, this time using high-quality human models with backgrounds consisting of HDRIs to mimic different lighting conditions. We used a shadow catcher to obtain accurate shading and utilized GPU-accelerated path tracing with subsequent denoising to render the scenes with a resolution of $1024 \times 1024$ pixels. Our evaluation dataset consists of 13 human models with different poses and 15 HDRI backgrounds, resulting in a total of 195 samples. Please see Figure 5.13 for representative examples from our evaluation dataset.



Input [62] + [42] Tang *et al.* Ours w/o Refi. Ours w/ Refi. Truth

Figure 5.14: Qualitative comparison of human depth predictions. Please consider the supplementary video to see an animated comparison of the resulting renderings.

### 5.4.2 Quantitative Evaluation

We compare our human shape estimation results with several state-of-the-art methods. Please see Table 5.1 for the results on our dedicated evaluation dataset. Our proposed shape estimation compares favorably to state-of-the-art methods with respect to the quality of the estimated human depth as well as the quality of the estimated human mask, regardless of whether we just consider our coarse prediction or the refined one. When comparing our refined prediction with our coarse one, we notice that the MAE and the RMSE remain largely the same whereas the percentage of valid predictions, the precision, the recall, and the F-1 score increase slightly. This matches our expectations, indicating that the refinement improves boundary predictions.

### 5.4.3 Qualitative Evaluation

We show a visual comparison of the human depth predictions in Figure 5.14, please consider the supplementary video to see an animated comparison of the resulting renderings. The visual comparison shows that HMR [62] together with DensePose [42] yields believable depth orderings but is incomplete at boundary regions and is subject

Input Mask R-CNN [44] + [163] DensePose Ours w/o Refi. Ours w/ Refi.

Figure 5.15: Qualitative comparison of human mask predictions. Blue indicates a predicted mask that is too big, red indicates a predicted mask that is too small.

to depth discontinuities. The depth predictions from Tang *et al.* [142] are smoother and better capture the silhouette of the human but they deviate from the ground truth human depth. In comparison, our estimated human depth closely resembles the ground truth and better captures the human silhouette. These findings support the results from the quantitative evaluation.

We show a visual comparison of the human mask predictions in Figure 5.15, where blue indicates a predicted mask that is too big and red indicates a predicted mask that is too small. The visual comparison shows that Mask R-CNN[44] is subject to significant inaccuracies due to only estimating instance segments at $28 \times 28$ pixels. While matting [163] is able to improve this prediction, the estimated human mask still deviates significantly from the ground truth. DensePose [42] is able to better capture the area of the head and of the legs but misses a piece of the jacket due to being agnostic to clothing. In comparison, our estimated human mask closely resembles the ground truth. Please consider zooming into this image to better compare the results from our coarse shape estimation with those from our fine shape estimation.

Figure 5.16: Example action shots created using our proposed framework. Please consider our supplementary video to see video action shots in motion.

### 5.4.4 Results

Videos are at the heart of this paper and we provide video action shots created with our proposed framework in the supplementary video. We additionally provide still results in Figure 5.16, which show traditional still action shots.

### 5.4.5 Limitations

Synthesizing video action shots on footage with a moving camera is inherently challenging due to having to estimating the camera motion as well as the per-frame location and shape of the moving protagonist. Furthermore, this cannot be done separately since the estimates need to be in the same coordinate system. While our proposed framework accounts for these factors, it involves several steps and errors can easily propagate. Furthermore, we expect a specific human shape for the main subject. This assumption can easily be violated as shown in Figure 5.17 where our predicted human mask is erroneous due to an additional jacket. Other examples where this assumption may break are backpacks and hats. Our framework also expects that the main subject is not occluded. This assumption does not hold true if the protagonist is, for example, briefly occluded by a lamp pole. Lastly, our human shape prediction does not extend the predicted shape beyond the visible area. As such, renderings from

Input Mask R-CNN [44] + [163] DensePose Ours w/o Refi. Ours w/ Refi.

Figure 5.17: Example failure case for the human shape estimation. Our predicted human mask is erroneous due to the additional jacket.

significantly different perspectives may, as shown in Figure 5.17, be incomplete due to self-occlusion. Our rendering approach also does not account for the environment lighting and we thus do not not correctly synthesize shadows.

## 5.5 Conclusion

In this paper, we developed an automated framework for synthesizing video action shots from everyday video footage of human subjects. To achieve this, we perform human-aware odometry prediction, estimate the shape and location of the protagonist within the world geometry, and rerender the video while depicting past and future occurrences of the human subject. Experiments with a dedicated evaluation dataset show that our method is able to estimate state-of-the-art human shapes. Our supplementary video with results on a wide variety of video content shows that our framework enables realistic video action shot results.

# 6 Conclusion

This dissertation addressed three applications of novel view synthesis and provides practical solutions that do not require difficult-to-acquire multi-view imagery. To conclude, this chapter will summarize the main contributions and discuss future directions for subsequent research.

## 6.1 Summary of Contributions

Broadly, this dissertation contributes approaches for practical novel view synthesis in space and time. These contributions are summarized below.

1. Softmax Splatting for Video Frame Interpolation:
   (a) Forward warping via softmax splatting.
   (b) Feature pyramids for image synthesis.
2. 3D Ken Burns Effect from a Single Image:
   (a) Semantic-aware depth estimation.
   (b) Context-aware inpainting for view synthesis.
3. Synthesizing Video Action Shots with Human Priors:
   (a) Human-aware odometry estimation.

(b) Depth-based human localization.

(c) Depth-based human reconstruction and rerendering.

## 6.2 Future Directions

Research on novel view synthesis in time in the form of video frame interpolation still focuses on relatively low resolutions. While state-of-the-art approaches can produce high-resolution results, they are computationally expensive and have difficulties dealing with the increased per-pixel motion magnitude. Future research may focus on targeting high-resolution video frame interpolation explicitly.

Research on novel view synthesis in space by augmenting a single image with depth is still in its infancy. While the 3D Ken Burns paper has made great strides towards synthesizing realistic results, its output is hit and miss. Future research may focus on improving the robustness of this extreme form of novel view synthesis, for example by better tailoring the depth estimate to the task of image synthesis.

Research on novel view synthesis in time and space has seen little attention so far. While the paper on video action shot synthesis successfully explored this area, the proposed framework included multiple sophisticated steps. Future research may focus on combining individual aspects of this pipeline, for example by merging the depth-based human localization and the human shape estimation.

## Bibliography

[1] Abarghouei, A.A., Breckon, T.P.: Real-Time Monocular Depth Estimation Using Synthetic Data With Domain Adaptation via Image Style Transfer. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 10

[2] Agudo, A., Pijoan, M., Moreno-Noguer, F.: Image Collection Pop-Up: 3D Reconstruction and Clustering of Rigid and Non-Rigid Categories. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 68

[3] Aliev, K.A., Ulyanov, D., Lempitsky, V.S.: Neural Point-Based Graphics. arXiv/1906.08240 (2019) 13, 49, 79

[4] Alldieck, T., Magnor, M.A., Bhatnagar, B.L., Theobalt, C., Pons-Moll, G.: Learning to Reconstruct People in Clothing From a Single RGB Camera. arXiv/1903.05885 (2019) 13

[5] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: Shape Completion and Animation of People. ACM Transactions on Graphics 24(3), 408–416 (2005) 13

[6] Assa, J., Caspi, Y., Cohen-Or, D.: Action Synopsis: Pose Selection and Illustration. ACM Transactions on Graphics 24(3), 667–676 (2005) 12, 67

[7] Aydin, T.O., Stefanoski, N., Croci, S., Gross, M.H., Smolic, A.: Temporally Coherent Local Tone Mapping of HDR Video. ACM Transactions on Graphics 33(6), 196:1–196:13 (2014) 9

[8] Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A Database and Evaluation Methodology for Optical Flow. International Journal of Computer Vision 92(1), 1–31 (2011) 15, 21, 25, 27, 30, 31

[9] Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-Aware Video Frame Interpolation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 7, 8, 15, 16, 17, 20, 21, 25, 33

[10] Bao, W., Lai, W.S., Zhang, X., Gao, Z., Yang, M.H.: MEMC-Net: Motion Estimation and Motion Compensation Driven Neural Network for Video Interpolation and Enhancement. arXiv/1810.08768 (2018) 7

[11] de Bem, R., Ghosh, A., Ajanthan, T., Miksik, O., Siddharth, N., Torr, P.H.S.: DGPose: Disentangled Semi-Supervised Deep Generative Models for Human Body Analysis. arXiv/1804.06364 (2018) 13

[12] Blau, Y., Michaeli, T.: The Perception-Distortion Tradeoff. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 24, 29

[13] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P.V., Romero, J., Black, M.J.: Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape From a Single Image. In: European Conference on Computer Vision (2016) xiv, 13, 67, 73, 76

[14] Bregler, C., Hertzmann, A., Biermann, H.: Recovering Non-Rigid 3D Shape From Image Streams. In: IEEE Conference on Computer Vision and Pattern Recognition (2000) 68

[15] Brooks, T., Barron, J.T.: Learning to Synthesize Motion Blur. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 11, 15

[16] Bui, G., Le, T., Morago, B., Duan, Y.: Point-Based Rendering Enhancement via Deep Learning. The Visual Computer 34(6-8), 829–841 (2018) 49, 79

[17] Caballero, J., Ledig, C., Aitken, A.P., Acosta, A., Totz, J., Wang, Z., Shi, W.: Real-Time Video Super-Resolution With Spatio-Temporal Networks and Motion Compensation. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 7

[18] Caspi, Y., Axelrod, A., Matsushita, Y., Gamliel, A.: Dynamic Stills and Clip Trailers. The Visual Computer 22(9-11), 642–652 (2006) 12, 67

[19] Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody Dance Now. arXiv/1808.07371 (2018) 13

[20] Chaurasia, G., Duchêne, S., Sorkine-Hornung, O., Drettakis, G.: Depth Synthesis and Local Warps for Plausible Image-Based Navigation. ACM Transactions on Graphics 32(3), 30:1–30:12 (2013) 6

[21] Chaurasia, G., Sorkine, O., Drettakis, G.: Silhouette-Aware Warping for Image-Based Rendering. Computer Graphics Forum 30(4), 1223–1232 (2011) 6

[22] Chen, W., Fu, Z., Yang, D., Deng, J.: Single-Image Depth Perception in the Wild. In: Advances in Neural Information Processing Systems (2016) 10, 13, 59, 60, 61

[23] Cun, X., Xu, F., Pun, C.M., Gao, H.: Depth-Assisted Full Resolution Network for Single Image-Based View Synthesis. In: IEEE Computer Graphics and Applications (2019) 7, 9

[24] Dekel, T., Rubinstein, M., Liu, C., Freeman, W.T.: On the Effectiveness of Visible Watermarks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 70

[25] Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: Structure From Motion Without Correspondence. In: IEEE Conference on Computer Vision and Pattern Recognition (2000) 12

[26] Didyk, P., Sitthi-amorn, P., Freeman, W.T., Durand, F., Matusik, W.: Joint View Expansion and Filtering for Automultiscopic 3D Displays. ACM Transactions on Graphics 32(6), 221:1–221:8 (2013) 6

[27] Dorling, G.W.: Stroboscopic Photography. Physics Education 1(4), 236 (1966) 11

[28] Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning Optical Flow With Convolutional Networks. In: IEEE International Conference on Computer Vision (2015) 29

[29] Eigen, D., Fergus, R.: Predicting Depth, Surface Normals and Semantic Labels With a Common Multi-Scale Convolutional Architecture. In: IEEE International Conference on Computer Vision (2015) 43, 74

[30] Engel, J., Koltun, V., Cremers, D.: Direct Sparse Odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(3), 611–625 (2018) 12, 70

[31] Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. In: European Conference on Computer Vision (2014) 12

[32] Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing Higher-Layer Features of a Deep Network. Tech. rep. (2009) 28

[33] Flynn, J., Neulander, I., Philbin, J., Snavely, N.: DeepStereo: Learning to Predict New Views From the World'S Imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 6

[34] Fourure, D., Emonet, R., Fromont, É., Muselet, D., Trémeau, A., Wolf, C.: Residual Conv-Deconv Grid Network for Semantic Segmentation. In: British Machine Vision Conference (2017) 24, 41, 50, 74

[35] Freeman, W.T., Zhang, H.: Shape-Time Photography. In: IEEE Conference on Computer Vision and Pattern Recognition (2003) 12, 67

[36] Gadde, R., Jampani, V., Gehler, P.V.: Semantic Video CNNs Through Representation Warping. In: IEEE International Conference on Computer Vision (2017) 8

[37] Garg, R., Kumar, B.G.V., Carneiro, G., Reid, I.D.: Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In: European Conference on Computer Vision (2016) 10, 13

[38] Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision Meets Robotics: The KITTI Dataset. International Journal of Robotics Research 32(11), 1231–1237 (2013) 54

[39] Godard, C., Aodha, O.M., Brostow, G.J.: Unsupervised Monocular Depth Estimation With Left-Right Consistency. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 7, 10, 13

[40] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative Adversarial Nets. In: Advances in Neural Information Processing Systems (2014) 33

[41] Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras. arXiv/1904.04998 (2019) 10

[42] Güler, R.A., Neverova, N., Kokkinos, I.: DensePose: Dense Human Pose Estimation in the Wild. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 13, 73, 80, 81, 82, 83

[43] Habtegebrial, T., Varanasi, K., Bailer, C., Stricker, D.: Fast View Synthesis With Deep Stereo Vision. arXiv/1804.09690 (2018) 9

[44] He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-Cnn. In: IEEE International Conference on Computer Vision (2017) xi, 40, 41, 44, 70, 74, 79, 81, 83, 85

[45] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 22, 26

[46] Hedman, P., Alsisan, S., Szeliski, R., Kopf, J.: Casual 3D Photography. ACM Transactions on Graphics 36(6), 234:1–234:15 (2017) 6

[47] Hedman, P., Kopf, J.: Instant 3D Photography. ACM Transactions on Graphics 37(4), 101:1–101:12 (2018) 6

[48] Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.J.: Deep Blending for Free-Viewpoint Image-Based Rendering. ACM Transactions on Graphics 37(6), 257:1–257:15 (2018) 6

[49] Hoiem, D., Efros, A.A., Hebert, M.: Automatic Photo Pop-Up. ACM Transactions on Graphics 24(3), 577–584 (2005) 10

[50] Horry, Y., Anjyo, K.I., Arai, K.: Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation From a Single Image. In: Conference on Computer Graphics and Interactive Techniques (1997) 10, 56

[51] Hsieh, J.T., Liu, B., Huang, D.A., Li, F.F., Niebles, J.C.: Learning to Decompose and Disentangle Representations for Video Prediction. In: Advances in Neural Information Processing Systems (2018) 11

[52] Huang, H., Wang, H., Luo, W., Ma, L., Jiang, W., Zhu, X., Li, Z., Liu, W.: Real-Time Neural Style Transfer for Videos. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 9

[53] Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Temporally Coherent Completion of Dynamic Video. ACM Transactions on Graphics 35(6), 196:1–196:11 (2016) 9

[54] Huang, J., Chen, Z., Ceylan, D., Jin, H.: 6-Dof VR Videos With a Single 360-Camera. In: IEEE Virtual Reality (2017) 6

[55] Hui, T.W., Tang, X., Loy, C.C.: LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7, 8, 23, 28, 29

[56] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 23, 29

[57] Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. In: Advances in Neural Information Processing Systems (2015) 7, 15, 18

[58] Janai, J., Güney, F., Wulff, J., Black, M.J., Geiger, A.: Slow Flow: Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) x, 29, 30

[59] Ji, D., Kwon, J., McFarland, M., Savarese, S.: Deep View Morphing. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 6

[60] Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E.G., Kautz, J.: Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7, 9, 15

[61] Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-Based View Synthesis for Light Field Cameras. ACM Transactions on Graphics 35(6), 193:1–193:10 (2016) 6, 15

[62] Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-To-End Recovery of Human Shape and Pose. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 13, 80, 81, 82

[63] Kang, H.W., Pyo, S.H., ichi Anjyo, K., Shin, S.Y.: Tour Into the Picture Using a Vanishing Line and Its Extension to Panoramic Images. Computer Graphics Forum 20(3), 132–141 (2001) 11

[64] Kang, S.B., Li, Y., Tong, X., Shum, H.Y.: Image-Based Rendering. Foundations and Trends in Computer Graphics and Vision 2(3), 173–258 (2006) 6

[65] Kellnhofer, P., Didyk, P., Wang, S.P., Sitthi-amorn, P., Freeman, W.T., Durand, F., Matusik, W.: 3DTV at Home: Eulerian-Lagrangian Stereo-To-Multiview Conversion. ACM Transactions on Graphics 36(4), 146:1–146:13 (2017) 6

[66] Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv/1412.6980 (2014) 42, 45, 50, 77

[67] Klose, F., Wang, O., Bazin, J.C., Magnor, M.A., Sorkine-Hornung, A.: Sampling Based Scene-Space Video Processing. ACM Transactions on Graphics 34(4), 67:1–67:11 (2015) 6, 12, 67

[68] Koch, T., Liebel, L., Fraundorfer, F., Körner, M.: Evaluation of CNN-based Single-Image Depth Estimation Methods. arXiv/1805.01328 (2018) 10, 60

[69] Kopf, J.: 360°Video Stabilization. ACM Transactions on Graphics 35(6), 195:1–195:9 (2016) 6

[70] Kuroki, Y., Nishi, T., Kobayashi, S., Oyaizu, H., Yoshimura, S.: A Psychophysical Study of Improvements in Motion-Image Quality by Using High Frame Rates. Journal of the Society for Information Display 15(1), 61–68 (2007) 15

[71] Kuroki, Y., Takahashi, H., Kusakabe, M., Yamakoshi, K.i.: Effects of Motion Image Stimuli With Normal and High Frame Rates on EEG Power Spectra:

Comparison With Continuous Motion Image Stimuli. Journal of the Society for Information Display 22(4), 191–198 (2014) 15

[72] Lai, C.J., Han, P.H., Hung, Y.P.: View Interpolation for Video See-Through Head-Mounted Display. In: Conference on Computer Graphics and Interactive Techniques (2016) 6

[73] Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning Blind Video Temporal Consistency. In: European Conference on Computer Vision (2018) 9

[74] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper Depth Prediction With Fully Convolutional Residual Networks. In: International Conference on 3D Vision (2016) 10

[75] Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., Gross, M.H.: Non-linear Disparity Mapping for Stereoscopic 3D. ACM Transactions on Graphics 29(4), 75:1–75:10 (2010) 6

[76] Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic Adversarial Video Prediction. arXiv/1804.01523 (2018) 11

[77] Li, N., Huang, Z.: Tour Into the Picture Revisited. In: Conference on Computer Graphics, Visualization and Computer Vision (2001) 11

[78] Li, Y., Huang, J.B., Ahuja, N., Yang, M.H.: Deep Joint Image Filtering. In: European Conference on Computer Vision (2016) 45

[79] Li, Z., Dekel, T., Cole, F., Tucker, R., Snavely, N., Liu, C., Freeman, W.T.: Learning the Depths of Moving People by Watching Frozen People. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 10, 13

[80] Li, Z., Snavely, N.: MegaDepth: Learning Single-View Depth Prediction From Internet Photos. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 10, 13, 43, 53, 59

[81] Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual Motion GAN for Future-Flow Embedded Video Prediction. In: IEEE International Conference on Computer Vision (2017) 11

[82] Liu, B., Gould, S., Koller, D.: Single Image Depth Estimation From Predicted Semantic Labels. In: IEEE Conference on Computer Vision and Pattern Recognition (2010) 43

[83] Liu, F., Gleicher, M., Jin, H., Agarwala, A.: Content-Preserving Warps for 3D Video Stabilization. ACM Transactions on Graphics 28(3), 44 (2009) 6

[84] Liu, M., He, X., Salzmann, M.: Geometry-Aware Deep Network for Single-Image Novel View Synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7, 9

[85] Liu, Y.L., Liao, Y.T., Lin, Y.Y., Chuang, Y.Y.: Deep Video Frame Interpolation Using Cyclic Frame Generation. In: AAAI Conference on Artificial Intelligence (2019) 7, 9, 16, 25, 33

[86] Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video Frame Synthesis Using Deep Voxel Flow. In: IEEE International Conference on Computer Vision (2017) 7, 25, 30

[87] Long, G., Kneip, L., Alvarez, J.M., Li, H., Zhang, X., Yu, Q.: Learning Image Matching by Simply Watching Video. In: European Conference on Computer Vision (2016) 15

[88] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A Skinned Multi-Person Linear Model. ACM Transactions on Graphics 34(6), 248:1–248:16 (2015) 13

[89] Luo, Y., Ren, J.S.J., Lin, M., Pang, J., Sun, W., Li, H., Lin, L.: Single View Stereo Matching. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 10

[90] Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised Learning of Depth and Ego-Motion From Monocular Video Using 3D Geometric Constraints. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7, 13

[91] Mathieu, M., Couprie, C., LeCun, Y.: Deep Multi-Scale Video Prediction Beyond Mean Square Error. arXiv/1511.05440 (2015) 11

[92] Meghdadi, A.H., Irani, P.: Interactive Exploration of Surveillance Video Through Action Shot Summarization and Trajectory Visualization. IEEE Transactions on Visualization and Computer Graphics 19(12), 2119–2128 (2013) 12, 67

[93] Meister, S., Hur, J., Roth, S.: UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. In: AAAI Conference on Artificial Intelligence (2018) 7

[94] Meshry, M., Goldman, D.B., Khamis, S., Hoppe, H., Pandey, R., Snavely, N., Martin-Brualla, R.: Neural Rerendering in the Wild. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 6, 49, 79

[95] Meyer, S., Cornillère, V., Djelouah, A., Schroers, C., Gross, M.H.: Deep Video Color Propagation. In: British Machine Vision Conference (2018) 15

[96] Meyer, S., Djelouah, A., McWilliams, B., Sorkine-Hornung, A., Gross, M.H., Schroers, C.: PhaseNet for Video Frame Interpolation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7

[97] Meyer, S., Wang, O., Zimmer, H., Grosse, M., Sorkine-Hornung, A.: Phase-Based Frame Interpolation for Video. In: IEEE Conference on Computer Vision and Pattern Recognition (2015) 7

[98] Mildenhall, B., Srinivasan, P.P., Cayon, R.O., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local Light Field Fusion: Practical View Synthesis With Prescriptive Sampling Guidelines. ACM Transactions on Graphics 38(4), 29:1–29:14 (2019) 6

[99] Mousavian, A., Pirsiavash, H., Kosecka, J.: Joint Semantic Segmentation and Depth Estimation With Deep Convolutional Networks. In: International Conference on 3D Vision (2016) 43

[100] Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Robotics 31(5), 1147–1163 (2015) 12, 68, 70

[101] Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Trans. Robotics 33(5), 1255–1262 (2017) 12, 68, 70

[102] Nekrasov, V., Dharmasiri, T., Spek, A., Drummond, T., Shen, C., Reid, I.D.: Real-Time Joint Semantic Segmentation and Depth Estimation Using Asymmetric Annotations. arXiv/1809.04766 (2018) 43

[103] Nguyen, C., Niu, Y., Liu, F.: Video Summagator: An Interface for Video Summarization and Navigation. In: Conference on Human Factors in Computing Systems (2012) 12

[104] Nguyen-Phuoc, T., Li, C., Balaban, S., Yang, Y.L.: RenderNet: A Deep Convolutional Network for Differentiable Rendering From 3D Shapes. In: Advances in Neural Information Processing Systems (2018) 8, 16

[105] Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. arXiv/1904.01326 (2019) 9

[106] Niklaus, S., Liu, F.: Context-Aware Synthesis for Video Frame Interpolation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7, 8, 15, 16, 22, 24, 25, 26, 28, 32, 41, 49

[107] Niklaus, S., Mai, L., Liu, F.: Video Frame Interpolation via Adaptive Convolution. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 7

[108] Niklaus, S., Mai, L., Liu, F.: Video Frame Interpolation via Adaptive Separable Convolution. In: IEEE International Conference on Computer Vision (2017) 7, 16, 25

[109] Niklaus, S., Mai, L., Yang, J., Liu, F.: 3D Ken Burns Effect From a Single Image. ACM Transactions on Graphics 38(6), 184:1–184:15 (2019) xiii, xiv, 71, 72, 74, 78

[110] Odena, A., Dumoulin, V., Olah, C.: Deconvolution and Checkerboard Artifacts. Tech. rep. (2016) 24, 41

[111] Olszewski, K., Tulyakov, S., Woodford, O.J., Li, H., Luo, L.: Transformable Bottleneck Networks. arXiv/1904.06458 (2019) 9

[112] Park, E., Yang, J., Yumer, E., Ceylan, D., Berg, A.C.: Transformation-Grounded Image Generation Network for Novel 3D View Synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 9

[113] Penner, E., Zhang, L.: Soft 3D Reconstruction for View Synthesis. ACM Transactions on Graphics 36(6), 235:1–235:11 (2017) 6

[114] Pritch, Y., Rav-Acha, A., Gutman, A., Peleg, S.: Webcam Synopsis: Peeking Around the World. In: IEEE International Conference on Computer Vision (2007) 12, 67

[115] Pritch, Y., Rav-Acha, A., Peleg, S.: Nonchronological Video Synopsis and Indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(11), 1971–1984 (2008) 12, 67

[116] Pumarola, A., Agudo, A., Porzi, L., Sanfeliu, A., Lepetit, V., Moreno-Noguer, F.: Geometry-Aware Network for Non-Rigid Shape Prediction From a Single View. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 68

[117] Pumarola, A., Sanchez, J., Choi, G., Sanfeliu, A., Moreno-Noguer, F.: 3DPeople: Modeling the Geometry of Dressed Humans. In: IEEE International Conference on Computer Vision (2019) 13

[118] Qi, X., Liao, R., Liu, Z., Urtasun, R., Jia, J.: GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 13, 60

[119] Rahaman, D.M.M., Paul, M.: Virtual View Synthesis for Free Viewpoint Video and Multiview Video Compression Using Gaussian Mixture Modelling. IEEE Transactions on Image Processing 27(3), 1190–1201 (2018) 6

[120] Rakêt, L.L., Roholm, L., Bruhn, A., Weickert, J.: Motion Compensated Frame Interpolation With a Symmetric Optical Flow Constraint. In: Advances in Visual Computing (2012) 7

[121] Ranieri, N., Heinzle, S., Smithwick, Q., Reetz, D., Smoot, L.S., Matusik, W., Gross, M.H.: Multi-Layered Automultiscopic Displays. Computer Graphics Forum 31(7-2), 2135–2143 (2012) 6

[122] Ranjan, A., Black, M.J.: Optical Flow Estimation Using a Spatial Pyramid Network. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 7

[123] Rav-Acha, A., Pritch, Y., Peleg, S.: Making a Long Video Short: Dynamic Video Synopsis. In: IEEE Conference on Computer Vision and Pattern Recognition (2006) 12, 67

[124] Reda, F.A., Liu, G., Shih, K.J., Kirby, R., Barker, J., Tarjan, D., Tao, A., Catanzaro, B.: SDC-Net: Video Prediction Using Spatially-Displaced Convolution. In: European Conference on Computer Vision (2018) 11

[125] Reda, F.A., Sun, D., Dundar, A., Shoeybi, M., Liu, G., Shih, K.J., Tao, A., Kautz, J., Catanzaro, B.: Unsupervised Video Interpolation Using Cycle Consistency. In: IEEE International Conference on Computer Vision (2019) 7, 9, 33

[126] Rematas, K., Kemelmacher-Shlizerman, I., Curless, B., Seitz, S.: Soccer on Your Tabletop. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 6

[127] Rematas, K., Nguyen, C.H., Ritschel, T., Fritz, M., Tuytelaars, T.: Novel Views of Objects From a Single Image. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(8), 1576–1590 (2017) 9

[128] Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In: IEEE International Conference on Computer Vision (2019) 13

[129] Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3D Scene Structure From a Single Still Image. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(5), 824–840 (2009) 10

[130] Schönberger, J.L., Frahm, J.M.: Structure-From-Motion Revisited. In: IEEE Conference on Computer Vision and Pattern Recognition (2016) 12, 68, 70

[131] Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise View Selection for Unstructured Multi-View Stereo. In: European Conference on Computer Vision (2016) 12, 68, 70

[132] Shysheya, A., Zakharov, E., Aliev, K.A., Bashirov, R., Burkov, E., Iskakov, K., Ivakhnenko, A., Malkov, Y., Pasechnik, I., Ulyanov, D., Vakhitov, A., Lempitsky, V.S.: Textured Neural Avatars. arXiv/1905.08776 (2019) 13

[133] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference From RGBD Images. In: European Conference on Computer Vision (2012) 10, 43, 54, 60

[134] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv/1409.1556 (2014) xi, 40, 41, 50, 74

[135] Sitzmann, V., Thies, J., Heide, F., Niessner, M., Wetzstein, G., Zollhofer, M.: DeepVoxels: Learning Persistent 3D Feature Embeddings. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 6

[136] Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: Exploring Photo Collections in 3D. ACM Transactions on Graphics 25(3), 835–846 (2006) 12

[137] Soomro, K., Zamir, A.R., Shah, M.: UCF101: A Dataset of 101 Human Actions Classes From Videos in the Wild. arXiv/1212.0402 (2012) 25

[138] Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the Boundaries of View Extrapolation With Multiplane Images. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 6

[139] Srinivasan, P.P., Wang, T., Sreelal, A., Ramamoorthi, R., Ng, R.: Learning to Synthesize a 4D RGBD Light Field From a Single Image. In: IEEE International Conference on Computer Vision (2017) 9

[140] Srivastava, S., Saxena, A., Theobalt, C., Thrun, S., Ng, A.Y.: I23 - Rapid Interactive 3D Reconstruction From a Single Image. In: Vision, Modeling, and Visualization Workshop (2009) 10

[141] Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7, 8, 23, 28, 29

[142] Tang, S., Tan, F., Cheng, K., Li, Z., Zhu, S., Tan, P.: A Neural Network for Detailed Human Depth Estimation From a Single Image. In: IEEE International Conference on Computer Vision (2019) xiv, 13, 73, 76, 80, 81, 83

[143] Tao, X., Gao, H., Liao, R., Wang, J., Jia, J.: Detail-Revealing Deep Video Super-Resolution. In: IEEE International Conference on Computer Vision (2017) 7

[144] Tatarchenko, M., Dosovitskiy, A., Brox, T.: Single-View to Multi-View: Reconstructing Unseen Views With a Convolutional Network. arXiv/1511.06702 (2015) 9

[145] Teramoto, O., Park, I.K., Igarashi, T.: Interactive Motion Photography From a Single Image. The Visual Computer 26(11), 1339–1348 (2010) 12

[146] Thies, J., Zollhöfer, M., Nießner, M.: Deferred Neural Rendering: Image Synthesis Using Neural Textures. ACM Transactions on Graphics 38(4), 66:1–66:12 (2019) 6

[147] Thies, J., Zollhöfer, M., Theobalt, C., Stamminger, M., Nießner, M.: IGNOR: Image-Guided Neural Object Rendering. arXiv/1811.10720 (2018) 6

[148] Tulsiani, S., Tucker, R., Snavely, N.: Layer-Structured 3D Scene Inference via View Synthesis. In: European Conference on Computer Vision (2018) 9, 48

[149] Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: DeMoN: Depth and Motion Network for Learning Monocular Stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 10, 13, 42, 74

[150] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning From Synthetic Humans. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) xiv, 13, 73, 75, 76, 77

[151] Vondrick, C., Pirsiavash, H., Torralba, A.: Generating Videos With Scene Dynamics. In: Advances in Neural Information Processing Systems (2016) 11

[152] Wadhwa, N., Garg, R., Jacobs, D.E., Feldman, B.E., Kanazawa, N., Carroll, R., Movshovitz-Attias, Y., Barron, J.T., Pritch, Y., Levoy, M.: Synthetic Depth-Of-Field With a Single-Camera Mobile Phone. ACM Transactions on Graphics 37(4), 64:1–64:13 (2018) 10

[153] Wang, L., Shen, X., Zhang, J., Wang, O., Lin, Z.L., Hsieh, C.Y., Kong, S., Lu, H.: DeepLens: Shallow Depth of Field From a Single Image. ACM Transactions on Graphics 37(6), 245:1–245:11 (2018) 10, 59

[154] Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion Aware Unsupervised Learning of Optical Flow. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 7

[155] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004) 25

[156] Weng, C.Y., Curless, B., Kemelmacher-Shlizerman, I.: Photo Wake-Up: 3D Character Animation From a Single Photo. arXiv/1812.02246 (2018) 13

[157] Wu, C.Y., Singhal, N., Krähenbühl, P.: Video Compression Through Image Interpolation. In: European Conference on Computer Vision (2018) 15

[158] Wulff, J., Black, M.J.: Temporal Interpolation as an Unsupervised Pretraining Task for Optical Flow Estimation. In: German Conference on Pattern Recognition (2018) 15

[159] Xian, K., Shen, C., Cao, Z., Lu, H., Xiao, Y., Li, R., Luo, Z.: Monocular Relative Depth Perception With Web Stereo Data Supervision. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 10

[160] Xiao, J., Chai, J., Kanade, T.: A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. In: European Conference on Computer Vision (2004) 68

[161] Xie, J., Girshick, R.B., Farhadi, A.: Deep3D: Fully Automatic 2d-To-3d Video Conversion With Deep Convolutional Neural Networks. In: European Conference on Computer Vision (2016) 6, 10

[162] Xu, J., Ni, B., Li, Z., Cheng, S., Yang, X.: Structure Preserving Video Prediction. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 11

[163] Xu, N., Price, B.L., Cohen, S., Huang, T.S.: Deep Image Matting. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 80, 81, 83, 85

[164] Xu, Z., Bi, S., Sunkavalli, K., Hadap, S., Su, H., Ramamoorthi, R.: Deep View Synthesis From Sparse Photometric Images. ACM Transactions on Graphics 38(4), 76:1–76:13 (2019) 6

[165] Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video Enhancement With Task-Oriented Flow. International Journal of Computer Vision 127(8), 1106–1125 (2019) 7, 8, 16, 23, 25, 27, 29, 30, 33

[166] Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction Without 3D Supervision. In: Advances in Neural Information Processing Systems (2016) 9

[167] Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-Supervised Disentangling With Recurrent Transformations for 3D View Synthesis. In: Advances in Neural Information Processing Systems (2015) 9

[168] Yin, Z., Shi, J.: GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 13

[169] Yu, J.J., Harley, A.W., Derpanis, K.G.: Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. In: ECCV Workshops (2016) 7

[170] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 26

[171] Zhang, X., Dekel, T., Xue, T., Owens, A., He, Q., Wu, J., Mueller, S., Freeman, W.T.: MoSculp: Interactive Visualization of Shape and Time. In: ACM Symposium on User Interface Software and Technology (2018) 12, 67, 69

[172] Zheng, K.C., Colburn, A., Agarwala, A., Agrawala, M., Salesin, D., Curless, B., Cohen, M.F.: Parallax Photography: Creating 3D Cinematic Effects From Stills. In: Graphics Interface Conference (2009) 11

[173] Zheng, K., Zha, Z.J., Cao, Y., Chen, X., Wu, F.: LA-Net: Layout-Aware Dense Network for Monocular Depth Estimation. In: ACM Multimedia (2018) 10

[174] Zheng, Z., Yu, T., Wei, Y., Dai, Q., Liu, Y.: DeepHuman: 3D Human Reconstruction From a Single Image. arXiv/1903.06473 (2019) 13

[175] Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised Learning of Depth and Ego-Motion From Video. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 7, 10, 13

[176] Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo Magnification: Learning View Synthesis Using Multiplane Images. ACM Transactions on Graphics 37(4), 65:1–65:12 (2018) 6

[177] Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View Synthesis by Appearance Flow. In: European Conference on Computer Vision (2016) 7, 9

[178] Zhou, Y., Wang, Z., Fang, C., Bui, T., Berg, T.L.: Dance Dance Generation: Motion Transfer for Internet Videos. arXiv/1904.00129 (2019) 13

[179] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In: IEEE International Conference on Computer Vision (2017) 33

[180] Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S.A.J., Szeliski, R.: High-Quality Video View Interpolation Using a Layered Representation. ACM Transactions on Graphics 23(3), 600–608 (2004) 6

[181] Zoran, D., Isola, P., Krishnan, D., Freeman, W.T.: Learning Ordinal Relationships for Mid-Level Vision. In: IEEE International Conference on Computer Vision (2015) 60

# Appendix: Supplemental Files

The following files are provided in the supplementary material, they show video examples for each of the three main chapters.

1. `chapter-3.avi` A video that shows the results of the proposed softmax splatting for video frame interpolation. Totalling 100 megabytes in size, best viewed using the VLC media player.

2. `chapter-4.avi` A video that shows the results of the proposed 3D Ken Burns effect from a single image. Totalling 100 megabytes in size, best viewed using the VLC media player.

3. `chapter-5.avi` A video that shows the results of the proposed video action shot synthesis framework. Totalling 100 megabytes in size, best viewed using the VLC media player.