

Learning-based Face Reconstruction and Editing

A dissertation submitted towards the degree
Doctor of Engineering
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Hyeongwoo Kim

Saarbrücken
2019

Dean of the Faculty:

Prof. Dr. Sebastian Hack

Defense:

12th December, 2019, in Saarbrücken

Chair of the Committee:

Prof. Dr. Jürgen Steimle

Examiners:

Prof. Dr. Christian Theobalt

Dr. Shahram Izadi

Prof. Dr. Christian Richardt

Academic Assistant:

Dr. Mohamed Elgharib

Abstract

Photo-realistic face editing – an important basis for a wide range of applications in movie and game productions, and applications for mobile devices – is based on computationally expensive algorithms that often require many tedious time-consuming manual steps. This thesis advances state-of-the-art face performance capture and editing pipelines by proposing machine learning-based algorithms for high-quality inverse face rendering in real time and highly realistic neural face rendering, and a video-based refocusing method for faces and general videos. In particular, the proposed contributions address fundamental open challenges towards real-time and highly realistic face editing. The first contribution addresses face reconstruction and introduces a deep convolutional inverse rendering framework that jointly estimates all facial rendering parameters from a single image in real time. The proposed method is based on a novel boosting process that iteratively updates the synthetic training data to better reflect the distribution of real-world images. Second, the thesis introduces a method for face video editing at previously unseen quality. It is based on a generative neural network with a novel space-time architecture, which enables photo-realistic re-animation of portrait videos using an input video. It is the first method to transfer the full 3D head position, head rotation, face expression, eye gaze and eye blinking from a source actor to a portrait video of a target actor. Third, the thesis contributes a new refocusing approach for faces and general videos in postprocessing. The proposed algorithm is based on a new depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video and the focus distance for each frame. The high-quality results shown with various applications and challenging scenarios demonstrate the contributions presented in the thesis, and also show potential for machine learning-driven algorithms to solve various open problems in computer graphics.

Kurzzusammenfassung

Fotorealistische Gesichtsbearbeitung ist eine wichtige Grundlage für eine breite Palette von Anwendungen in Film- und Spielproduktionen sowie für mobile Geräte. Sie basiert auf rechenintensiven Algorithmen, die oft aufwändige manuelle Schritte erfordern. Diese Arbeit entwickelt moderne Pipelines zum Erfassen und Bearbeiten von Gesichtern mittels auf maschinellem Lernen basierenden Algorithmen, die ein qualitativ hochwertiges inverses Gesichtsrendering in Echtzeit, ein sehr realistisches neuronales Gesichtsrendering und eine videobasierte Refokussierungsmethode für Gesichter und allgemeine Videos ermöglicht. Mit den vorgeschlagenen Beiträgen werden insbesondere grundlegende Herausforderungen an die Echtzeitbearbeitung und hochrealistische Gesichtsbearbeitung angesprochen. Der erste Beitrag befasst sich mit der Gesichtsrekonstruktion und führt ein CNN-basiertes Rendering-Framework ein, das alle Gesichtsmodellparameter in Echtzeit aus einem einzigen Bild schätzt. Das vorgeschlagene Verfahren basiert auf einem neuartigen Boosting-Prozess, der die synthetischen Trainingsdaten iterativ aktualisiert, um die Verteilung der realen Bilder besser widerzuspiegeln. Zweitens führt die Dissertation eine Methode zur Gesichtsvideobearbeitung in bisher nicht gekannter Qualität ein. Sie basiert auf einem generativen neuronalen Netzwerk mit einer neuartigen Raum-Zeit-Architektur, die eine fotorealistische Re-Animation von Porträtvideos mithilfe eines Eingabevideos ermöglicht. Es ist die erste Methode, die die vollständige 3D-Kopfposition, Kopfdrehung, Gesichtsausdruck, Augenblick und -blinzeln von einem Ursprungsdarsteller auf ein Porträtvideo eines Zieldarstellers übertragen kann. Drittens steuert die Dissertation einen neuen Ansatz für eine nachträgliche Refokussierung von Gesichtern und allgemeinen Videos bei. Der vorgeschlagene Algorithmus benutzt Schärfentiefe, um räumlich-zeitkohärente Tiefenkarten, durchgängig scharfe Bilder sowie den Fokusabstand für jedes Videobild zu berechnen. Die qualitativ hochwertigen Ergebnisse, die mit verschiedenen Anwendungen und schwierigen Szenarien gezeigt werden, demonstrieren die in der Dissertation vorgestellten Beiträge und zeigen auch das Potenzial für von maschinellem Lernen gesteuerten Algorithmen zur Lösung verschiedener offener Probleme in der Computergrafik.

Summary

A full digitization of the human face, body and potentially even behavior is a long-standing goal in computer graphics and computer vision. The movie and game industries have harnessed computer graphics technologies and brought computer-generated special effects into the mainstream. Nowadays, the cutting-edge advances of computer graphics technologies allow us to digitize human characters into a virtual 3D world, render them in new digital worlds, and apply various visual effects to the digital characters for better storytelling in modern movies and games. Despite these advances, it is still considered challenging to model human faces in terms of appearance and geometry. To improve the photorealism of digital face models, the movie industry usually goes through several production steps which involve computationally expensive 3D reconstruction algorithms in a studio setup and tedious manual corrections by skilled artists in postproduction. Furthermore, most post-production processes resort to model-based face editing methods that modify facial attributes with pre-defined models of facial appearance and geometry. These models can scale across various face identities, but usually fail to represent the complex individual details. It is known that we feel uncanny when looking at the digital faces generated by such pre-defined models. Lens effects are another important factor for us to accept synthesized videos as photorealistic. Recent computational methods that enable focus editing of digital human models need to modify the hardware design of the camera and its optics, or require additional hardware, which is inconvenient.

This thesis is motivated by the need for real-time and highly realistic 3D face modeling and editing, and additional focus editing effects for faces in the visual effect pipeline. Particularly, the thesis addresses the limitations of the current state-of-the-art methods, and develops robust algorithms that push the boundaries further. As will be discussed later, each section presents the contributions step by step starting from high-fidelity 3D face modeling in real time towards highly realistic facial animation and focus editing. Note that the goal is ambitious and challenging due to the real-time and photorealism constraints that are essential for high-performance face editing pipelines. In a nutshell, the thesis presents a learning-based framework in combination with a model-based method for real-time inverse rendering from in-the-wild face images, highly realistic re-animation of portrait videos, and a video-based focus editing method for faces as well as general scenes. These methods can potentially be integrated into a unified framework to perform more advanced editing tasks for movie and mobile applications. As a proof of concept, the proposed methods are applied on various real-world applications such as inverse face rendering, face reenactment, visual dubbing of foreign language movies, interactive face editing,

postproduction, video conferencing, refocusing, tilt-shift editing and dolly-zoom photography.

The technical contributions of this thesis can be divided into three main areas: inverse rendering, neural rendering and focus editing.

Inverse Rendering In this area, we contribute a new method for real-time inverse face rendering with high fidelity from in-the-wild monocular images. Chapter 3 introduces InverseFaceNet, a deep convolutional neural network framework that jointly estimates facial pose, shape, expression, reflectance and illumination from a single face input image. The proposed framework benefits from a real-time graphics pipeline that automatically generates a synthetic face database annotated with facial rendering parameters at a large scale to train the neural network. To better reflect the distribution of real-world imagery, a new boosting process is further proposed in the network training loop, which iteratively updates the synthetic training dataset. With all the facial parameters estimated from a single input, advanced editing possibilities, such as appearance editing and relighting, become feasible in real time.

Neural Rendering Chapter 4 presents a novel approach that enables highly realistic re-animation of portrait videos using an input video. The approach is built upon a generative neural network with a novel space-time architecture, that takes as input synthetic renderings of a parametric face model and converts them into video frames with a high level of photorealism. Full control of the target face video is achieved by feeding into the trained network the synthetic renderings of modified facial parameters. Note that this framework automatically makes hair, body and background comply with the edited face images. To the best of our knowledge, this is the first approach that shows the possibility of editing portrait videos in all dimensions of full 3D head position, head rotation, face expression, eye gaze and eye blinking with a high level of photorealism. The potential of the framework is demonstrated with a large variety of video rewrite applications such as face reenactment, visual dubbing, postproduction, gaze correction and video teleconferencing.

Focus Editing The contribution in this area is on focus capture and editing for faces in post-production. Chapter 5 presents a new video-based depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video and the focus distance for each frame from a commodity video camera. Unlike existing computational methods that directly capture depth from light-field imaging or active RGB-D cameras, this framework only requires a video input in which the focus plane is continuously moving back and forth during capture, and thus defocus blur is provoked and strongly visible. With the ability to recover all the focus settings, all-in-focus video and depth maps from a commodity video camera, many compelling video post-processing effects, in particular aesthetic focus editing and refocusing effects, become feasible.

To summarize, the thesis presents integral and robust methods towards a highly realistic face editing framework. In particular, each section in the thesis addresses the technical limitations of model-based inverse face rendering and editing methods, and hardware-based focus editing frameworks, respec-

tively. The main technical contributions advance the state of the art in monocular 3D face reconstruction and editing, and video-based focus editing technologies. The results presented throughout various application scenarios show great potential for real-time and highly realistic face editing in movie production, home video editing and mobile applications.

Contents

1	Introduction	15
1.1	Motivation	15
1.2	Scope	16
1.3	Structure	17
1.4	Summary of Technical Chapters	17
1.5	Technical Contributions	19
1.6	List of Publications	20
2	Basics	23
2.1	Camera, Lens and Image Formation Models	23
2.1.1	Camera Model	23
2.1.2	Lens Model	24
2.1.3	Image Formation Model	25
2.2	Facial Geometry	26
2.2.1	Parameteric Face Representation	26
2.2.2	Blendshapes	26
2.3	Deep Learning	27
3	Face Reconstruction	31
3.1	Introduction	32
3.2	Related Work	33
3.3	Overview	35
3.4	The Space of Facial Imagery	36
3.4.1	Affine Face Model	36
3.4.2	Image Formation	37
3.5	Initial Synthetic Training Corpus	37
3.6	InverseFaceNet	38
3.6.1	Network Architecture	38
3.6.2	Model-Space Parameter Loss	39
3.7	Self-Supervised Boosting	39
3.7.1	Boosting	39
3.7.2	Algorithm	40

3.8	Experiments and Results	41
3.8.1	Evaluation of Design Choices	43
3.8.2	Quantitative Evaluation	44
3.8.3	Qualitative Evaluation	46
3.9	Limitations	46
3.10	Summary	48
4	Face Editing	49
4.1	Introduction	50
4.2	Related Work	51
4.3	Overview	53
4.4	Monocular Face Reconstruction	55
4.5	Synthetic Conditioning Input	57
4.6	Rendering-to-Video Translation	58
4.7	Results	60
4.7.1	Applications	63
4.7.2	Quantitative Evaluation	67
4.7.3	Comparisons to the State-of-the-Art	68
4.7.4	User Study	70
4.8	Dataset	74
4.9	Discussion	74
4.10	Summary	76
5	Focus Editing	79
5.1	Introduction	80
5.2	Related Work	81
5.3	Overview	83
5.4	All-In-Focus RGB-D Video Recovery	84
5.4.1	Patch-Based Defocus-Preserving Alignment	86
5.4.2	Filtering-Based Depth Estimation	88
5.4.3	Defocus Deblurring	90
5.4.4	Focus Distance Refinement	91
5.4.5	Initialization and Implementation	93
5.5	Results and Evaluation	95
5.6	Applications	101
5.7	Summary	102
6	Conclusion	103
6.1	Summary and Discussion	104
6.2	Alternatives	105
6.2.1	Model-based Face Autoencoder	105

6.2.2	Multi-level Face Model	106
6.3	Future Work and Outlook	106
6.3.1	Challenges	106
6.3.2	Detection and Verification	107
6.4	Closing Remarks	108

Chapter 1

Introduction

This chapter presents the motivation and the goal of the thesis, and explains how the thesis is outlined to cover technical challenges. It also summarizes the technical contributions and the peer-reviewed scientific papers that the thesis encompasses.

1.1 Motivation

Digital face editing is of importance to both professional and consumer-level media creation applications, as witnessed by VFX productions and the hundreds of hours of video footage that are uploaded to online communities every minute. High-quality face reconstruction and editing frameworks have facilitated a professional level of visual effects on 3D face models in the post-production process of filmmaking. Recent advances that enhance the usability of face editing technologies have also led non-expert users to create digital contents in home video and mobile applications with much less effort.

In the classical computer graphics pipeline, face modeling and editing are achieved by multiple stages: face reconstruction, editing and compositing. The face reconstruction step recovers the high-quality 3D face models of actors in professional studios. In the second step, trained artists add visual effects to the 3D face model by re-writing the facial appearance and geometry. Finally, the modified 3D face model is rendered back to the image frame by compositing with other layers such as background and illumination. Recently, many approaches [Thies et al. 2016; Averbuch-Elor et al. 2017; Suwajanakorn et al. 2017] for face editing have been introduced with promising results. These methods first reconstruct and track the target face model which is represented as dense 3D points or a set of sparse 2D landmarks. Face editing, e.g., expression change, is then applied, transferring the facial expression of the source face model via modified rendering parameters [Thies et al. 2016] or dense 2D motion fields [Averbuch-Elor et al. 2017]. Alternatively, a recurrent neural network can be employed to synthesize mouth texture to match the input audio track [Suwajanakorn et al. 2017]. However, most of the approaches consume a lot of computing resources to jointly or independently estimate all face rendering parameters, i.e., facial pose, shape, expression, reflectance and illumination. In addition, these methods can modify facial appearance and geometry only within the space of a coarser 3D face

model. As a result, they only allow to modify the face interior, e.g., facial expressions. Moreover, additional lens effects have been ignored in these methods.

Several open challenges require to be addressed to further improve computational efficiency as well as visual realism in the state-of-the-art face editing pipelines. First, the inverse rendering process, i. e., the reconstruction of 3D face models from images and videos, has to be of high efficiency, preferably real-time. The model-based face reconstruction methods often require 3D scanning systems [Huang et al. 2004; Wang et al. 2004; Weise et al. 2009] or multiview camera setups [Beeler et al. 2011; Beeler and Bradley 2014]. Such systems allow us to capture high-quality 3D face model. However, it comes at a high cost and also demands high computational power to process the captured data [Garrido et al. 2015; Thies et al. 2016]. Second, the level of detail that 3D face models can represent has to be improved beyond the uncanny valley [Alexander et al. 2010]. Despite the advances of computer graphics over decades [Blanz and Vetter 1999; Blanz et al. 2004; Suwajanakorn et al. 2015b; Averbuch-Elor et al. 2017], computer-generated face images still provide us with a sense of repulsion or distaste. Third, camera lens effects have to be integrated in the final rendering process. The lens effect, e.g., defocus blur, is an important cue that affects the human perception of depth [Mather 1996; Held et al. 2010] and visual realism. Existing computational methods [Isaksen et al. 2000; Moreno-Noguer et al. 2007; Yu and Gallup 2014; Barron et al. 2015] often require to employ light-field cameras [Ng et al. 2005] or modify the camera optics [Levin et al. 2007] in order to apply lens effects to face models, which is cumbersome and inconvenient.

The thesis is motivated by the aforementioned open challenges concerning highly efficient and realistic face editing frameworks. More precisely, we address several core limitations of the state of the art in face reconstruction and editing, and focus editing technologies. In short, we facilitate machine learning, in particular deep learning-based methods, in combination with computer graphics pipelines, to jointly estimate all face rendering parameters from single images in real time, and also to enable fully controllable face editing with a high level of photorealism. In addition, we develop a video-based method that enables various focus editing effects from an unmodified commodity video camera.

1.2 Scope

The goal of the thesis is to develop efficient methods that capture and modify facial performances and lens effects at high fidelity. This brings up several technical challenges, and therefore we make certain assumptions to make our goal tractable. The input videos are assumed to contain no strong cast shadows and illumination changes as it confuses face reflectance with lighting conditions. We also assume that there are no fast and shaky motions in the input video to robustly track face models and to separate a defocus blur from a motion blur.

With these assumptions, we make the technical contributions in the following areas: face reconstruction, face editing and focus modification. First, the contributions for face reconstruction include a discriminative model of the inverse rendering process with convolutional neural networks, and a boost-

ing framework to reduce the domain gap between synthetic and real face images using a large-scale face dataset without annotation. This serves as the groundwork for efficient editing possibilities such as appearance editing and relighting. Second, the advance in face editing is a novel rendering framework that capitalizes on model-based 3D face reconstruction in combination with generative adversarial networks to enable fully controllable face manipulation with a high level of photorealism. Finally, an improvement on focus editing covers a video-based focus editing framework that recovers lens models, space-time-coherent depth maps and all-in-focus video. The potential of the proposed methods is demonstrated on various application scenarios such as single-shot inverse rendering, relighting, face reenactment, visual dubbing, interactive face editing, postproduction, video teleconferencing, refocusing, tilt-shift editing and dolly-zoom photography.

We structure the aforementioned contributions of the thesis according to the steps needed for face capture and editing. This organization helps to better emphasize the contributions of the subtopics. Moreover, it gradually shows the improvements made by each contribution, and illustrates the capabilities of the proposed methods in various application scenarios.

1.3 Structure

We divide the thesis into six chapters to cover the main technical contributions. A brief overview is given as follows.

Chapter 1 provides an introduction to the thesis topic, a statement of the goal and scope, an outline of the thesis structure, a summary of the technical chapters, and an emphasis of the main technical contributions.

Chapter 2 describes the fundamental principles and the mathematical notations that are used throughout the thesis. These are mainly concerned with 3D face modeling, lens models and neural networks.

Chapters 3–5 present the main technical contributions. As mentioned earlier, these chapters are structured to emphasize the advances on the face editing pipeline. Improvements are discussed at the end of each chapter and linked to subsequent chapters. Moreover, each chapter presents challenging application scenarios that demonstrate the potential and capabilities of the contributions.

Chapter 6 summarizes the main contributions and results, and it briefly discusses future challenges which are not explored in the thesis. Furthermore, it gives an outlook towards detection and verification of face images modified with a high level of photorealism.

1.4 Summary of Technical Chapters

This section gives a more detailed overview of the technical chapters of the thesis.

Chapter 3 introduces InverseFaceNet, a deep convolutional inverse rendering framework for faces that

jointly estimates facial pose, shape, expression, reflectance and illumination from a single input image. By estimating all parameters from just a single image, advanced editing possibilities on a single face image, such as appearance editing and relighting, become feasible in real time. Most previous learning-based face reconstruction approaches do not jointly recover all dimensions, or are severely limited in terms of visual quality. In contrast, the proposed method described in this chapter recovers high-quality facial pose, shape, expression, reflectance and illumination using a deep neural network that is trained using a large, synthetically created training dataset. It builds on a novel loss function that measures model-space similarity directly in parameter space and significantly improves reconstruction accuracy. A new boosting process is further proposed in the network training loop, which iteratively updates the synthetic training dataset to better reflect the distribution of real-world imagery. Quantitative validations demonstrate that this strategy outperforms completely synthetically trained networks. In addition, comparisons to several state-of-the-art approaches and high-quality relighting results are provided.

Next, Chapter 4 presents a novel approach that enables highly realistic re-animation of portrait videos using only an input video of a person. In contrast to existing approaches that are restricted to modification of facial expressions only, this is the first to transfer the full 3D head position, head rotation, face expression, eye gaze, and eye blinking from a source actor to a portrait video of a target actor. The core of the approach is a generative neural network with a novel space-time architecture. The network takes as input synthetic renderings of a parametric face model, based on which it predicts highly realistic video frames for a given target actor. The realism in this rendering-to-video transfer is achieved by careful adversarial training, and as a result, it can create modified target videos that mimic the behavior of the synthetically-created input. In order to enable source-to-target video re-animation, it renders a synthetic target video with the reconstructed head animation parameters from a source video, and feed it into the trained network – thus taking full control of the target. With the ability to freely recombine source and target parameters, it demonstrates a large variety of video rewrite applications without explicitly modeling hair, body or background. For instance, it can reenact the full head using interactive user-controlled editing, and realize high-fidelity visual dubbing, postproduction and gaze correction for video teleconferencing. An extensive series of experiments and evaluations demonstrate the high quality of our output, where for instance a user study shows that our video edits are hard to detect.

Finally, Chapter 5 addresses the problem of dynamic refocusing for faces and general scenes. Many compelling video effects can be performed in post-processing if a video is given in the form of an all-in-focus video with per-frame depth maps and focus distances. In particular, this enables a variety of focus editing effects, such as video refocusing, which are important stylistic elements in video. Recent computational methods that allow to capture such information in an easy and robust manner modify the hardware design of the camera and its optics, or require additional hardware. Hence, they are less practical and unavailable to normal users with commodity cameras. We therefore presents an algorithm to capture all-in-focus RGB-D video of dynamic scenes with commodity video cameras that are unmodified and need no special calibration. Our algorithm turns defocus blur – an effect often regarded as an unwanted artifact – into a valuable signal. The input to our method is a video in which the focus plane is continuously moving back and forth during capture, and thus defocus blur

is provoked and strongly visible. This can be achieved by manually turning the focus ring of the lens during recording. The core algorithmic ingredient is a new video-based depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video, and the focus distance for each frame. Compelling video post-processing effects, such as different types of refocusing, illustrate the effectiveness of the proposed method.

1.5 Technical Contributions

In the following, we provide a more detailed list of technical contributions that enable the methods described above.

The main contributions of Chapter 3 are outlined as follows:

- A real-time and deep inverse face rendering network that estimates pose, shape, expression, color reflectance and illumination from just a single input image in a single forward pass.
- A loss function that measures model-space distances directly in a modified parameter space.
- A boosting framework that reduces the domain gap between synthetic and real-world parameter distribution.

The main contributions of Chapter 4 are summarized as follows:

- A rendering-to-video translation network that transforms coarse face model renderings into full photo-realistic portrait video output.
- A novel space-time encoding as conditional input for temporally coherent video synthesis that represents face geometry, reflectance, and motion as well as eye gaze and eye blinks.
- A comprehensive evaluation on several applications to demonstrate the flexibility and effectiveness of our approach.

The main contributions of Chapter 5 are:

- A hierarchical alignment scheme between video frames of different focus settings and dynamic scene contents.
- An approach to estimate per-frame depth maps and deblurred all-in-focus color images in a spacetime coherent way.
- An image-guided algorithm for focus distance initialization, and an optimization method for refining focus distances.

1.6 List of Publications

The methods presented in the thesis encompass peer-reviewed scientific papers published at conferences and journals in the field of computer graphics and vision. These papers listed below independently address the aforementioned technical challenges towards highly efficient and realistic face editing.

- H. Kim, C. Richardt and C. Theobalt. “Video depth-from-defocus”. In *3DV*, 370–379, 2016.
- H. Kim, M. Zollhöfer, A. Tewari, J. Thies, C. Richardt and C. Theobalt. “InverseFaceNet: Deep monocular inverse face rendering”. In *CVPR*, 4625–4634, 2018.
- H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollhöfer and C. Theobalt. “Deep video portraits”. In *ACM TOG (Proc. SIGGRAPH)*, 37(4), 163:1–163:14, 2018.
- H. Kim, M. Elgharib, M. Zollhöfer, H.-P. Seidel, T. Beeler, C. Richardt and C. Theobalt. “Neural style-preserving visual dubbing”. In *ACM TOG (Proc. SIGGRAPH Asia)*, 38(6), 178:1–178:13, 2019.

The co-authored papers that address the problem of wide-baseline scene flow, face reconstruction with a model-based autoencoder and a multi-layer model, and neural rendering for human actor videos are listed below.

- C. Richardt, H. Kim, L. Valgaerts and C. Theobalt. “Dense wide-baseline scene flow from two handheld video cameras”. In *3DV*, 276–285, 2016.
- A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez and C. Theobalt. “Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction”. In *ICCV*, 3735–3744, 2017.
- A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez and C. Theobalt. “Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz”. In *CVPR*, 2549–2559, 2018.
- L. Liu, W. Xu, M. Zollhöfer, H. Kim, F. Bernard, M. Habermann, W. Wang and C. Theobalt. “Neural rendering and reenactment of human actor videos”. In *ACM TOG (Proc. SIGGRAPH)*, 38(5), 139:1–139:14, 2019.
- A. Tewari, M. Zollhöfer, F. Bernard, P. Garrido, H. Kim, P. Pérez and C. Theobalt. “High-fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder”. In *IEEE TPAMI*, 42(2), 357–370, 2020.

- L. Liu, W. Xu, M. Habermann, M. Zollhöfer, F. Bernard, H. Kim, W. Wang and C. Theobalt. “Neural human video rendering: Joint learning of dynamic textures and rendering-to-video translation”. In *IEEE TVCG*, 2020.

Chapter 2

Basics

This chapter provides an overview of the technical background of the thesis. Each section discusses the details about an advanced camera model with an optical lens, face-specific geometric models represented as triangle meshes, and convolutional neural networks – the main mathematical tool for the thesis.

2.1 Camera, Lens and Image Formation Models

In the thesis, 3D face models are rendered back to screen space through a camera model. We briefly describe the mathematical relationship between a point in 3D space and its projection onto the image plane. In addition, we provide preliminaries on a camera lens model which enables to add lens effects to rendered images, and a reflection model used in the thesis.

2.1.1 Camera Model

A camera model allows a projection of the 3D geometry, for instance vertices of a mesh or a 3D point coordinate, onto a 2D image space, as Figure 2.1 shows. For computational convenience, a 3D face vertex \mathbf{v} is first represented in camera coordinates via a Euclidean transformation from a world space to a camera space, i. e., 3D rotation and translation. A pinhole camera model [Forsyth and Ponce 2012], commonly used in computer graphics and vision, then takes the 3D vertex along the optical ray converging at the camera center, providing its corresponding 2D projection \mathbf{p} on a 2D image plane as follows:

$$\mathbf{p}(\mathbf{K}, \mathbf{R}, \mathbf{t}) = \mathbf{K}\Pi(\mathbf{R}\mathbf{v} + \mathbf{t}) = \mathbf{K}\Pi(\hat{\mathbf{v}}) \quad , \quad (2.1)$$

where $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3 \times 4}$ is the camera extrinsic parameters describing the rigid transformation between the 3D object and camera coordinates. $\hat{\mathbf{v}}$ refers to the same 3D point represented with respect to the camera coordinate system. A projection operator $\Pi(\cdot)$ takes the aligned 3D point $\hat{\mathbf{v}}$ onto the 2D image plane in a normalized coordinate system. Camera intrinsics \mathbf{K} , involving the focal length f and the principal point

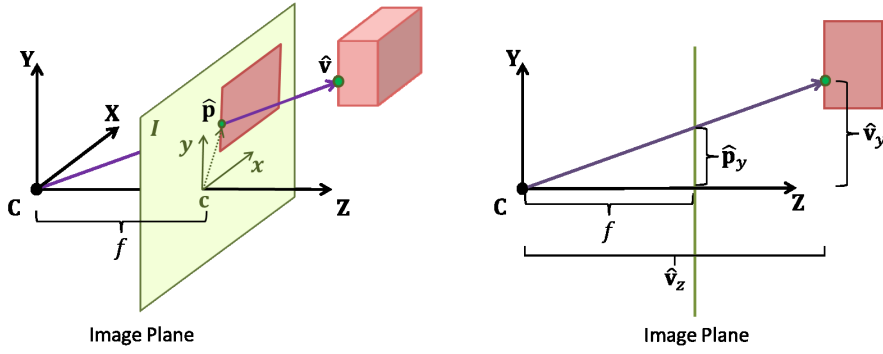


Figure 2.1: Camera model. A 3D point \hat{v} is projected onto the image plane at position \hat{p} with the camera properties, i. e., the focal length f and the principal point $\mathbf{c} = [c_x, c_y]^\top$. Images from Garrido (2017).

$\mathbf{c} = [c_x, c_y]^\top$, represents a linear scaling and translation to map the normalized space into screen space:

$$\mathbf{K} = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.2)$$

In practice, the principal point \mathbf{c} is commonly assumed to lie at the image center, and the focal length f is pre-calibrated [Bradski and Kaehler 2013; Zhang 2000]. In this thesis, homogeneous coordinates [Forsyth and Ponce 2012] are employed for the camera model unless stated otherwise.

2.1.2 Lens Model

The pinhole camera model cannot explain the optical properties of a camera lens. A thin-lens model is introduced to enable additional lens effects in face editing pipelines. We assume a standard video camera with a finite aperture lens that produces a limited depth of field in which the image is sharply focused only in a narrow depth range. According to the thin-lens model, which is illustrated in Figure 2.2, the amount of defocus blur is quantified by the diameter c of the circle of confusion [Potmesil and Chakravarty 1982]:

$$c = \frac{Af|D-F|}{D(F-f)} = \frac{f^2|D-F|}{N_f D(F-f)}, \quad (2.3)$$

where $A = f/N_f$ is the diameter of the aperture, f is the focal length of the lens, N_f is the f -number of the aperture, D is the depth of a scene point and F is the focus distance. We assume that the aperture and focal length are fixed in the input video, and that the focus distance F changes over time. Therefore, the defocus blur of a 3D point only depends on its depth D and the focus distance F , which we express as the point-spread function $\Phi(D, F)$ corresponding to the circle of confusion according to Equation 2.3.

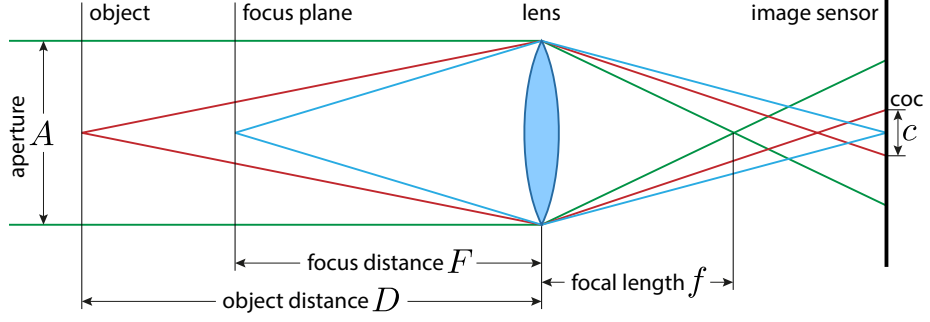


Figure 2.2: The thin-lens model and the circle of confusion c ('coc').

Then, we can model the color of a defocused image \mathbf{V} at a pixel \mathbf{x} by the following convolution:

$$\mathbf{V}(\mathbf{x}) = (\Phi(\mathbf{D}(\mathbf{x}), F) * \mathbf{I})(\mathbf{x}), \quad (2.4)$$

where \mathbf{I} denotes the all-in-focus image. Note that Φ is spatially varying because each pixel \mathbf{x} may have a different depth value $\mathbf{D}(\mathbf{x})$. For brevity, we omit the pixel index \mathbf{x} in the thesis unless stated otherwise.

2.1.3 Image Formation Model

An image formation model determines the color values for each projected 2D coordinate, taking the reflection model with a face surface and a lighting condition into consideration. As commonly used in the literature [Wu et al. 2011; Valgaerts et al. 2012; Garrido et al. 2013; Thies et al. 2015; Garrido et al. 2016], we employ a Lambertian reflection model, i. e., an isotropic diffuse bidirectional reflectance distribution function (BRDF) that reflects radiance equally into all directions, to represent the face surface property. The intensity of the radiance is also proportional to the incident lighting $L(\hat{\mathbf{v}}, \omega) \in \mathbb{R}^3$ at a mesh vertex $\hat{\mathbf{v}}$ from an incoming light direction $\omega \in \mathbb{R}^3$. More formally, this process is described with the rendering equation as follows [Kajiya 1986]:

$$\mathcal{B}(\hat{\mathbf{v}}, \omega) = \mathbf{c}(\hat{\mathbf{v}}) \circ \int_{\Omega} L(\hat{\mathbf{v}}, \omega) V(\hat{\mathbf{v}}) \max(\langle \omega, \hat{\mathbf{n}}(\hat{\mathbf{v}}) \rangle, 0) d\omega, \quad (2.5)$$

where $\mathcal{B}(\hat{\mathbf{v}}, \omega)$ is the irradiance at vertex $\hat{\mathbf{v}}$ from direction ω sampled on the hemisphere Ω . Here, $\mathbf{c}(\hat{\mathbf{v}}) \in \mathbb{R}^3$, $\hat{\mathbf{n}} \in \mathbb{R}^3$ and $V \in \{0, 1\}$ denote the surface albedo, normal and the binary visibility map at vertex $\hat{\mathbf{v}}$ respectively. $\langle \cdot \rangle$ represents the inner product while \circ a point-wise multiplication. The lighting function $L(\omega)$ is often modeled, as in [Wu et al. 2011; Valgaerts et al. 2012], using spherical harmonics (SH) functions:

$$L(\omega) = \sum_{l=0}^{j-1} \sum_{m=-l}^l \theta_l^{[i], m} Y_l^m(\omega) = \sum_{l=1}^{j^2} \theta_l^{[i]} Y_l(\omega), \quad (2.6)$$

where $Y_l^m \in \mathbb{R}$, $\forall l, m$ denote the SH functions with j bands, and l is the index of the band. $\theta^{[i]} =$

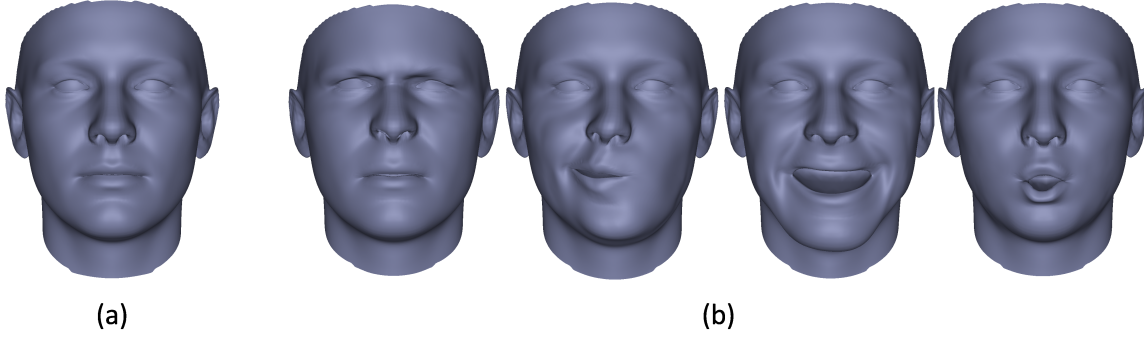


Figure 2.3: Example of a blendshape model. (a) Neutral face. (b) Semantic shapes. *From left to right:* Frown, mouth to the right, smile and “O”-like mouth shape. Images from [Garrido \(2017\)](#).

$[\theta^{[i],r}, \theta^{[i],g}, \theta^{[i],b}]^\top$ balances the effect of the lighting at each color channel. Throughout the thesis, illumination and environment maps are approximated by the SH functions, unless explicitly stated otherwise.

2.2 Facial Geometry

Facial geometry is composed of shape (also referred to as identity) and expression. To efficiently model the variations of each component, it is represented by a linear combination of an average face model and displacement vectors mathematically. We provide backgrounds about parametric face representation and blendshapes that are used to model facial geometry in the thesis.

2.2.1 Parametric Face Representation

We represent facial shapes with a low-dimensional parametric model [[Blanz and Vetter 1999](#)]. In this model, the geometric deformation of a 3D face model is achieved through an affine model $\mathbf{v} \in \mathbb{R}^{3N}$ that stacks the per-vertex deformations of the underlying template mesh with N vertices, as follows:

$$\mathbf{v}(\theta^{[s]}) = \mathbf{a}^{[g]} + \sum_{k=1}^{N_g} \theta_k^{[s]} \mathbf{b}_k^{[s]}. \quad (2.7)$$

Here, $\mathbf{a}^{[g]} \in \mathbb{R}^{3N}$ and $\{\mathbf{b}_k^{[s]}\}_{k=1}^{N_g}$ represent the average facial geometry and the basis that is computed by applying principal component analysis (PCA) to 200 high-quality face scans, respectively. The low-dimensional parametric face model is employed throughout the thesis unless stated otherwise.

2.2.2 Blendshapes

We adopt a blendshape model [[Lewis et al. 2014](#)] in order to describe facial expression, which has been widely used in the 3D facial animation literature due to its flexibility in face editing. Each blendshape is a static face geometry that refers to a semantically meaningful deformation such as blink, smile and

frown. To create in-between facial expression and animation, these deformations are linearly blended with the weights that represent the strength of each deformation. Consequently, the blendshape model provides an intuitive control of facial expression, allowing part-based 3D face editing.

Mathematically, the blendshape model is formulated with additive shape deformations on top of a neutral face geometry, as Figure 2.3 shows. Let $\mathbf{a}_0^{[e]}$ be the neutral face and $\mathbf{B} = \{\mathbf{b}_1^{[e]}, \dots, \mathbf{b}_n^{[e]}\}$ be the set of n blendshapes. Here, $\mathbf{b}_i^{[e]} \in \mathbb{R}^{3k}, \forall i$ represents column vectors of k vertices of 3D face geometry. Facial expression \mathbf{e} is then represented by a linear combination of the neutral face shape and its per-vertex 3D displacements to each blendshape:

$$\mathbf{e} = \mathbf{a}_0^{[e]} + \sum_{i=1}^n \theta_i^{[e]} (\mathbf{b}_i^{[e]} - \mathbf{b}_0^{[e]}) = \mathbf{b}_0^{[e]} + \sum_{i=1}^n \theta_i^{[e]} \mathbf{d}_i^{[e]} = \mathbf{b}_0^{[e]} + \mathbf{B}\boldsymbol{\theta}^{[e]}, \quad (2.8)$$

where $0 \leq \theta_i^{[e]} \leq 1, \forall i = 1 : n$ denote the linear weights. With $\boldsymbol{\theta}^{[e]} = [\theta_1^{[e]}, \dots, \theta_n^{[e]}]^\top \in \mathbb{R}^n$ and $\mathbf{B} = [\mathbf{d}_1^{[e]} | \dots | \mathbf{d}_n^{[e]}] \in \mathbb{R}^{3k \times n}$ that are a stack of the linear weights and per-vertex 3D displacements respectively, the linear combination can be also expressed in a matrix form.

Although it encourages semantics-preserving facial animation, each blendshape is not necessarily orthogonal to each other. Thus, the same facial expression can be found by different linear combinations of the weights and blendshapes. Moreover, some blendshapes should not be combined by certain weights in order to avoid implausible facial expressions. As an example, adding semantically similar expressions doubles the deformation, leading to unrealistic facial expressions or breaking anatomical facial symmetry. To prevent such inconsistent blendshape combinations, additional constraints or a pairwise activation of blendshapes [Lewis et al. 2014] or a sparsity of the blending weights [Bouaziz et al. 2013] can be incorporated to restrict the activation of the blendshapes [Li et al. 2013a; Thies et al. 2015].

Despite the limitations, Equation 2.8 has been widely used in most 3D face modeling approaches in the literature [Bouaziz et al. 2013; Li et al. 2010; Li et al. 2013b; Thies et al. 2015; Weise et al. 2011] and commercial packages such as Blender and Maya to enable facial animation. In the thesis, we also employ the same blendshape model unless stated otherwise.

2.3 Deep Learning

This thesis revisits the face reconstruction and editing in the context of deep learning. Recent advances in the deep learning field, e.g., AlexNet [Krizhevsky et al. 2012] and Generative Adversarial Networks (GANs) [Goodfellow et al. 2014] to name a few, have achieved human-level performance in many challenging tasks such as image classification and generation. This has been made possible with deeper neural networks (NNs) at high capacity, which are implicit functions flexible enough to represent complex models with high accuracy. Especially, convolutional neural networks (CNNs) [LeCun et al. 1995], a kind of NNs specially suited for many computer vision tasks, extract low- to high-level image features in a hierarchical manner to better describe images as they go deeper through the

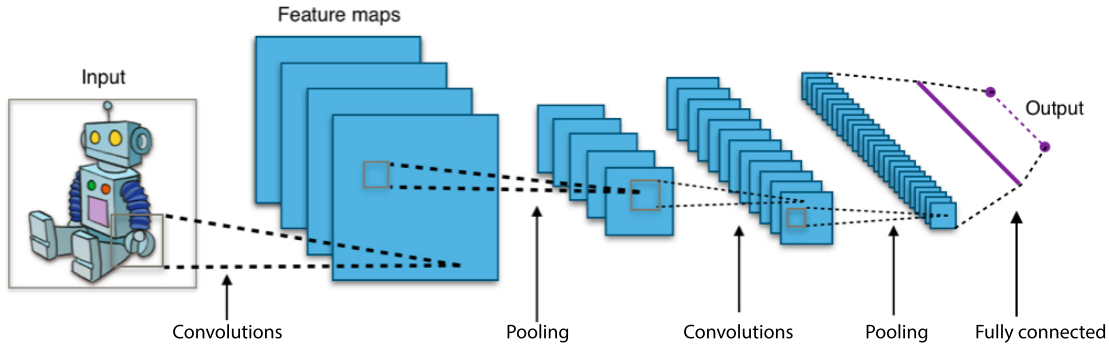


Figure 2.4: Example of a convolutional neural network for image classification. It extracts high-level image features using convolution, nonlinear activation and pooling operations. The feature map at the last convolutional layer is translated to a corresponding class label via fully connected layers. The feature extraction and the label mapping are optimized end-to-end. Images from [Wikipedia](#).

network, as Figure 2.4 shows. When trained end-to-end over a large-scale dataset, CNNs outperform most explicit model-based methods. In the thesis, we adapt deep learning-based methods and combine them with model-based approaches in a new way with the goal of highly realistic face editing.

CNNs are composed by a series of layers: convolution, batch normalization, nonlinear activation, dropout, pooling, fully connection and deconvolution. A general overview of each layer is briefly given in the following. Convolutional layers use kernels (also referred to as filters) to extract low-level image features such as corners and edges from an input, which are then used to build up high-level image representations in the subsequent layers. Mathematically, a kernel performs convolution operation in a sliding window manner over the whole input tensor, providing the feature map (also known as a activation map) as an output r_j^l :

$$r_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l, \quad (2.9)$$

where a_k^{l-1} is the response of the k -th hidden unit at the previous layer $l-1$, and w_{jk}^l and b_j^l are the convolution weight and bias term respectively. In this layer, the size of the kernel as well as a stride – the step of a convolution operation, determine the receptive field over which CNNs aggregate the kernel response. In general, the receptive field becomes wider across subsequent convolutional layers, and this allows deeper CNNs to compute high-level image features with higher level of abstraction. The distribution of feature maps changes during training – the process of minimizing the loss function parameterized by learnable variables of CNNs, leading to internal covariate shift [Ioffe and Szegedy 2015]. To address this issue, batch normalization \hat{r}_j^l is commonly employed:

$$\hat{r}_j^l = \frac{r_j^l - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \quad (2.10)$$

where μ and σ^2 are the mean and variance of the feature response r_j^l across the batch. ε is a small constant to prevent a zero division error. A feature map is fed through an activation layer afterwards.

This layer introduces nonlinearity to the feature map so that CNNs can approximate more complicated functions:

$$a_j^l = \phi(\hat{r}_j^l), \quad (2.11)$$

where $\sigma(\cdot)$ is an element-wise nonlinear function. Among others, most popular choices include rectified linear units (ReLU) [Nair and Hinton 2010], hyperbolic tangent (Tanh) and sigmoid functions. The activation functions are applied element-wise to the feature map, for instance, truncating the individual negative activations with ReLU. When the capacity of CNNs is large in comparison to the complexity of training data, it is prone to statistical overfitting. To avoid this, dropout layers [Srivastava et al. 2014] are often introduced. The idea behind dropout is simple yet effective. Dropout nullifies a subset of activations randomly over training iterations. A dropout layer is not applied at test time. Another layer often introduced between convolutional layers is a pooling layer, also sometimes referred to as a downsampling layer. With max pooling being the most popular, it samples a maximum value over each local neighborhood in an activation map. This provides CNNs with translation invariance as well as less model complexity. Throughout the aforementioned layers, a deep image representation which encodes low- and high-level features is obtained at the last nonlinear activation layer. As a last stage, it is common that fully connected or transposed convolutional layers process the rich image representation to derive class labels or images respectively. Fully connected layers – full linear combinations of all responses between one and another layers, also known as dense layers, are often exploited for classification and regression tasks to find the correlation of the feature maps with labels. In contrast, transposed convolutional layers used in image generation and synthesis tasks reverse the preceded convolution operations to recover an image output back from the response map. The training process of CNNs, i.e., the optimization of a loss function with respect to the kernel and the bias values in the convolutional and fully connected layers, is achieved by backpropagation [LeCun et al. 1989]. For computational efficiency, backpropagation updates the kernel weights by interleaving forward and backward passes. In the backward pass, the weight update is determined by the gradient of a loss function, also considering a learning rate and a momentum term. In the thesis, the aforementioned CNNs and the backpropagation algorithm are employed unless stated otherwise.

Face Reconstruction

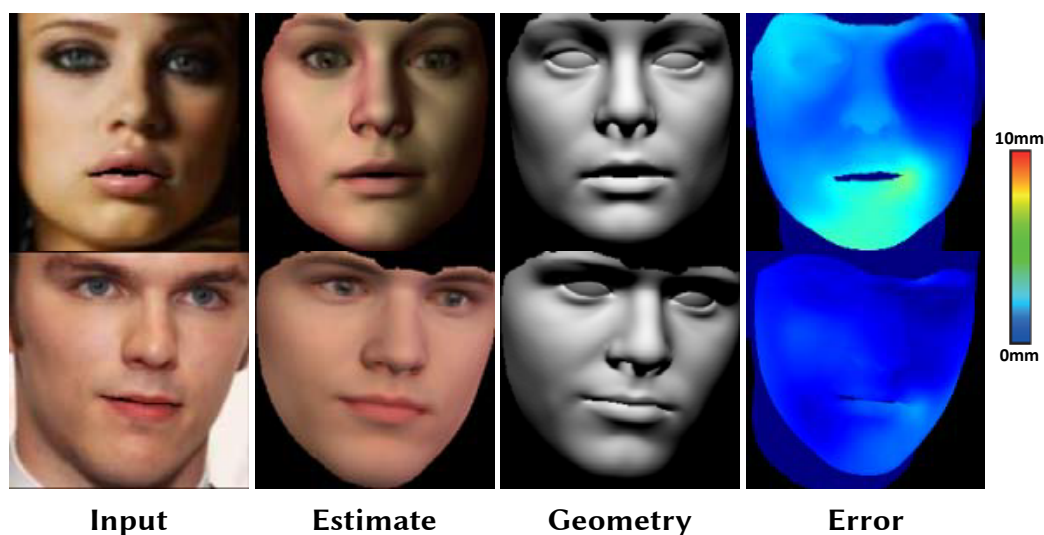


Figure 3.1: The single-shot deep inverse face renderer *InverseFaceNet* obtains a high-quality geometry, reflectance and illumination estimate from just a single input image. *InverseFaceNet* jointly recover the facial pose, shape, expression, reflectance and incident scene illumination. *From left to right:* the input photo, our estimated face model, its geometry, and the pointwise Euclidean geometry error compared to Garrido et al. (2016).

This chapter introduces *InverseFaceNet*, a deep convolutional inverse rendering framework for faces that jointly estimates facial pose, shape, expression, reflectance and illumination from a single input image (see Figure 3.1). By estimating all parameters from just a single image, advanced editing possibilities on a single face image, such as appearance editing and relighting, become feasible in real time. Most previous learning-based face reconstruction approaches do not jointly recover all dimensions, or are severely limited in terms of visual quality. In contrast, we propose to recover high-quality facial pose, shape, expression, reflectance and illumination using a deep neural network that is trained using a large, synthetically created training corpus. Our approach builds on a novel loss function that measures model-space similarity directly in parameter space and significantly improves reconstruction accuracy. We further propose a self-supervised boosting process in the network training loop, which iteratively updates the synthetic training corpus to better reflect the distribution of real-

world imagery. The method and results presented in this chapter are based on [Kim et al. \(2018b\)](#).

3.1 Introduction

Inverse rendering aims to reconstruct scene properties such as geometry, reflectance and illumination from image data. This reconstruction is fundamentally challenging, as it inevitably requires inverting the complex real-world image formation process. It is also an ill-posed problem as certain effects, such as low-frequency reflectance and illumination, can be indistinguishable [[Ramamoorthi and Hanrahan 2001b](#)]. Inverse rendering, for example, enables relighting of faces by modifying the scene illumination and keeping the face reflectance and geometry fixed.

Recently, optimization-based approaches for inverse face rendering were introduced with convincing results [[Garrido et al. 2016](#); [Thies et al. 2016](#); [Aldrian and Smith 2013](#); [Li et al. 2014a](#); [Kemelmacher-Shlizerman and Seitz 2011](#)]. One of the key ingredients that enables to disentangle pose, geometry (both related to shape and facial expression), reflectance and illumination are specific priors that constrain parameters to plausible values and distributions. Formulating such priors accurately for real faces is difficult, as they are unknown a priori. The priors could be learned by applying inverse rendering to a large dataset of real face images, but this is highly challenging without having the priors a priori.

We take a different approach to solve this chicken-and-egg problem. Instead of formulating explicit priors, we directly learn inverse face rendering with a deep neural network that implicitly learns priors based on the training corpus. As annotated training data is hard to come by, we train on synthetic face images with known model parameters (geometry, reflectance and illumination). This is similar to existing approaches [[Richardson et al. 2016](#); [Richardson et al. 2017](#); [Sela et al. 2017](#)], but the used parameter distribution does not match that of real-world faces and environments. As a result, the learned implicit priors are rather weak and do not generalize well to in-the-wild images.

The approach of [Li et al. \(2017b\)](#) introduces a self-augmented procedure for training a CNN to regress the spatially varying surface appearance of planar exemplars. Our self-supervised boosting approach extends their training strategy to handle unknown, varying geometry. In addition, we resample based on a mean-adaptive Gaussian in each boosting step, which helps to populate out-of-domain samples, especially at the domain boundary.

In contrast to many other approaches, InverseFaceNet also regresses color reflectance and illumination. Our main technical contribution is the introduction of a self-supervised boosting step in our training loop, which continuously updates the training corpus to better reflect the distribution of real-world face images. The key idea is to apply the latest version of the inverse face rendering network to real-world images without ground truth, to estimate the corresponding face model parameters, and then to create synthetic face renderings for perturbed, but known, parameter values. In this way, we are able to generate additional synthetic training data that better reflects the real-world distribution of face model parameters, and our network therefore better generalizes to the real-world setting. Our experiments

demonstrate that our approach greatly improves the quality of regressed face models for real face images compared to approaches that are trained exclusively on synthetic data.

The main contribution of the chapter is InverseFaceNet – a real-time, deep, single-shot inverse face rendering network that estimates pose, shape, expression, color reflectance and illumination from just a single input image in a single forward pass, and is multiple orders of magnitude faster than previous optimization-based methods estimating similar models. To improve the accuracy of the results, we further propose a loss function that measures model-space distances directly in a modified parameter space. We further propose self-supervised boosting of a synthetic training corpus based on real images without available ground truth to produce labeled training data that follows the real-world parameter distribution. This leads to significantly improved reconstruction results for in-the-wild face photos.

3.2 Related Work

Inverse Rendering (of Faces) The goal of inverse rendering is to invert the graphics pipeline, i.e., to recover the geometry, reflectance (albedo) and illumination from images or videos of a scene – or, in our case, a face. Early work on inverse rendering made restrictive assumptions like known scene geometry and calibrated input images [Yu et al. 1999; Ramamoorthi and Hanrahan 2001b]. However, recent work has started to relax these assumptions for specific classes of objects such as faces. Deep neural networks have been shown to be able to invert simple graphics pipelines [Nair et al. 2008; Kulka-rni et al. 2015], although these techniques are so far only applicable to low-resolution grayscale images. In contrast, our approach reconstructs full-color facial reflectance and illumination, as well as geometry. Aldrian and Smith (2013) use a 3D morphable model for optimization-based inverse rendering. They sequentially solve for geometry, reflectance and illumination, while we jointly regress all dimensions at once. Thies et al. (2016) recently proposed a real-time inverse rendering approach for faces that estimates a person’s identity and expression using a blendshape model with reflectance texture and colored spherical harmonics illumination. Their approach is designed for reenactment and is visually convincing, but relies on non-linear least-squares optimization, which requires good initialization and a face model calibration step from multiple frames, while our approach estimates a very similar face model in a single shot, from a single in-the-wild image, in a fraction of the time. Inverse rendering has also been applied to face image editing [Lu et al. 2016; Shu et al. 2017], for example to apply makeup [Li et al. 2014a; Li et al. 2015a]. However, these approaches perform an image-based intrinsic decomposition without an explicit 3D face model, as in our case.

Face Models The appearance and geometry of faces are often modeled using 3D morphable models [Banz and Vetter 1999] or active appearance models [Cootes et al. 2001]. These seminal face models are powerful and expressive, and remain useful for many applications even though more complex and accurate appearance models exist [Klehm et al. 2015; Li et al. 2017a]. Recently, a large-scale parametric face model [Booth et al. 2018] was created from 10,000 facial scans, Booth et al. (2017) extend

3D morphable models to “in-the-wild” conditions, and deep appearance models [Duong et al. 2016] extend active appearance models by capturing geometry and appearance of faces more accurately under large unseen variations. We describe the face model we use in Section 3.4.

3D Face Reconstruction The literature on reconstructing face geometry, often with appearance, but without any illumination, is much more extensive compared to inverse rendering. We focus on single-view techniques and do not further discuss multi-view or multi-image approaches [Ichim et al. 2015; Suwajanakorn et al. 2014; Piotraschke and Blanz 2016; Klaudiny et al. 2017; Roth et al. 2017]. Recent techniques approach monocular face reconstruction by fitting active appearance models [Duong et al. 2016; Alabort-i Medina and Zafeiriou 2017], blendshape models [Cao et al. 2013; Garrido et al. 2013; Garrido et al. 2016; Thomas and Taniguchi 2016], affine face models [Shi et al. 2014; Tran et al. 2017; Richardson et al. 2016; Crispell and Bazik 2017; Dou et al. 2017; Schönborn et al. 2017; Guo et al. 2017; Tewari et al. 2017], mesh geometry [Richardson et al. 2017; Laine et al. 2017; Jiang et al. 2017; Roth et al. 2017; Sela et al. 2017], or volumetric geometry [Jackson et al. 2017] to input images or videos. Shading-based surface refinement can extract even fine-scale geometric surface detail [Cao et al. 2015; Garrido et al. 2016; Richardson et al. 2017; Jiang et al. 2017; Roth et al. 2017; Sela et al. 2017]. Many techniques use facial landmark detectors for more robustness to changes in the head pose and expression, and we discuss them in the next section. A range of approaches use RGB-D input [Weise et al. 2011/e.g.; Li et al. 2013b; Thies et al. 2015], and while they achieve impressive face reconstruction results, they rely on depth data which is typically not available for in-the-wild images or videos.

Deep neural networks have recently shown promising results on various face reconstruction tasks. In a paper before its time, Nair et al. (2008) proposed an analysis-by-synthesis algorithm that iteratively explores the parameter space of a black-box generative model, such as active appearance models (AAM) [Cootes et al. 2001], to learn how to invert it, e.g., to convert a photo of a face into an AAM parameter vector. We are inspired by their approach and incorporate a self-supervised boosting approach into our training process (see Section 3.7) to make our technique more robust to unseen inputs, in our case real photographs.

Richardson et al. (2016) use iterative error feedback [Carreira et al. 2016] to optimize the shape parameters of a grayscale morphable model from a single input image. Richardson et al. (2017) build on this to reconstruct detailed depth maps of faces with learned shape-from-shading. Sela et al. (2017) learn depth and correspondence maps directly using image-to-image translation, and follow this with non-rigid template mesh alignment. Dou et al. (2017) regress only the identity and expression components of a face. All these approaches are trained entirely on synthetic data [Blanz and Vetter 1999]. Tran et al. (2017) train using a photo collection, but their focus lies on estimating morphable model parameters to achieve robust face recognition. In contrast to these approaches, ours not only recovers face geometry and texture, but a more complete inverse rendering model that also comprises color reflectance and illumination, from just a single image without the need for iteration. Jackson et al. (2017) directly regress a volumetric face representation from a single input image, but this requires a

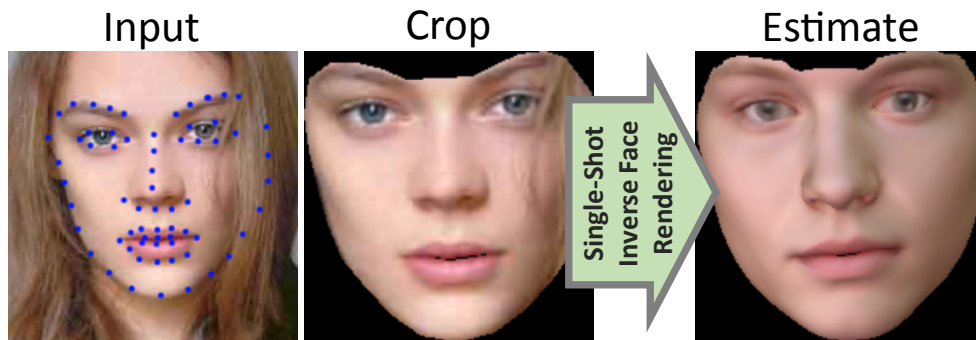


Figure 3.2: Our single-shot inverse face renderer regresses a dense reconstruction of the pose, shape, expression, skin reflectance and incident illumination from a single photograph.

large dataset with matching face images and 3D scans, and does not produce an editable face model, as in our case. Schönborn et al. (2017) optimize a morphable model using Bayesian inference, which is robust and accurate, but very slow compared to our approach (taking minutes rather than milliseconds). Tewari et al. (2017) learn a face regressor in a self-supervised fashion based on a CNN-based encoder and a differentiable expert-designed decoder. Our self-supervised boosting approach combines the advantages of synthetic and real training data, which leads to similar quality reconstructions without the need for a hand-crafted differentiable rendering engine.

Face Alignment Many techniques in 3D face reconstruction, including ours, draw on facial landmark detectors for robustly identifying the location of landmark keypoints in the photograph of a face, such as the outline of the eyes, nose and lips. These landmarks can provide valuable pose-independent initialization. Chrysos et al. (2017) and Jin and Tan (2017) provide two recent surveys on the many landmark detection approaches that have been proposed in the literature. Perhaps unsurprisingly, deep learning approaches [Zhu et al. 2016; Bhagavatula et al. 2017] are again among the best available techniques. However, none of these techniques works perfectly [Steger and Timofte 2016; Bulat and Tzimiropoulos 2017]: facial hair, glasses and poor lighting conditions pose the largest problems. In many cases, these problems can be overcome when looking at video sequences instead of single images [Peng et al. 2016], but this is a different setting to ours.

3.3 Overview

We first detect a set of 66 2D facial landmarks [Saragih et al. 2011a], see Figure 3.2. The landmarks are used to segment the face from the background, and mask out the mouth interior to effectively remove the parts of the image that cannot be explained by our model. The masked face is input to our deep inverse face rendering network (Section 3.6), which is trained on synthetic facial imagery (Section 3.5) using a parametric face and image formation model (Section 3.4). Starting from this low-quality corpus, we apply our self-supervised boosting approach that updates the parameter distribution of the training set (Section 3.7) to generate a training corpus that better approximates the real-world distribution. This leads to reconstructions of higher quality (Section 3.8). Finally, we discuss limitations (Section 3.9)

and summarize (Section 3.10).

3.4 The Space of Facial Imagery

We parameterize face images using $m = 350$ parameters:

$$\theta = (\mathbf{R}, \theta^{[s]}, \theta^{[e]}, \theta^{[r]}, \theta^{[i]}) \in \mathbb{R}^m. \quad (3.1)$$

Here, \mathbf{R} specifies the global rotation (3 parameters), $\theta^{[s]}$ the shape (128), $\theta^{[e]}$ the expression (64), $\theta^{[r]}$ the skin reflectance (128), and $\theta^{[i]}$ the incident illumination (27). Note that we do not include translation as our network works on consistently segmented input images (see Figure 3.2 and Section 3.3).

3.4.1 Affine Face Model

We employ an affine face model to parameterize facial geometry $\mathcal{F}^{[g]} \in \mathbb{R}^{3V}$ and reflectance $\mathcal{F}^{[r]} \in \mathbb{R}^{3V}$, where V is the number of vertices of the underlying manifold template mesh. The geometry vector $\mathcal{F}^{[g]}$ stacks the V 3D coordinates that define the mesh's embedding in space. Similarly, the reflectance vector $\mathcal{F}^{[r]}$ stacks the RGB per-vertex reflectance values. The space of facial geometry is modeled by the shape $\theta^{[s]} \in \mathbb{R}^{N_s}$ and expression $\theta^{[e]} \in \mathbb{R}^{N_e}$ parameters:

$$\mathcal{F}^{[g]}(\theta^{[s]}, \theta^{[e]}) = \mathbf{a}^{[g]} + \sum_{i=1}^{N_s} \mathbf{b}_i^{[s]} \sigma_i^{[s]} \theta_i^{[s]} + \sum_{j=1}^{N_e} \mathbf{b}_j^{[e]} \sigma_j^{[e]} \theta_j^{[e]}. \quad (3.2)$$

The spatial embedding is modeled by a linear combination of orthonormal basis vectors $\mathbf{b}_i^{[s]}$ and $\mathbf{b}_j^{[e]}$, which span the shape and expression space, respectively. $\mathbf{a}^{[g]} \in \mathbb{R}^{3V}$ is the average geometry of a neutral expression, the $\sigma_i^{[s]}$ are the shape standard deviations and the $\sigma_j^{[e]}$ are the standard deviations of the expression dimensions.

Per-vertex reflectance is modeled similarly using a small number of reflectance parameters $\theta^{[r]} \in \mathbb{R}^{N_r}$:

$$\mathcal{F}^{[r]}(\theta^{[r]}) = \mathbf{a}^{[r]} + \sum_{i=1}^{N_r} \mathbf{b}_i^{[r]} \sigma_i^{[r]} \theta_i^{[r]}. \quad (3.3)$$

Here, $\mathbf{b}_i^{[r]}$ are the reflectance basis vectors, $\mathbf{a}^{[r]}$ is the average reflectance and the $\sigma_i^{[r]}$ are the standard deviations.

The face model is computed from 200 high-quality 3D scans [Blanz and Vetter 1999] of Caucasians (100 male and 100 female) using PCA. We use the $N_s = N_r = 128$ most significant principal directions to span our face space. The used expression basis is a combination of the Digital Emily model [Alexander et al. 2010] and FaceWarehouse [Cao et al. 2014b] (see Thies et al. (2016) for details). We use PCA to compress the over-complete blendshapes (76 vectors) to a subspace of $N_e = 64$ dimensions.

3.4.2 Image Formation

We assume the face to be *Lambertian*, illumination to be distant and smoothly varying, and there is no self-shadowing. We thus represent the incident illumination on the face using second-order spherical harmonics (SH) [Müller 1966; Ramamoorthi and Hanrahan 2001b]. Therefore, the irradiance at a surface point with normal \mathbf{n} is given by

$$\mathcal{B}(\mathbf{n} | \theta^{[i]}) = \sum_{k=1}^{b^2} \theta_k^{[i]} Y_k(\mathbf{n}), \quad (3.4)$$

where Y_k are the $b^2 = 3^2 = 9$ SH basis functions, and the $\theta_k^{[i]}$ are the corresponding illumination coefficients. Since we consider colored illumination, the parameters $\theta_k^{[i]} \in \mathbb{R}^3$ specify RGB colors, leading to $3 \cdot 9 = 27$ parameters in total.

We render facial images based on the SH illumination using a full perspective camera model $\Pi: \mathbb{R}^3 \rightarrow \mathbb{R}^2$. We render the face using a mask (painted once in a preprocessing step) that ensures that the rendered facial region matches the crops produced by the 66 detected landmark locations (see Figure 3.2). The global rotation of the face is modeled with three Euler angles using $\mathbf{R} = \text{Rot}_{xyz}(\alpha, \beta, \gamma)$ that successively rotate around the x -axis (up, α), y -axis (right, β), and z -axis (front, γ) of the camera-space coordinate system.

3.5 Initial Synthetic Training Corpus

Training our deep inverse face rendering network requires ground-truth training data $\{\mathbf{I}_i, \theta_i\}_{i=1}^N$ in the form of corresponding pairs of image \mathbf{I}_i and model parameters θ_i . However, training on real images is challenging, since the ground-truth parameters cannot easily be obtained for a large dataset. We therefore train our network based on synthetically rendered data, where exact ground-truth labels are available.

We sample $N = 200,000$ parameter vectors θ_i and use the model described in Section 3.4 to generate the corresponding images \mathbf{I}_i . Data generation can be interpreted as sampling from a probability $P(\theta)$ that models the distribution of real-world imagery. However, sampling from this distribution is in general difficult and non-trivial. We therefore assume statistical independence between the components of θ , i.e.,

$$P(\theta) = P(\mathbf{R})P(\theta^{[s]})P(\theta^{[e]})P(\theta^{[r]})P(\theta^{[i]}). \quad (3.5)$$

This enables us to efficiently generate a parameter vector θ by independently sampling each subset of parameters.

We uniformly sample the yaw and pitch rotation angles $\alpha, \beta \sim \mathcal{U}(-40^\circ, 40^\circ)$ and the roll angle $\gamma \sim \mathcal{U}(-15^\circ, 15^\circ)$ to reflect common head rotations. We sample shape and reflectance parameters from the Gaussian distributions provided by the parametric PCA face model [Blanz and Vetter 1999]. Since we already scale with the appropriate standard deviations during face generation (see Equations 3.2

and 3.3), we sample both from a standard normal distribution, i.e., $\theta^{[s]}, \theta^{[r]} \sim \mathcal{N}(0, 1)$. The expression basis is based on artist-created blendshapes that only approximate the real-world distribution of the space of human expressions; this will be addressed by the self-supervised boosting presented in Section 3.7. We thus uniformly sample the expression parameters using $\theta^{[e]} \sim \mathcal{U}(-12, 12)$. To prevent closing the mouth beyond anatomical limits, we apply a bias of 4.8 to the distribution of the first parameter¹. Finally, we sample the illumination parameters using $\theta^{[i]} \sim \mathcal{U}(-0.2, 0.2)$, except for the constant coefficient $\theta_1^{[i]} \sim \mathcal{U}(0.6, 1.2)$ to account for the average image brightness, and set all RGB components to the same value. The self-supervised boosting step presented in Section 3.7 automatically introduces colored illumination.

3.6 InverseFaceNet

Given the training data $\{\mathbf{I}_i, \theta_i\}_{i=1}^N$ consisting of N images \mathbf{I}_i and the corresponding ground-truth parameters θ_i , we train a deep inverse face rendering network \mathcal{F} to invert image formation. In the following, we provide details on our network architecture and the employed loss function.

3.6.1 Network Architecture

We have tested several different networks based on the popular AlexNet [Krizhevsky et al. 2012] and ResNet [He et al. 2016] architectures, both pre-trained on ImageNet [Russakovsky et al. 2015]. In both cases, we resize the last fully-connected layer to match the dimensionality of our model (350 outputs), and initialize biases with 0, and weights $\sim \mathcal{N}(0, 0.01)$. These minimally modified networks provide the baseline we build on. We propose more substantial changes to the training procedure by introducing a novel model-space loss in Section 3.6.2, which more effectively trains the same network architecture. The color channels of the input images are normalized to the range $[-0.5, 0.5]$ before feeding the data to the network. We show a comparison between the results of AlexNet and ResNet-101 in Section 3.8.1, and thus choose AlexNet for our results.

Input Pre-Processing The input to our network is a color image of a masked face with a resolution of 240×240 pixels (see Figure 3.2). We mask the face to remove any background and the mouth interior, which cannot be explained by our face model. For this, we use detected landmarks [Saragih et al. 2011a] and resize their bounding box uniformly to fit inside 240×240 pixels, to approximately achieve scale and translation invariance.

Training We train all our inverse face rendering networks using the Caffe deep learning framework [Jia et al. 2014] with stochastic gradient descent based on AdaDelta [Zeiler 2012]. We perform 75K

¹The first parameter mainly corresponds to mouth opening and closing.

batch iterations with a batch size of 32 for training our baseline approaches. To prevent overfitting, we use an ℓ_2 -regularizer (*aka* weight decay) of 0.001. We train with a base learning rate of 0.01.

3.6.2 Model-Space Parameter Loss

We use a weighted norm to define a model-space loss between the predicted parameters θ and ground-truth θ_g by taking the statistics of the face model into account:

$$\mathcal{L}(\theta, \theta_g) = \|\theta - \theta_g\|_{\mathbf{A}}^2 \quad (3.6)$$

$$= (\theta - \theta_g)^\top \underbrace{\mathbf{A}}_{\Sigma^\top \Sigma} (\theta - \theta_g). \quad (3.7)$$

Here, Σ is a weight matrix that incorporates the standard deviations σ^\bullet of the different parameter dimensions:

$$\Sigma = \text{diag}(\omega_{\mathbf{R}} \mathbf{1}_3, \omega_s \sigma^{[s]}, \omega_e \sigma^{[e]}, \omega_r \sigma^{[r]}, \omega_i \mathbf{1}_{27}) \in \mathbb{R}^{m \times m}. \quad (3.8)$$

The coefficients ω_\bullet balance the global importance of the different groups of parameters, and $\mathbf{1}_k$ is a k -dimensional vector of ones. We use the same values $(\omega_{\mathbf{R}}, \omega_s, \omega_e, \omega_r, \omega_i) = (400, 50, 50, 100, 20)$ for all our results. Note that we do not scale the rotation and illumination dimensions individually. Intuitively speaking, our model-space loss enforces that the first PCA coefficients (higher variation basis vectors) should match the ground truth more accurately than the later coefficients (lower-variation basis vectors), since the former have a larger contribution to the final 3D geometry and skin reflectance of the reconstructed face in model space (see Equations 3.2 and 3.3). As shown in Section 3.8, this leads to more accurate reconstruction results. The difference to [Zhu et al. \(2016\)](#) is the computation of the weights, which leads to a statistically meaningful metric.

3.7 Self-Supervised Boosting

The real-world distribution of the model parameters θ is in general unknown for in-the-wild images \mathbf{I}_{real} . Until now, we have sampled from a manually prescribed probability distribution $P(\theta)$, which does not exactly represent the real-world distribution. The goal of the self-supervised boosting step is to make the training data distribution better match the real-world distribution of a corpus \mathcal{R} of in-the-wild face photographs. To this end, we automatically update the parameters for the training corpus. Note that this step is unsupervised and does not require the ground-truth parameters for images in \mathcal{R} to be available.

3.7.1 Boosting

Boosting based on uniform resampling with replacement $\mathbf{I}_r \sim P(\mathbf{I}) = 1/N$ cannot solve the problem of mismatched distributions. Hence, we propose a domain-adaptive approach that resamples new proposals from a mean-adaptive Gaussian distribution based on real images:

$$P(\mathbf{I}_r(\theta) | \mathbf{I}_{\text{real}}) \sim \theta(\mathbf{I}_{\text{real}}) + \mathcal{N}(\mathbf{0}, \sigma^2), \quad (3.9)$$



Figure 3.3: Our approach updates the initial training corpus (left) based on real-world images without available ground truth (right) using a self-supervised boosting approach. The generated new training corpus (middle) better matches the real-world face distribution.

Algorithm 1 Self-Supervised Boosting

- 1: $\mathcal{F} \leftarrow \text{train_network_on_synthetic_faces}();$
 - 2: $\mathcal{R} \leftarrow \text{corpus_of_real_images}();$
 - 3: **for** (number of boosting steps N_{boot}) **do**
 - 4: $\theta_r \leftarrow \text{inverse_rendering}(\mathcal{R}, \mathcal{F});$ ▷ (step 1)
 - 5: $\theta'_r \leftarrow \text{resample_parameters}(\theta_r);$ ▷ (step 2)
 - 6: $\mathcal{R}' \leftarrow \{\text{generate_images}(\theta'_r, \theta'_r)\};$ ▷ (step 3)
 - 7: $\mathcal{F} \leftarrow \text{continue_training}(\mathcal{F}, \mathcal{R}');$ ▷ (step 4)
 - 8: **end for**
-

where $\mathbf{I}_r(\theta)$ is the deterministic rendering process, we compute the inverse of the rendering process $\theta(\mathbf{I}_{\text{real}})$ using InverseFaceNet, and $\mathcal{N}(\cdot)$ is a noise distribution. This shifts the distribution closer to the target distribution of real images \mathbf{I}_{real} . Moreover, adding a non-zero variance $\sigma^2 > \mathbf{0}$ populates out-of-domain samples especially at the domain boundary. Our approach takes the network of the last boosting iteration as final output, instead of averaging the intermediate networks. This prevents from being biased to the manually prescribed sampling distribution of earlier training stages.

3.7.2 Algorithm

Our self-supervised parameter boosting is a four-step process (see Algorithm 1). It starts with a deep neural network \mathcal{F} initially trained on a synthetic training corpus (see Section 3.5) for 15K batch iterations. This guarantees a suitable initialization for all weights in the network. Given a set of images from the corpus of real-world images \mathcal{R} , we first obtain an estimate of the corresponding model parameters θ_r , i.e., $\theta(\mathbf{I}_{\text{real}})$ in Equation 3.9, using the synthetically trained network (step 1). These reconstructed parameters are used to seed the boosting. In step 2, we apply small perturbations to the reconstructed parameters based on the noise distribution $\mathcal{N}(\mathbf{0}, \sigma^2)$. This generates new data around the seed points in model space, and allows the network to slowly adapt to the real-world parameter distribution. We use the following to resample the pose, shape, expression, reflectance and illumination parameters, generating two perturbed parameter vectors for each reconstruction: $\alpha, \beta, \gamma: \mathcal{U}(-5^\circ, 5^\circ)$, $\theta^{[s]}: \mathcal{N}(0, 0.05)$, $\theta^{[r]}: \mathcal{N}(0, 0.2)$, $\theta^{[e]}: \mathcal{N}(0, 0.1)$, and $\theta^{[i]}: \mathcal{N}(0, 0.02)$. In step 3, we generate new synthetic training images \mathbf{I}_r based on the resampled parameters θ'_r , i.e., $\theta(\mathbf{I}_{\text{real}}) + \mathcal{N}(\mathbf{0}, \sigma^2)$. The result is a new synthetic training set \mathcal{R}' that better reflects the real-world distribution of model parameters.

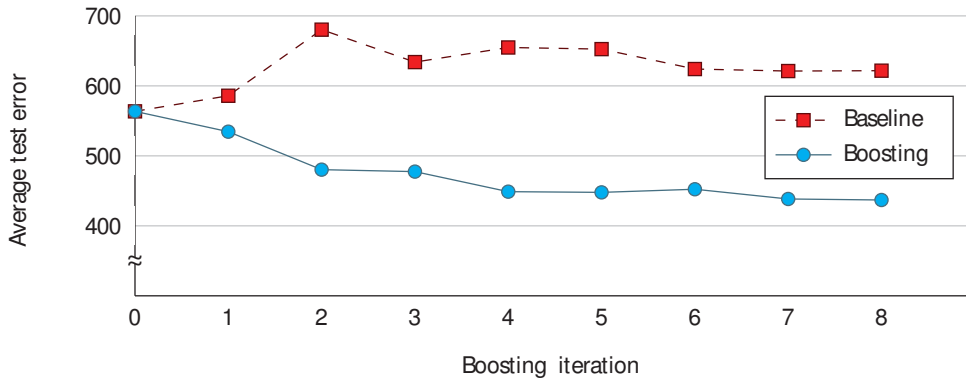


Figure 3.4: Model-space parameter loss (Equation 3.7) for the baseline and boosting approaches on a synthetic test corpus with higher parameter variation than the used training corpus. While our domain-adaptive boosting approach, based on a high-variation training corpus without available ground truth, continuously decreases in loss, the baseline network fails to generalize.

Finally, the network \mathcal{F} is fine-tuned for $N_{\text{iter}} = 7.5\text{K}$ batch iterations on the new training corpus (step 4). In total, we repeat this process for $N_{\text{boot}} = 8$ self-supervised boosting steps.

Over the iterations, the data distribution of the training corpus adapts and better reflects the real-world distribution of the provided in-the-wild facial imagery, as illustrated in Figure 3.3. We also evaluate the parameter loss throughout boosting iterations in Figure 3.4, and observe a clear reduction with our self-supervised boosting. This leads to higher quality results at test time, as shown in Section 3.8. The variance σ^2 could be adaptively scaled based on the photometric error of estimates. However, we found empirically that our framework works well with a fixed variance.

3.8 Experiments and Results

We evaluate our InverseFaceNet on several publicly available datasets. We validate our design choices regarding network architecture, model-space loss, and self-supervised boosting. We then show quantitative and qualitative results and comparisons on the datasets *LFW* (Labeled Faces in the Wild) [Huang et al. 2007], *300-VW* (300 Videos in the Wild) [Shen et al. 2015], *CelebA* [Liu et al. 2015], *FaceWarehouse* [Cao et al. 2014b], *Volker* [Valgaerts et al. 2012] and *Thomas* [Garrido et al. 2013]. For more results, we refer to our supplemental document and video at the project website².

Error Measures We compute the **photometric error** using the RMSE of RGB pixel values (within the mask of the input image) between the input image and a rendering of the reconstructed face model. An error of 0 is a perfect color match, and 255 is the difference between black and white (i.e. lower is better). The **geometric error** measures the RMSE in mm between corresponding vertices in our reconstruction and the ground-truth geometry. We quantify the image-space overlap of the estimated face model and the input face image using the **intersection over union** (IOU) of face masks (e.g. see

²Project page: <http://gvv.mpi-inf.mpg.de/projects/InverseFaceNet>

Table 3.1: Quantitative architecture comparison, model-space parameter loss and our boosting step on 5,914 test images from *CelebA* [Liu et al. 2015]. The best values for each column are highlighted in bold. Training time includes all steps except the initial training data generation. Test times are averaged over 5K images. Training on a GTX Titan and testing on a Titan Xp. Errors show means and standard deviations. * For boosting, we first train 15K iterations on normal synthetic face images (see Section 3.5), and then update for 60K iterations (see Section 3.7). InverseFaceNet (AlexNet [Krizhevsky et al. 2012] with model-space loss and boosting) produces the best geometric error and intersection over union.

Approach	Training iterations	Training time [h]	Test time [ms/image]	Photometric error [8 bits]	Geometric error [mm]	Intersection over union [%]
AlexNet [Krizhevsky et al. 2012]	75K	4.14	3.9	46.26 ± 12.42	2.91 ± 0.99	90.44 ± 3.81
+ model-space loss	75K	4.36	3.9	39.71 ± 9.86	2.77 ± 1.00	92.51 ± 2.59
+ boosting (= InverseFaceNet)	75K*	29.40	3.9	34.03 ± 7.56	2.11 ± 0.84	93.96 ± 2.08
ResNet-101 [He et al. 2016] + model-space loss	150K	40.99	21.0	41.23 ± 10.58	2.54 ± 0.87	92.07 ± 2.87
MoFA [Tewari et al. 2017]	–	–	3.9	17.23 ± 4.42	3.94 ± 1.34	84.20 ± 4.23

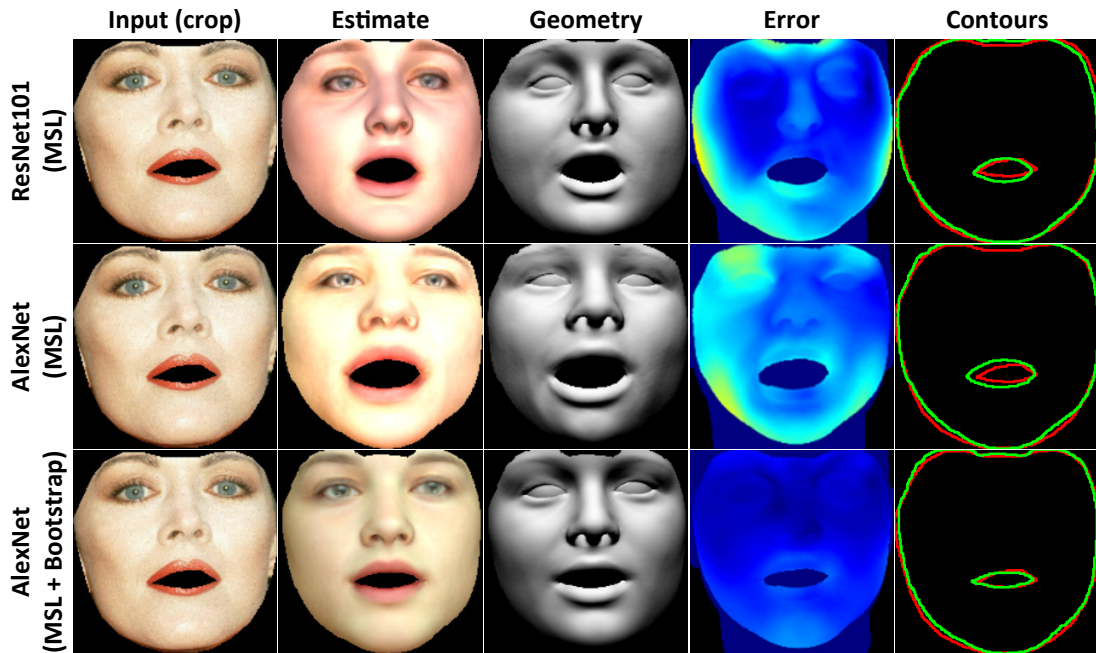


Figure 3.5: Qualitative comparison of ResNet-101 [He et al. 2016] and AlexNet [Krizhevsky et al. 2012] applied to inverse face rendering, both with model-space loss (MSL): ResNet-101 produces lower geometric error (see heatmap) while AlexNet has lower photometric error (also on average, see Table 3.1). AlexNet with MSL and boosting clearly improves the reconstruction of reflectance and geometry, in all error categories.

‘contours’ in Figure 3.5). An IOU of 0% means no overlap, and 100% means perfect overlap (i.e. higher is better).

3.8.1 Evaluation of Design Choices

Table 3.1 evaluates different design choices on a test dataset of 5,914 images (one shown in Figure 3.5) from *CelebA* [Liu et al. 2015] using the error measures described earlier (using our implementation of Garrido et al. (2016) as ground-truth geometry, up to blendshape level).

Network Architecture We first compare the results of the AlexNet [Krizhevsky et al. 2012] and ResNet-101 [He et al. 2016] architectures, both with our model-space loss (see Section 3.6). Reconstructions using ResNet-101 have smaller geometric errors, but worse photometric error and IOU than AlexNet, which is exemplified by Figure 3.5. ResNet-101 is significantly deeper than AlexNet, so training takes about $10\times$ longer and testing about $5\times$ longer. We thus use AlexNet for our inverse face rendering network, which only requires 3.9 ms for the forward pass (on an Nvidia Titan Xp). Landmark detection takes 4.5 ms and face morphing 1 ms (on the GPU). In total, our approach requires 9.4 ms.

Importance of Model-Space Loss Table 3.1 shows that our model-space loss improves on baseline AlexNet [Krizhevsky et al. 2012] in all error categories, particularly the photometric error and IOU. As our model-space loss does not modify the network architecture, the time for the forward pass remains

Table 3.2: Quantitative evaluation of the geometric accuracy on 180 meshes of the FaceWarehouse [Cao et al. 2014b] dataset.

	Our approach		Other approaches		
	Boosting	Baseline	Garrido et al. (2016)	Tewari et al. (2017)	MonoFit (<i>see text</i>)
Error	2.11 mm	2.33 mm	1.59 mm	2.19 mm	2.71 mm
SD	0.46 mm	0.47 mm	0.30 mm	0.54 mm	0.52 mm

the same fast 3.9 ms as before.

Importance of Self-supervised Boosting Our self-supervised boosting (see Section 3.7) significantly improves the reconstruction quality and produces the lowest errors in all categories, as shown in Table 3.1. This can also be seen in Figure 3.5, which shows plausible reconstruction of appearance and geometry, the lowest geometric errors, and improved contour overlap for our network with boosting. Note that the training time for self-supervised boosting includes all steps (see Algorithm 1), in particular reconstructing 100K face models (0.25 h), rendering 200K synthetic faces (2.8 h) and training for 7.5K iterations (0.5 h) for each of the 8 boosting iterations (on an Nvidia GeForce GTX Titan). AlexNet with boosting significantly outperforms ResNet-101 without boosting in reconstruction quality, training time and test time. Note that our approach is better than Tewari et al. (2017) in terms of geometry and overlap, and worse in terms of the photometric error on this test set.

3.8.2 Quantitative Evaluation

We compare the geometric accuracy of our approach to state-of-the-art monocular reconstruction techniques in Figure 3.6. As ground truth, we use the high-quality stereo reconstructions of Valgaerts et al. (2012). Compared to Thies et al. (2016), our approach obtains similar quality results, but without the need for explicit optimization. Therefore, our approach is two orders of magnitude faster (9.4 ms vs 600 ms) than optimization-based approaches. Note that while Thies et al. (2016) run in real time for face tracking, it requires significantly longer to estimate all model parameters from an initialization based on the average model. In contrast to the state-of-the-art learning-based methods by Richardson et al. (2016); Richardson et al. (2017), Jackson et al. (2017) and Tran et al. (2017), ours obtains a reconstruction of all dimensions, including pose, shape, expression, and colored skin reflectance and illumination.

In addition, we performed a large quantitative ground-truth comparison on the FaceWarehouse [Cao et al. 2014b] dataset, see Table 3.2. We show the mean error (in mm) and standard deviation (SD) for 180 meshes (9 different identities, each with 20 different expressions). As can be seen, our boosting approach increases accuracy. Our approach is only slightly worse than the optimization-based approach of Garrido et al. (2016), while being orders of magnitude faster. Boosting is on par with the weakly supervised approach of Tewari et al. (2017), which is trained on real images and landmarks. We also compare to a baseline network ‘MonoFit’ that has been directly trained on the monocular fits of Garrido et al. (2016) on the *CelebA* [Liu et al. 2015] dataset. Our self-supervised boosting approach obtains higher accuracy results.

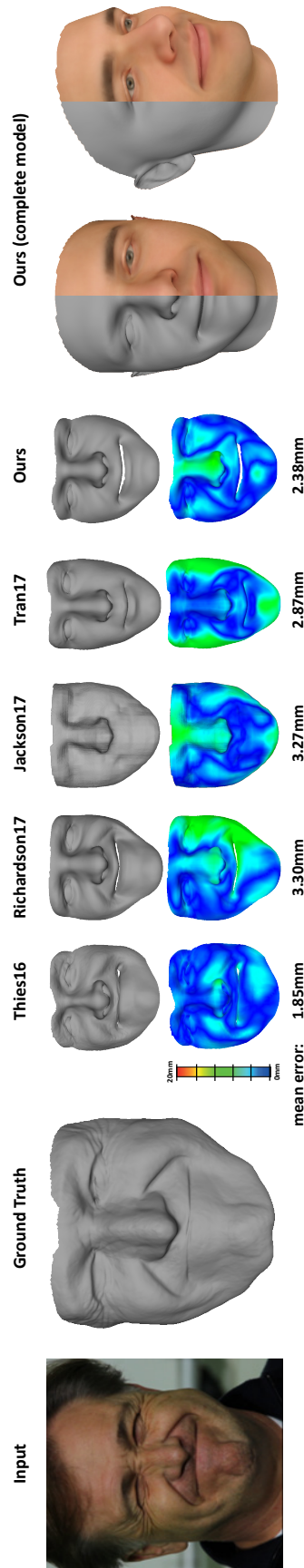


Figure 3.6: Quantitative comparison of geometric accuracy compared to Thies et al. (2016), Richardson et al. (2017), Jackson et al. (2017) and Tran et al. (2017) on Volker [Valgaerts et al. 2012]. The heat maps visualize the pointwise Hausdorff distance (in mm) between the input and the ground-truth. The ground-truth has been obtained by the high-quality binocular reconstruction approach of Valgaerts et al. (2012).



Figure 3.7: Qualitative comparison to optimization-based approaches [Garrido et al. 2013; Garrido et al. 2016] on *Thomas* [Garrido et al. 2013]. For more, see our supplemental document at the project website.

3.8.3 Qualitative Evaluation

We next compare our reconstruction results qualitatively to current state-of-the-art approaches. Figure 3.7 compares our reconstruction to optimization-based approaches that fit a parametric face model [Garrido et al. 2016] or a person-specific template mesh [Garrido et al. 2013]. Our learning-based approach is significantly faster (9.4 ms vs about 2 minutes [Garrido et al. 2016]), and orthogonal to optimization-based approaches, since it can be used to provide a good initial solution.

In Figure 3.8, we also compare to the state-of-the-art deep-learning-based approaches by Richardson et al. (2016); Richardson et al. (2017), Sela et al. (2017), Jackson et al. (2017), Tran et al. (2017) and Tewari et al. (2017). We obtain high-quality results in 9.4 ms. Most of the other approaches are slower, do *not* estimate colored skin reflectance and illumination [Richardson et al. 2016; Richardson et al. 2017; Sela et al. 2017; Jackson et al. 2017], do *not* regress the facial expressions [Tran et al. 2017], or suffer from geometric shrinking artifacts [Tewari et al. 2017]. Note, we compare to Richardson et al.’s ‘CoarseNet’ [Richardson et al. 2017], which corresponds to their earlier method [Richardson et al. 2016], and estimates pose, shape and expression, followed by a model-based optimization of monochrome reflectance and illumination. We also compare to Sela et al.’s aligned template mesh. We don’t compare to ‘FineNet’ [Richardson et al. 2017] or ‘fine detail reconstruction’ [Sela et al. 2017] as these estimate a refined depth map/mesh, and we are interested in comparing the reconstructed parametric face models.

Figure 3.9 shows several monocular reconstruction results obtained with our InverseFaceNet. As can be seen, our approach obtains good estimates of all model parameters.

3.9 Limitations

In this chapter, we propose a solution to the highly challenging problem of inverse face rendering from a single image. Similar to previous learning-based approaches, ours has a few limitations. Our approach does not perfectly generalize to inputs that are outside of the training corpus. Profile views of the head are problematic and hard to reconstruct, even if they are part of the training corpus. Note that even state-of-the-art landmark trackers often fail in this scenario. Handling these cases robustly

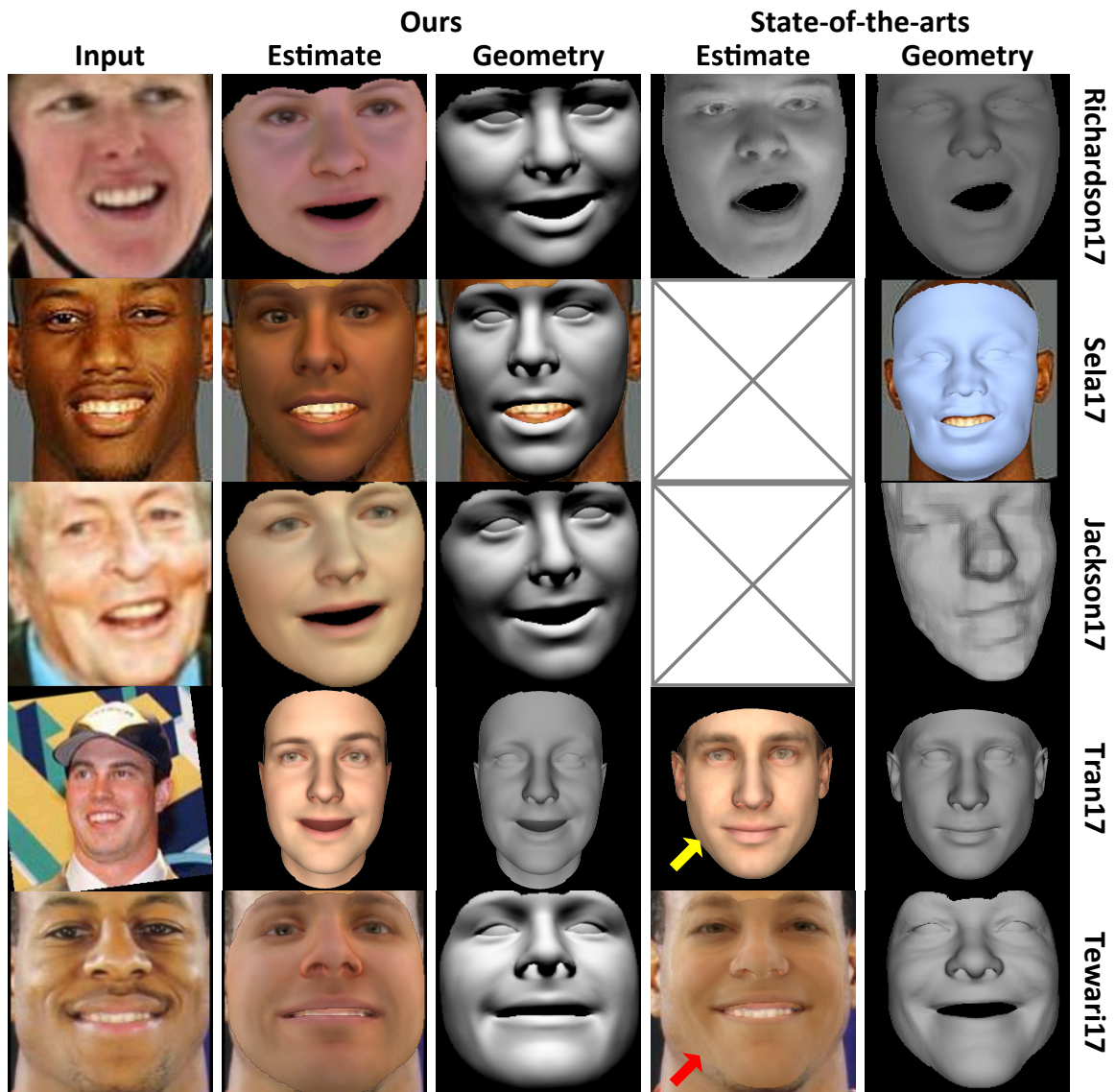


Figure 3.8: Comparison to a wide range of state-of-the-art learning-based approaches. From top to bottom: Comparison to Richardson et al. (2017), Sela et al. (2017), Jackson et al. (2017), Tran et al. (2017) and Tewari et al. (2017). We obtain high-quality results in 9.4 ms. Most other approaches are significantly slower, do not estimate colored skin reflectance and illumination (empty box), do not regress facial expressions (yellow arrow), or suffer from geometric shrinking (red arrow). Images from LFW [Huang et al. 2007], 300-VW [Shen et al. 2015], CelebA [Liu et al. 2015] and FaceWarehouse [Cao et al. 2014b]. For more results, see our supplemental document at the project website.

remains an open research question. Incorrect landmark localization might produce inconsistent input to our network, which harms the quality of the regressed face model. This could be addressed by more sophisticated face detection algorithms, or by joint learning of landmarks and reconstruction. Occlusions of the face, such as hair, beards, sun glasses or hands, can also be problematic. To handle these situations robustly, our approach could be trained in an occlusion-aware manner by augmenting our training corpus with artificial occlusions, similar to Zhao et al. (2018).

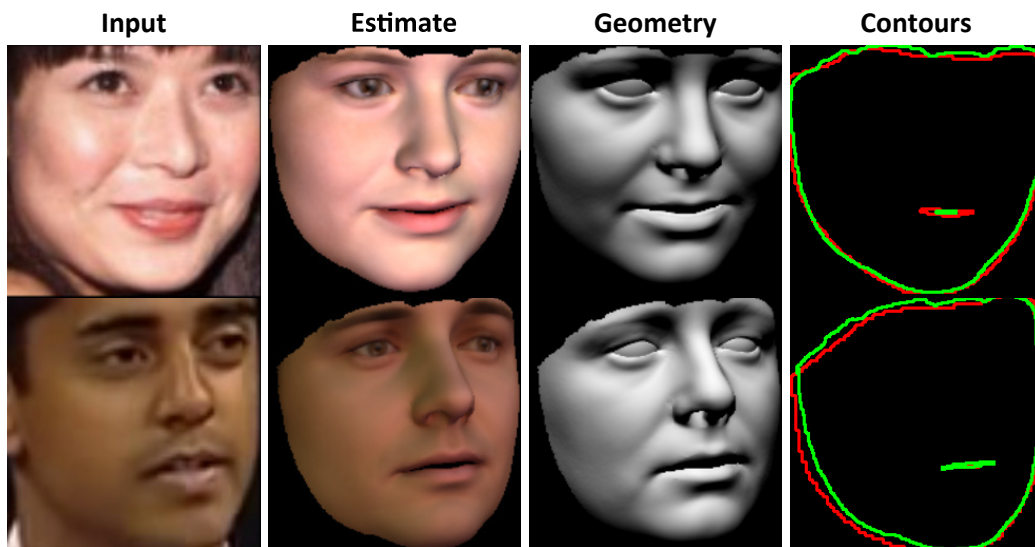


Figure 3.9: Qualitative results on *LFW* [Huang et al. 2007] and *300-VW* [Shen et al. 2015]. Top to bottom: input image, our estimated face model and geometry, and contours (red: input mask, green: ours). Our approach achieves high-quality reconstructions from just a single input image. For more results, we refer to the supplemental document at the project website.

3.10 Summary

In this chapter, we have presented InverseFaceNet – a single-shot inverse face rendering framework. Our key contribution is to overcome the lack of well-annotated image datasets by self-supervised boosting of a synthetic training corpus that captures the real-world distribution. This enables high-quality face reconstruction from just a single monocular image. Our evaluation shows that our approach compares favorably to the state-of-the-art. InverseFaceNet could be used to quickly and robustly initialize optimization-based reconstruction approaches close to the global minimum.

In the next chapter, we show how monocular face reconstruction can be combined with neural face rendering. The reconstructed facial rendering parameters will provide the basis for a rendering-to-video translation network that transforms coarse face model renderings into realistic portrait video outputs for a wide range of applications such as an interactive full head reenactment, visual dubbing and video teleconferencing.

Chapter 4

Face Editing

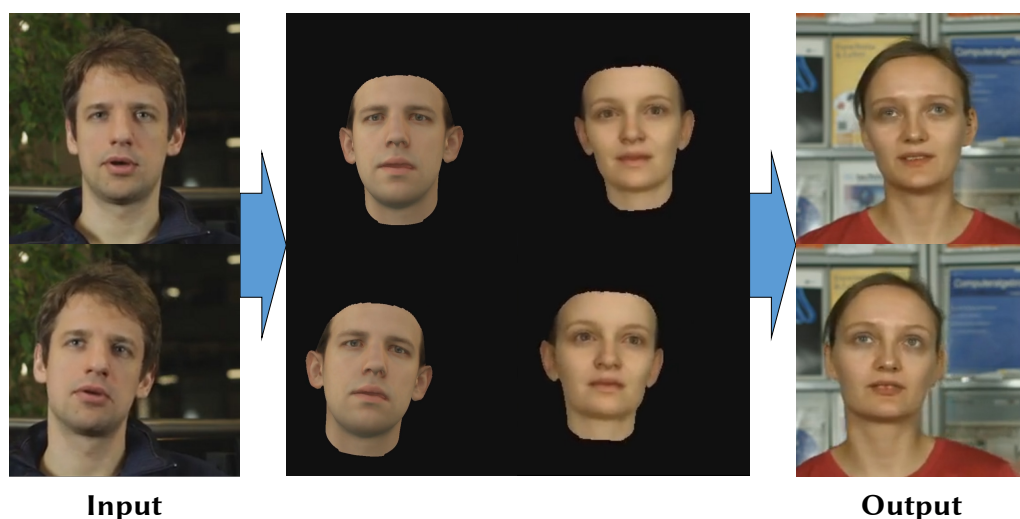


Figure 4.1: Unlike current face reenactment approaches that only modify the expression of a target actor in a video, our novel deep video portrait approach enables full control over the target by transferring the rigid head pose, facial expression and eye motion with a high level of photorealism.

This chapter presents a novel approach that enables photo-realistic re-animation of portrait videos using only an input video. In contrast to existing approaches that are restricted to manipulations of facial expressions only, we are the first to transfer the full 3D head position, head rotation, face expression, eye gaze, and eye blinking from a source actor to a portrait video of a target actor. The core of our approach is a generative neural network with a novel space-time architecture. The network takes as input synthetic renderings of a parametric face model, based on which it predicts photo-realistic video frames for a given target actor. The realism in this rendering-to-video transfer is achieved by careful adversarial training, and as a result, we can create modified target videos that mimic the behavior of the synthetically-created input. In order to enable source-to-target video re-animation, we render a synthetic target video with the reconstructed head animation parameters from a source video, and feed it into the trained network – thus taking full control of the target. With the ability to freely recombine source and target parameters, we are able to demonstrate a large variety of video rewrite applications

without explicitly modeling hair, body or background (see Figure 4.1). For instance, we can reenact the full head using interactive user-controlled editing, and realize high-fidelity visual dubbing. The method and results presented in this chapter are based on Kim et al. (2018a).

4.1 Introduction

Synthesizing and editing video portraits, i.e., videos framed to show a person’s head and upper body, is an important problem in computer graphics, with applications in video editing and movie postproduction, visual effects, visual dubbing, virtual reality, and telepresence, among others. In this chapter, we address the problem of synthesizing a photo-realistic video portrait of a *target* actor that mimics the actions of a *source* actor, where source and target can be different subjects. More specifically, our approach enables a source actor to take full control of the rigid head pose, face expressions and eye motion of the target actor; even face identity can be modified to some extent. All of these dimensions can be manipulated together or independently. Full target frames, including the entire head and hair, but also a realistic upper body and scene background complying with the modified head, are automatically synthesized.

Recently, many methods have been proposed for face-interior reenactment [Liu et al. 2001; Vlasic et al. 2005; Thies et al. 2015; Thies et al. 2016; Suwajanakorn et al. 2017; Olszewski et al. 2017]. Here, only the face expression can be modified realistically, but not the full 3D head pose, including a consistent upper body and a consistently changing background. Many of these methods fit a parametric 3D face model to RGB(-D) video [Vlasic et al. 2005; Thies et al. 2015; Thies et al. 2016], and re-render the modified model as a blended overlay over the target video for reenactment, even in real time [Thies et al. 2015; Thies et al. 2016]. Synthesizing a complete portrait video under full 3D head control is much more challenging. Averbuch-Elor et al. (2017) enable mild head pose changes driven by a source actor based on image warping. They generate reactive dynamic profile pictures from a static target portrait photo, but not fully reenacted videos. Also, large changes in head pose cause artifacts (see Section 4.7.3), the target gaze cannot be controlled, and the identity of the target person is not fully preserved (mouth appearance is copied from the source actor).

Performance-driven 3D head animation methods [Weise et al. 2011; Cao et al. 2014a; Cao et al. 2015; Ichim et al. 2015; Li et al. 2015b; Olszewski et al. 2016; Cao et al. 2016; Hu et al. 2017] are related to our work, but have orthogonal methodology and application goals. They typically drive the full head pose of stylized 3D CG avatars based on visual source actor input, e.g., for games or stylized VR environments. Recently, Cao et al. (2016) proposed image-based 3D avatars with dynamic textures based on a real-time face tracker. However, their goal is full 3D animated head control and rendering, often intentionally in a stylized rather than a photo-realistic fashion.

We take a different approach that directly generates entire photo-realistic video portraits in front of general static backgrounds under full control of a target’s head pose, facial expression, and eye motion. We formulate video portrait synthesis and reenactment as a rendering-to-video translation task. Input to our algorithm are synthetic renderings of *only* the coarse and fully-controllable 3D face interior model of a

target actor and separately rendered eye gaze images, which can be robustly and efficiently obtained via a state-of-the-art model-based reconstruction technique. The input is automatically translated into full-frame photo-realistic video output showing the entire upper body and background. Since we only track the face, we cannot actively control the motion of the torso or hair, or control the background, but our rendering-to-video translation network is able to implicitly synthesize a plausible body and background (including some shadows and reflections) for a given head pose. This translation problem is tackled using a novel space-time encoder–decoder deep neural network, which is trained in an adversarial manner. At the core of our approach is a conditional generative adversarial network (cGAN) [Isola et al. 2017], which is specifically tailored to video portrait synthesis. For temporal stability, we use a novel space-time network architecture that takes as input short sequences of conditioning input frames of head and eye gaze in a sliding window manner to synthesize each target video frame. Our target and scene-specific networks only require a few minutes of portrait video footage of a person for training. To the best of our knowledge, our approach is the first to synthesize full photo-realistic video portraits of a target person’s upper body, including realistic clothing and hair, and consistent scene background, under full 3D control of the target’s head. To summarize, this chapter makes the following technical contributions:

- A rendering-to-video translation network that transforms coarse face model renderings into full photo-realistic portrait video output.
- A novel space-time encoding as conditional input for temporally coherent video synthesis that represents face geometry, reflectance, and motion as well as eye gaze and eye blinks.
- A comprehensive evaluation on several applications to demonstrate the flexibility and effectiveness of our approach.

We demonstrate the potential and high quality of our method in many intriguing applications, ranging from face reenactment and visual dubbing for foreign language movies to user-guided interactive editing of portrait videos for movie postproduction. A comprehensive comparison to state-of-the-art methods and a user study confirm the high fidelity of our results.

4.2 Related Work

We discuss related optimization and learning-based methods that aim at reconstructing, animating and re-writing faces in images and videos, and review relevant image-to-image translation work. For a comprehensive overview of current methods we refer to a recent state-of-the-art report on monocular 3D face reconstruction, tracking and applications [Zollhöfer et al. 2018].

Monocular Face Reconstruction Face reconstruction methods aim to reconstruct 3D face models of shape and appearance from visual data. Optimization-based methods fit a 3D template model, mainly

the inner face region, to single images [Blanz and Vetter 1999; Blanz et al. 2004], unstructured image collections [Kemelmacher-Shlizerman et al. 2011; Kemelmacher-Shlizerman 2013; Roth et al. 2017] or video [Shi et al. 2014; Garrido et al. 2016; Thies et al. 2016; Suwajanakorn et al. 2014; Fyffe et al. 2014; Wu et al. 2016; Cao et al. 2014b; Ichim et al. 2015]. Recently, Booth et al. (2018) proposed a large-scale parametric face model constructed from almost ten thousand 3D scans. Learning-based approaches leverage a large corpus of images or image patches to learn a regressor for predicting either 3D face shape and appearance [Tewari et al. 2017; Tran et al. 2017; Richardson et al. 2016], fine-scale skin details [Cao et al. 2015], or both [Richardson et al. 2017; Sela et al. 2017]. Deep neural networks have been shown to be quite robust for inferring the coarse 3D facial shape and appearance of the inner face region, even when trained on synthetic data [Richardson et al. 2016]. Tewari et al. (2017) showed that encoder–decoder architectures can be trained fully unsupervised on in-the-wild images by integrating physical image formation into the network. Richardson et al. (2017) trained an end-to-end regressor to recover facial geometry at a coarse and fine-scale level. Sela et al. (2017) use an encoder–decoder network to infer a detailed depth image and a dense correspondence map, which serve as a basis for non-rigidly deforming a template mesh. Still, none of these methods creates a fully generative model for the entire head, hair, mouth interior, and eye gaze, like we do.

Video-based Facial Reenactment Facial reenactment methods re-write the face content of a target actor in a video or image by transferring facial expressions from a source actor. Facial expressions are commonly transferred via dense motion fields [Liu et al. 2001; Suwajanakorn et al. 2015b; Averbuch-Elor et al. 2017], parameters [Vlasic et al. 2005; Thies et al. 2016; Thies et al. 2018], or by warping candidate frames that are selected based on the facial motion [Dale et al. 2011], appearance metrics [Kemelmacher-Shlizerman et al. 2010] or both [Garrido et al. 2014; Li et al. 2014b]. The methods described above first reconstruct and track the source and target faces, which are represented as a set of sparse 2D landmarks or dense 3D models. Most approaches only modify the inner region of the face and thus are mainly intended for altering facial expressions, but they do not take full control of a video portrait in terms of rigid head pose, facial expression, and eye gaze. Recently, Wood et al. (2018) proposed an approach for eye gaze redirection based on a fitted parametric eye model. Their approach only provides control over the eye region.

One notable exception to pure facial reenactment is Averbuch-Elor et al.’s approach (2017), which enables the reenactment of a portrait image and allows for slight changes in head pose via image warping [Fried et al. 2016]. Since this approach is based on a single target image, it copies the mouth interior from the source to the target, thus preserving the target’s identity only partially. We take advantage of learning from a target video to allow for larger changes in head pose, facial reenactment, and joint control of the eye gaze.

Visual Dubbing Visual dubbing is a particular instance of face reenactment that aims to alter the mouth motion of the target actor to match a new audio track, commonly spoken in a foreign language by a dubbing actor. Here, we can find speech-driven [Bregler et al. 1997; Chang and Ez-

zat 2005; Ezzat et al. 2002; Liu and Ostermann 2011; Suwajanakorn et al. 2017] or performance-driven [Garrido et al. 2015; Thies et al. 2016] techniques. Speech-driven dubbing techniques learn a person-specific phoneme-to-viseme mapping from a training sequence of the actor. These methods produce accurate lip sync with visually imperceptible artifacts, as recently demonstrated by Suwajanakorn et al. (2017). However, they cannot directly control the target’s facial expressions. Performance-driven techniques overcome this limitation by transferring semantically-meaningful motion parameters and re-rendering the target model with photo-realistic reflectance [Thies et al. 2016], and fine-scale details [Garrido et al. 2015; Garrido et al. 2016]. These approaches generalize better, but do not edit the head pose and still struggle to synthesize photo-realistic mouth deformations. In contrast, our approach learns to synthesize photo-realistic facial motion and actions from coarse renderings, thus enabling the synthesis of expressions and joint modification of the head pose, with consistent body and background.

Image-to-image Translation Approaches using conditional GANs [Mirza and Osindero 2014], such as Isola et al.’s “pix2pix” (2017), have shown impressive results on image-to-image translation tasks which convert between images of two different domains, such as maps and satellite photos. These combine encoder–decoder architectures [Hinton and Salakhutdinov 2006], often with skip-connections [Ronneberger et al. 2015], with adversarial loss functions [Goodfellow et al. 2014; Radford et al. 2016]. Chen and Koltun (2017) were the first to demonstrate high-resolution results with 2 megapixel resolution, using cascaded refinement networks without adversarial training. The latest trends show that it is even possible to train high-resolution GANs [Karras et al. 2018] and conditional GANs [Wang et al. 2018] at similar resolutions. However, the main challenge is the requirement for paired training data, as corresponding image pairs are often not available. This problem is tackled by CycleGAN [Zhu et al. 2017], DualGAN [Yi et al. 2017], and UNIT [Liu et al. 2017] – multiple concurrent unsupervised image-to-image translation techniques that only require two sets of unpaired training samples. These techniques have captured the imagination of many people by translating between photographs and paintings, horses and zebras, face photos and depth as well as correspondence maps [Sela et al. 2017], and translation from face photos to cartoon drawings [Taigman et al. 2017]. Ganin et al. (2016) learn photo-realistic gaze manipulation in images. Olszewski et al. (2017) synthesize a realistic inner face texture, but cannot generate a fully controllable output video, including person-specific hair. Lassner et al. (2017) propose a generative model to synthesize people in clothing, and Ma et al. (2017) generate new images of persons in arbitrary poses using image-to-image translation. In contrast, our approach enables the synthesis of temporally-coherent video portraits that follow the animation of a source actor in terms of head pose, facial expression and eye gaze.

4.3 Overview

Our deep video portraits approach provides full control of the head of a *target actor* by transferring the rigid head pose, facial expression, and eye motion of a *source actor*, while preserving the target’s identity and appearance. Full target video frames are synthesized, including consistent upper body pos-

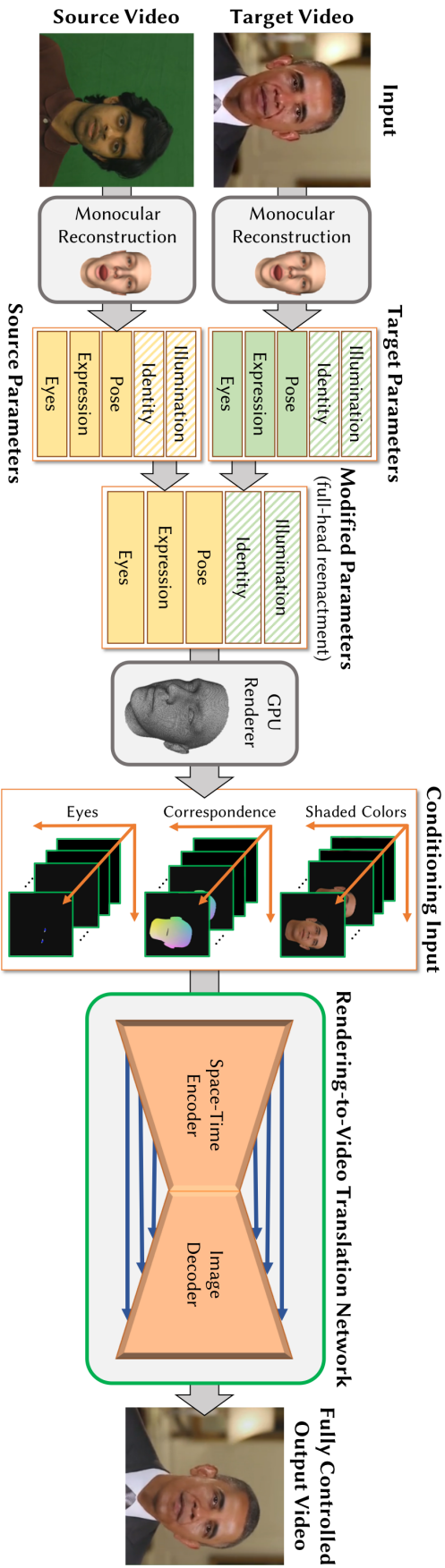


Figure 4.2: Deep video portraits enable a source actor to fully control a target video portrait. First, a low-dimensional parametric representation (left) of both videos is obtained using monocular face reconstruction. The head pose, expression and eye gaze can now be transferred in parameter space (middle). We do not focus on the modification of the identity and scene illumination (hatched background), since we are interested in reenactment. Finally, we render conditioning input images that are converted to a photo-realistic video portrait of the target actor (right). *Obama* video courtesy of the White House (public domain).

ture, hair and background. First, we track the source and target actor using a state-of-the-art monocular face reconstruction approach that uses a parametric face and illumination model (see Section 4.4). The resulting sequence of low-dimensional parameter vectors represents the actor’s identity, head pose, expression, eye gaze, and the scene lighting for every video frame (Figure 4.2, left). This allows us to transfer the head pose, expression, and/or eye gaze parameters from the source to the target, as desired. In the next step (Figure 4.2, middle), we generate new synthetic renderings of the target actor based on the modified parameters (see Section 4.5). In addition to a normal color rendering, we also render correspondence maps and eye gaze images. These renderings serve as conditioning input to our novel rendering-to-video translation network (see Section 4.6), which is trained to convert the synthetic input into photo-realistic output (see Figure 4.2, right). For temporally coherent results, our network works on space-time volumes of conditioning inputs. To process a complete video, we input the conditioning space-time volumes in a sliding window fashion, and assemble the final video from the output frames. We evaluate our approach (see Section 4.7) and show its potential on several video rewrite applications, such as full-head reenactment, gaze redirection, video dubbing, and interactive parameter-based video control.

4.4 Monocular Face Reconstruction

We employ a state-of-the-art dense face reconstruction approach that fits a parametric model of face and illumination to each video frame. It obtains a meaningful parametric face representation for the source $\mathcal{V}^s = \{I_f^s \mid f = 1, \dots, N_s\}$ and target $\mathcal{V}^t = \{I_f^t \mid f = 1, \dots, N_t\}$ video sequence, where N_s and N_t denote the total number of source and target frames, respectively. Let $\mathcal{P}^\bullet = \{\mathcal{P}_f^\bullet \mid f = 1, \dots, N_\bullet\}$ be the corresponding parameter sequence that fully describes the source or target facial performance. The set of reconstructed parameters encode the rigid head pose (rotation $\mathbf{R}^\bullet \in \text{SO}(3)$ and translation $\mathbf{t}^\bullet \in \mathbb{R}^3$), facial identity coefficients $\theta^{[s]} \in \mathbb{R}^{N_s}$ (geometry, $N_s = 80$) and $\theta^{[r]} \in \mathbb{R}^{N_r}$ (reflectance, $N_r = 80$), expression coefficients $\theta^{[e]} \in \mathbb{R}^{N_e}$ ($N_e = 64$), gaze direction for both eyes $\mathbf{e}^\bullet \in \mathbb{R}^4$, and spherical harmonics illumination coefficients $\theta^{[i]} \in \mathbb{R}^{27}$. Overall, our monocular face tracker reconstructs $N_p = 261$ parameters per video frame. In the following, we provide more details on the face tracking algorithm as well as the parametric face representation.

Parametric Face Representation We represent the space of facial identity based on a parametric head model [Banz and Vetter 1999], and the space of facial expressions via an affine model. Mathematically, we model geometry variation through an affine model $\mathbf{v} \in \mathbb{R}^{3N}$ that stacks per-vertex deformations of the underlying template mesh with N vertices, as follows:

$$\mathbf{v}(\theta^{[s]}, \theta^{[e]}) = \mathbf{a}^{[g]} + \sum_{k=1}^{N_s} \theta_k^{[s]} \mathbf{b}_k^{[s]} + \sum_{k=1}^{N_e} \theta_k^{[e]} \mathbf{b}_k^{[e]}. \quad (4.1)$$

Diffuse skin reflectance is modeled similarly by a second affine model $\mathbf{r} \in \mathbb{R}^{3N}$ that stacks the diffuse per-vertex albedo:

$$\mathbf{r}(\boldsymbol{\theta}^{[r]}) = \mathbf{a}^{[r]} + \sum_{k=1}^{N_r} \theta_k^{[r]} \mathbf{b}_k^{[r]}. \quad (4.2)$$

The vectors $\mathbf{a}^{[g]} \in \mathbb{R}^{3N}$ and $\mathbf{a}^{[r]} \in \mathbb{R}^{3N}$ store the average facial geometry and corresponding skin reflectance, respectively. The geometry basis $\{\mathbf{b}_k^{[s]}\}_{k=1}^{N_s}$ has been computed by applying principal component analysis (PCA) to 200 high-quality face scans [Blanz and Vetter 1999]. The reflectance basis $\{\mathbf{b}_k^{[r]}\}_{k=1}^{N_r}$ has been obtained in the same manner. For dimensionality reduction, the expression basis $\{\mathbf{b}_k^{[e]}\}_{k=1}^{N_e}$ has been computed using PCA, starting from the blendshapes of Alexander et al. (2010) and Cao et al. (2014b). Their blendshapes have been transferred to the topology of Blanz and Vetter (1999) using deformation transfer [Sumner and Popović 2004].

Image Formation Model To render synthetic head images, we assume a full perspective camera that maps model-space 3D points \mathbf{v} via camera space $\hat{\mathbf{v}} \in \mathbb{R}^3$ to 2D points $\mathbf{p} = \Pi(\hat{\mathbf{v}}) \in \mathbb{R}^2$ on the image plane. The perspective mapping Π contains the multiplication with the camera intrinsics and the perspective division. We assume a fixed and identical camera for all scenes, i.e., world and camera space are the same, and the face model accounts for all the scene motion. Based on a distant illumination assumption, we use the spherical harmonics (SH) basis functions $Y_b: \mathbb{R}^3 \rightarrow \mathbb{R}$ to approximate the incoming radiance \mathbf{B} from the environment:

$$\mathbf{B}(\mathbf{r}_i, \mathbf{n}_i, \boldsymbol{\theta}^{[i]}) = \mathbf{r}_i \cdot \sum_{b=1}^{B^2} \theta_b^{[i]} Y_b(\mathbf{n}_i). \quad (4.3)$$

Here, B is the number of spherical harmonics bands, $\theta_b^{[i]} \in \mathbb{R}^3$ are the SH coefficients, and \mathbf{r}_i and \mathbf{n}_i are the reflectance and unit normal vector of the i -th vertex, respectively. For diffuse materials, an average approximation error below 1 percent is achieved with only $B = 3$ bands, independent of the illumination [Ramamoorthi and Hanrahan 2001a], since the incident radiance is in general a smooth function. This results in $B^2 = 9$ parameters per color channel.

Dense Face Reconstruction We employ a dense data-parallel face reconstruction approach to efficiently compute the parameters $\boldsymbol{\theta}^*$ for both source and target videos. Face reconstruction is based on an *analysis-by-synthesis* approach that maximizes photo-consistency between a synthetic rendering of the model and the input. The reconstruction energy combines terms for dense photo-consistency, landmark alignment and statistical regularization:

$$E(\boldsymbol{\theta}) = w_{\text{photo}} E_{\text{photo}}(\boldsymbol{\theta}) + w_{\text{land}} E_{\text{land}}(\boldsymbol{\theta}) + w_{\text{reg}} E_{\text{reg}}(\boldsymbol{\theta}), \quad (4.4)$$

with $\boldsymbol{\theta} = \{\mathbf{R}^*, \mathbf{t}^*, \boldsymbol{\theta}^{[s]}, \boldsymbol{\theta}^{[r]}, \boldsymbol{\theta}^{[e]}, \boldsymbol{\theta}^{[i]}\}$. This enables the robust reconstruction of identity (geometry and skin reflectance), facial expression, and scene illumination. We use 66 automatically detected facial landmarks of the True Vision Solution tracker¹, which is a commercial implementation of Saragih et al.

¹<http://truevisionsolutions.net>

(2011b), to define the sparse alignment term E_{land} . Similar to Thies et al. (2016), we use a robust ℓ_1 -norm for dense photometric alignment E_{photo} . The regularizer E_{reg} enforces statistically plausible parameter values based on the assumption of normally distributed data. The eye gaze estimate \mathbf{e}^\bullet is directly obtained from the landmark tracker. The identity is only estimated in the first frame and is kept constant afterwards. All other parameters are estimated every frame. For more details on the energy formulation, we refer to Garrido et al. (2016) and Thies et al. (2016). We use a data-parallel implementation of iteratively re-weighted least squares (IRLS), similar to Thies et al. (2016), to find the optimal set of parameters. One difference to their work is that we compute and explicitly store the Jacobian \mathbf{J} and the residual vector \mathbf{F} to global memory based on a data-parallel strategy that launches one thread per matrix/vector element. Afterwards, a data-parallel matrix–matrix/matrix–vector multiplication computes the right- and left-hand side of the normal equations that have to be solved in each IRLS step. The resulting small linear system (97×97 in tracking mode, 6 DoF rigid pose, 64 expression parameters and 27 SH coefficients) is solved on the CPU using Cholesky factorization in each IRLS step. The reconstruction of a single frame takes 670 ms (all parameters) and 250 ms (without identity, tracking mode). This allows the efficient generation of the training corpus that is required by our space-time rendering-to-video translation network (see Section 4.6). Contrary to Garrido et al. (2016) and Thies et al. (2016), our model features dimensions to model eyelid closure, so eyelid motion is captured well.

4.5 Synthetic Conditioning Input

Using the method from Section 4.4, we reconstruct the face in each frame of the source and unmodified target video. Next, we obtain the modified parameter vector for every frame of the target sequence, e.g., for full-head reenactment, we modify the rigid head pose, expression and eye gaze of the target actor. All parameters are copied in a relative manner from the source to the target, i.e., with respect to a neutral reference frame. Then we render synthetic conditioning images of the target actor’s face model under the modified parameters using hardware rasterization. For higher temporal coherence, our rendering-to-video translation network takes a space-time volume of conditioning images $\{C_{f-o} \mid o=0, \dots, 10\}$ as input, with f being the index of the current frame. We use a temporal window of size $N_w = 11$, with the current frame being at its end. This provides the network a history of the earlier motions.

For each frame C_{f-o} of the window, we generate three different conditioning inputs: a color rendering, a correspondence image, and an eye gaze image (see Figure 4.3). The color rendering shows the modified target actor model under the estimated target illumination, while keeping the target identity (geometry and skin reflectance) fixed. This image provides a good starting point for the following rendering-to-video translation, since in the face region only the delta to a real image has to be learned. In addition to this color input, we also provide a correspondence image encoding the index of the parametric face model’s vertex that projects into each pixel. To this end, we texture the head model with a constant unique gradient texture map, and render it. Finally, we also provide an eye gaze image that solely contains the white region of both eyes and the locations of the pupils as blue circles. This

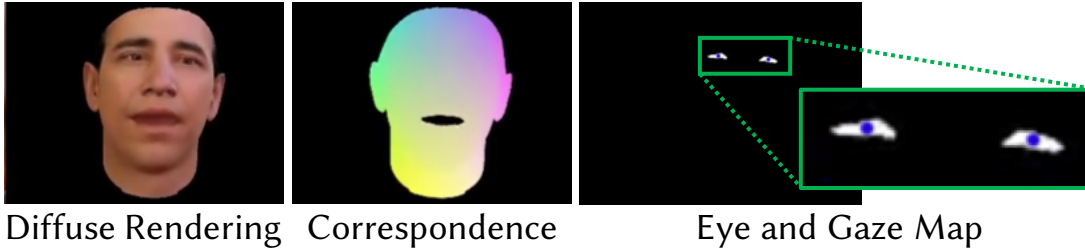


Figure 4.3: The synthetic input used for conditioning our rendering-to-video translation network: (1) colored face rendering under target illumination, (2) correspondence image, and (3) the eye gaze image.

image provides information about the eye gaze direction and blinking to the network.

We stack all N_w conditioning inputs of a time window in a 3D tensor \mathbf{X} of size $W \times H \times 9N_w$ (3 images, with 3 channels each), to obtain the input to our rendering-to-video translation network. To process the complete video, we feed the conditioning space-time volumes in a sliding window fashion. The final generated photo-realistic video output is assembled directly from the output frames.

4.6 Rendering-to-Video Translation

The generated conditioning space-time video tensors are the input to our rendering-to-video translation network. The network learns to convert the synthetic input into full frames of a photo-realistic target video, in which the target actor now mimics the head motion, facial expression and eye gaze of the synthetic input. The network learns to synthesize the entire actor in the foreground, i.e., the face for which conditioning input exists, but also all other parts of the actor, such as hair and body, so that they comply with the target head pose. It also synthesizes the appropriately modified and filled-in background, including even some consistent lighting effects between foreground and background. The network is trained for a specific target actor and a specific static, but otherwise general scene background. Our rendering-to-video translation network follows an encoder–decoder architecture and is trained in an adversarial manner based on a discriminator that is jointly trained. In the following, we explain the network architectures, the used loss functions and the training procedure in detail.

Network Architecture We show the architecture of our rendering-to-video translation network in Figure 4.4. Our conditional generative adversarial network consists of a space-time transformation network \mathbf{T} and a discriminator \mathbf{D} . The transformation network \mathbf{T} takes the $W \times H \times 9N_w$ space-time tensor \mathbf{X} as input and outputs a photo-real image $\mathbf{T}(\mathbf{X})$ of the target actor. The temporal input enables the network to take the history of motions into account by inspecting previous conditioning images. The temporal axis of the input tensor is aligned along the network channels, i.e., the convolutions in the first layer have $9N_w$ channels. Note, we store all image data in normalized $[-1, +1]$ -space, i.e, black is mapped to $[-1, -1, -1]^\top$ and white is mapped to $[+1, +1, +1]^\top$.

Our network consists of two main parts, an encoder for computing a low-dimensional latent representation, and a decoder for synthesizing the output image. We employ skip connections [Ron-

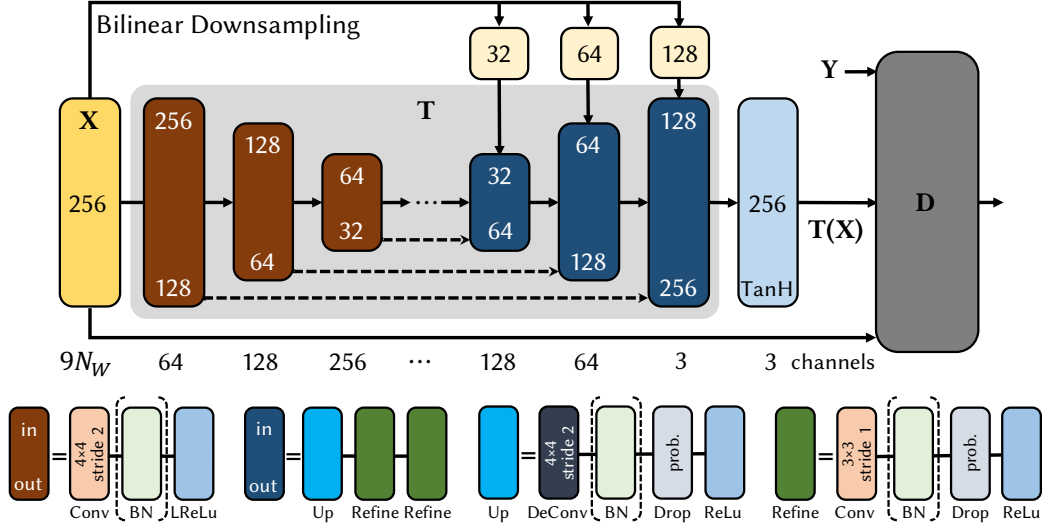


Figure 4.4: Architecture of our rendering-to-video translation network for an input resolution of 256×256 : The encoder has 8 downsampling modules with $(64, 128, 256, 512, 512, 512, 512, 512)$ output channels. The decoder has 8 upsampling modules with $(512, 512, 512, 512, 256, 128, 64, 3)$ output channels. The upsampling modules use the following dropout probabilities $(0.5, 0.5, 0.5, 0, 0, 0, 0, 0)$. The first downsampling and the last upsampling module do not employ batch normalization (BN). The final non-linearity (TanH) brings the output to the employed normalized $[-1, +1]$ -space.

[neberger et al. 2015] to enable the network to transfer fine-scale structure. To generate video frames with sufficient resolution, our network also employs a cascaded refinement strategy [Chen and Koltun 2017]. In each downsampling step, we use a convolution (4×4 , stride 2) followed by batch normalization and a leaky ReLU non-linearity. The upsampling module is specifically designed to produce high-quality output, and has the following structure: first, the resolution is increased by a factor of two based on deconvolution (4×4 , upsampling factor of 2), batch normalization, dropout and ReLU. Afterwards, two refinement steps based on convolution (3×3 , stride 1, stays on the same resolution) and ReLU are applied. The final hyperbolic tangent non-linearity (TanH) brings the output tensor to the normalized $[-1, +1]$ -space used for storing the image data. For more details, please refer to Figure 4.4.

The input to our discriminator \mathbf{D} is the conditioning input tensor \mathbf{X} (size $W \times H \times 9N_w$), and either the predicted output image $\mathbf{T}(\mathbf{X})$ or the ground-truth image, both of size $W \times H \times 3$. The employed discriminator is inspired by the PatchGAN classifier, proposed by Isola et al. (2017). We extended it to take volumes of conditioning images as input.

Objective Function We train in an adversarial manner to find the best rendering-to-video translation network:

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \max_{\mathbf{D}} E_{c\text{GAN}}(\mathbf{T}, \mathbf{D}) + \lambda E_{\ell_1}(\mathbf{T}). \quad (4.5)$$

This objective function comprises an adversarial loss $E_{c\text{GAN}}(\mathbf{T}, \mathbf{D})$ and an ℓ_1 -norm reproduction loss $E_{\ell_1}(\mathbf{T})$. The constant weight of $\lambda = 100$ balances the contribution of these two terms. The adversarial

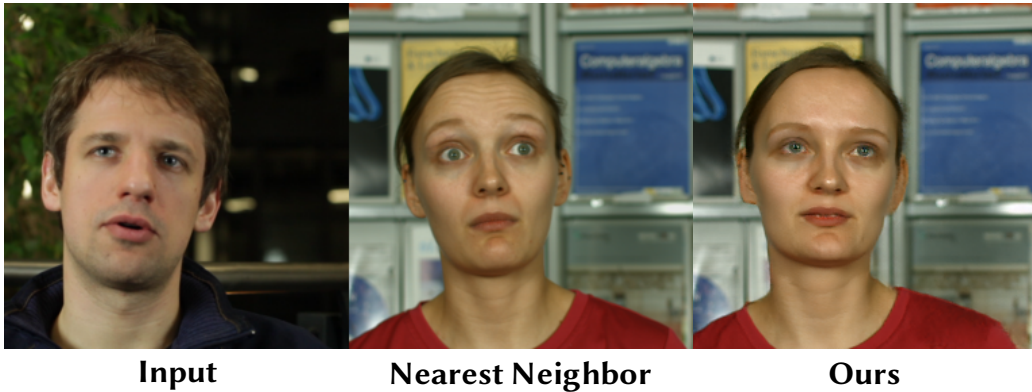


Figure 4.5: Comparison to a nearest-neighbor approach in parameter space (pose and expression). Our results have higher quality and are temporally more coherent (see supplemental video at the project website). For the nearest-neighbor approach, it is difficult to find the right trade-off between pose and expression. This leads to many results with one of the two dimensions not being well-matched. The results are also temporally unstable, since the nearest neighbor abruptly changes, especially for small training sets.

loss has the following form:

$$E_{\text{GAN}}(\mathbf{T}, \mathbf{D}) = \mathbb{E}_{\mathbf{X}, \mathbf{Y}} [\log \mathbf{D}(\mathbf{X}, \mathbf{Y})] + \mathbb{E}_{\mathbf{X}} [\log (1 - \mathbf{D}(\mathbf{X}, \mathbf{T}(\mathbf{X})))] . \quad (4.6)$$

We do not inject a noise vector while training our network to produce deterministic outputs. During adversarial training, the discriminator \mathbf{D} tries to get better at classifying given images as *real* or *synthetic*, while the transformation network \mathbf{T} tries to improve in fooling the discriminator. The ℓ_1 -norm loss penalizes the distance between the synthesized image $\mathbf{T}(\mathbf{X})$ and the ground-truth image \mathbf{Y} , which encourages the sharpness of the synthesized output:

$$E_{\ell_1}(\mathbf{T}) = \mathbb{E}_{\mathbf{X}, \mathbf{Y}} [\|\mathbf{Y} - \mathbf{T}(\mathbf{X})\|_1] . \quad (4.7)$$

Training We construct the training corpus $\mathcal{T} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_i$ based on the tracked video frames of the target video sequence. Typically, two thousand video frames, i.e., about one minute of video footage, are sufficient to train our network (see Section 4.7). Our training corpus consists of $N_t - (N_w - 1)$ rendered conditioning space-time volumes \mathbf{X}_i and the corresponding ground-truth image \mathbf{Y}_i (using a window size of $N_w = 11$). We train our networks using the TensorFlow [Abadi et al. 2015] deep learning framework. The gradients for back-propagation are obtained using Adam [Kingma and Ba 2015]. We train for 31,000 iterations with a batch size of 16 (approx. 250 epochs for a training corpus of 2000 frames) using a base learning rate of 0.0002 and first momentum of 0.5; all other parameters have their default value. We train our networks from scratch, and initialize the weights based on a Normal distribution $\mathcal{N}(0, 0.2)$.

4.7 Results

Our approach enables full-frame target video portrait synthesis under full 3D head pose control. We measured the runtime for training and testing on an Intel Xeon E5-2637 with 3.5 GHz (16 GB RAM)



Figure 4.6: Qualitative results of full-head reenactment: our approach enables full-frame target video portrait synthesis under full 3D head pose control. The output video portraits are photo-realistic and hard to distinguish from real videos. Note that even the shadow in the background of the second row moves consistently with the modified foreground head motion. In the sequence at the top, we only transfer the translation in the camera plane, while we transfer the full 3D translation for the sequence at the bottom. For full sequences, please refer to our video. *Obama* video courtesy of the White House (public domain).

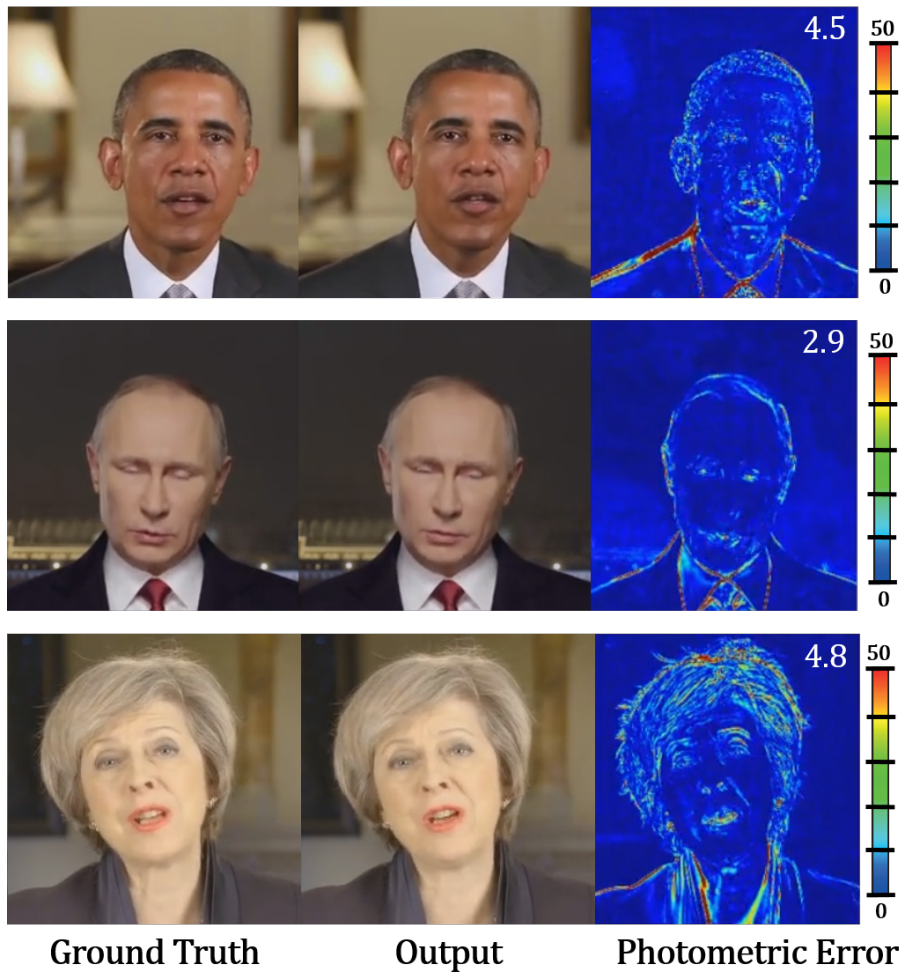


Figure 4.7: Quantitative evaluation of the photometric re-rendering error. We evaluate our approach quantitatively in a self-reenactment setting, where the ground-truth video portrait is known. We train our rendering-to-video translation network on two thirds of the video sequence, and test on the remaining third. The error maps show per-pixel Euclidean distance in RGB (color channels in $[0,255]$); the mean photometric error of the test set is shown in the top-right. The error is consistently low in regions with conditioning input, with higher errors in regions without conditioning, such as the upper body. *Obama* video courtesy of the White House (public domain). *Putin* video courtesy of the Kremlin (CC BY). *May* video courtesy of the UK government (Open Government Licence).

and an NVIDIA GeForce GTX Titan Xp (12 GB RAM). Training our network takes 10 hours for a target video resolution of 256×256 pixels, and 42 hours for 512×512 pixels. Tracking the source actor takes 250 ms per frame (without identity), and the rendering-to-video conversion (inference) takes 65 ms per frame for 256×256 pixels, or 196 ms for 512×512 pixels.

In the following, we evaluate the design choices of our deep video portrait algorithm, compare to current state-of-the-art reenactment approaches, and show the results of a large-scale web-based user study. We further demonstrate the potential of our approach on several video rewrite applications, such as reenactment under full head and facial expression control, facial expression reenactment only, video dubbing, and live video portrait editing under user control. For more results, we refer to our supplemen-

tal video at the project website². In total, we applied our approach to 14 different target sequences of 13 different subjects and used 5 different source sequences; see Section 4.8 for details. A comparison to a simple nearest-neighbor retrieval approach can be found in Figure 4.5 and in the supplemental video at the project website. Our approach requires only a few minutes of target video footage for training.

4.7.1 Applications

Our approach enables us to take full control of the rigid head pose, facial expression, and eye motion of a target actor in a video portrait, thus opening up a wide range of video rewrite applications. All parameter dimensions can be estimated and transferred from a source video sequence or edited manually through an interactive user interface.

Reenactment under full head control Our approach is the first that can photo-realistically transfer the full 3D head pose (spatial position and rotation), facial expression, as well as eye gaze and eye blinking of a captured source actor to a target actor video. Figure 4.6 shows some examples of full-head reenactment between different source and target actors. Here, we use the full target video for training and the source video as the driving sequence. As can be seen, the output of our approach achieves a high level of realism and faithfully mimics the driving sequence, while still retaining the mannerisms of the original target actor. Note that the shadow in the background moves consistently with the position of the actor in the scene, as shown in Figure 4.6 (second row). We also demonstrate the high quality of our results and evaluate our approach quantitatively in a self-reenactment scenario, see Figure 4.7. For the quantitative analysis, we use two thirds of the target video for training and one third for testing. We capture the face in the training and driving video with our model-based tracker, and then render the conditioning images, which serve as input to our network for synthesizing the output. For further details, please refer to Section 4.7.2. Note that the synthesized results are nearly indistinguishable from the ground truth.

Facial Reenactment and Video Dubbing Besides full-head reenactment, our approach also enables facial reenactment. In this experiment, we replace the expression coefficients of the target actor with those of the source actor before synthesizing the conditioning input to our rendering-to-video translation network. Here, the head pose and position, and eye gaze remain unchanged. Figure 4.8 shows facial reenactment results. Observe that the face expression in the synthesized target video nicely matches the expression of the source actor in the driving sequence. Please refer to the supplemental video at the project website for the complete video sequences.

Our approach can also be applied to visual dubbing. In many countries, foreign-language movies are dubbed, i.e., the original voice of an actor is replaced with that of a dubbing actor speaking in another language. Dubbing often causes visual discomfort due to the discrepancy between the actor’s mouth motion and the new audio track. Even professional dubbing studios achieve only approximate audio

²Project page: <http://gvv.mpi-inf.mpg.de/projects/DeepVideoPortraits>



Figure 4.8: Facial reenactment results of our approach. We transfer the expressions from the source to the target actor, while retaining the head pose (rotation and translation) as well as the eye gaze of the target actor. For the full sequences, please refer to the supplemental video at the project website. *Obama* video courtesy of the White House (public domain). *Putin* video courtesy of the Kremlin (CC BY). *Reagan* video courtesy of the National Archives and Records Administration (public domain).

alignment at best. Visual dubbing aims at altering the mouth motion of the target actor to match the new foreign-language audio track spoken by the dubber. Figure 4.9 shows results where we modify the facial motion of actors speaking originally in German to adhere to an English translation spoken by a professional dubbing actor, who was filmed in a dubbing studio [Garrido et al. 2015]. More precisely, we transfer the captured facial expressions of the dubbing actor to the target actor, while leaving the original target gaze and eye blinks intact, i.e., we use the original eye gaze images of the tracked target sequence as conditioning. As can be seen, our approach achieves dubbing results of high quality. In fact, we produce images with more realistic mouth interior and more emotional content in the mouth region. Please see the supplemental video at the project website for full video results.

Interactive Editing of Video Portraits We built an interactive editor that enables users to reanimate video portraits with live feedback by modifying the parameters of the coarse face model rendered

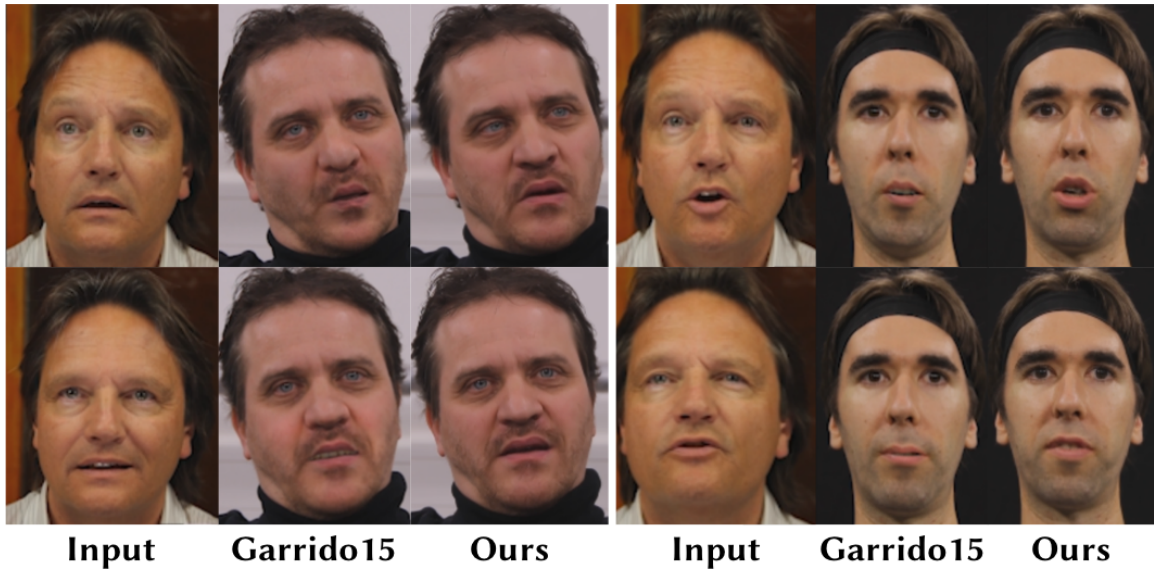


Figure 4.9: Dubbing comparison on two sequences of [Garrido et al. \(2015\)](#). For visual dubbing, we transfer the facial expressions of the dubbing actor (‘input’) to the target actor. We compare our results to [Garrido et al.’s](#). Our approach obtains higher quality results in terms of the synthesized mouth shape and mouth interior. Note that our approach also enables full-head reenactment in addition to expression transfer. For the full comparison, we refer to the supplemental video at the project website.

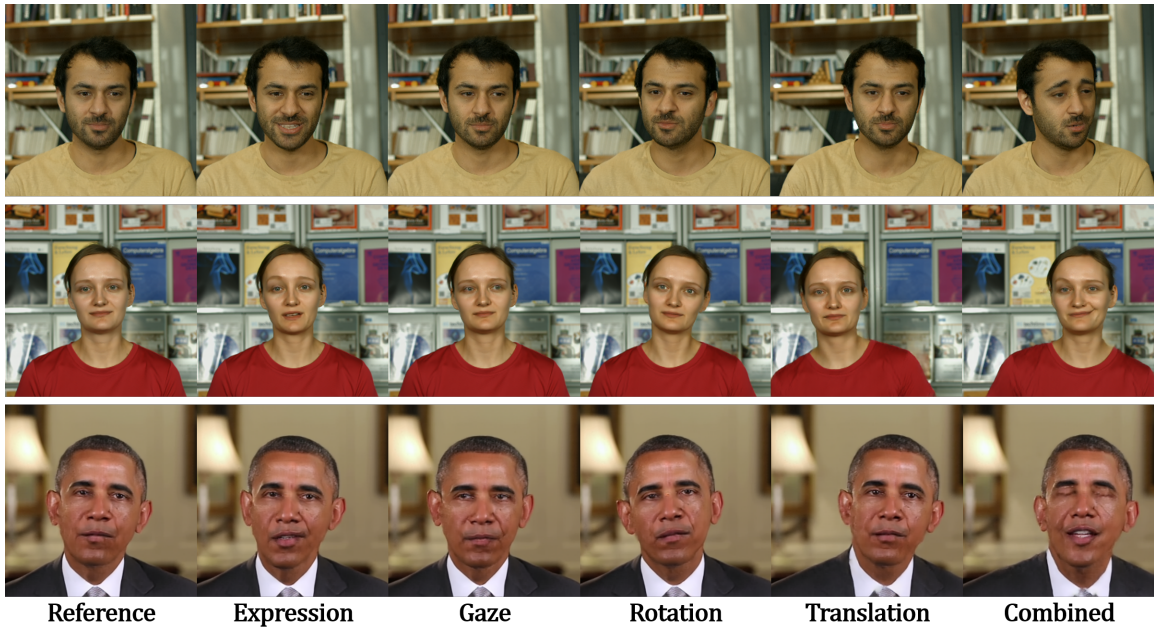


Figure 4.10: Interactive editing. Our approach provides full parametric control over video portraits (by controlling head model parameters in conditioning images). This enables modifications of the rigid head pose (rotation and translation), facial expression and eye motion. All of these dimensions can be manipulated together or independently. We also show these modifications live in the supplemental video at the project website. *Obama* video courtesy of the White House (public domain).



Figure 4.11: Identity modification. While not the main focus of our approach, it also enables modification of the facial shape via the geometry shape parameters. This shows that our network picks up the correspondence between the model and the video portrait. Note that the produced outputs are also consistent in regions that are not constrained by the conditioning input, such as the hair and background.



Figure 4.12: Video Teleconferencing. From left to right: Original camera view and synthesized image with head pose corrected. A traditional system for video teleconferencing often breaks the eye contact between participants due to cameras located on top of monitors. Our method enables head pose and gaze corrections, leading to a more natural conversation.

into the conditioning images (see our live demo in the supplemental video at the project website). Figure 4.10 shows a few static snapshots that were taken while the users were playing with our editor. Our approach enables changes of all parameter dimensions, either independently or all together, as shown in Figure 4.10. More specifically, we show independent changes of the expression, head rotation, head translation, and eye gaze (including eye blinks). Please note the realistic and consistent generation of the torso, head and background. Even shadows or reflections appear very consistently in the background. In addition, we show user edits that modify all parameters simultaneously. Our interactive editor runs at approximately 9 fps. While not the focus of this chapter, our approach also enables modifications of the geometric facial identity, see Figure 4.11. These combined modifications show as a proof of concept that our network generalizes beyond the training corpus.

Video Teleconferencing Our method can also be applied to a video teleconferencing system. A common problem in video teleconferencing is an arrangement of cameras located on top of monitors, which results in breaking the eye contact between participants. With our approach, we can modify

the head pose and gaze to restore the eye contact as shown in Figure 4.12. Another interesting aspect of our method regarding video teleconferencing is model-based video coding. Instead of compressing whole video, we can send model parameters, which only requires a bandwidth of 31 KB/s. In contrast, Skype’s h.264 video streaming requires a bandwidth of around 192 KB/s, which is 6 times more. Note that this requires to transfer the trained network beforehand. As a proof of concept, this shows a potential of video teleconferencing for the future.

4.7.2 Quantitative Evaluation

We performed a quantitative evaluation of the re-rendering quality. First, we evaluate our approach in a self-reenactment setting, where the ground-truth video portrait is known. We train our rendering-to-video translation network on the first two thirds of a video sequence and test it on the remaining last third of the video, see Figure 4.7. The photometric error maps show the per-pixel Euclidean distance in RGB color space, with each channel being in $[0,255]$. We performed this test for three different videos and the mean photometric errors are 2.88 (Vladimir Putin), 4.76 (Theresa May), and 4.46 (Barack Obama). Our approach obtains consistently low error in regions with conditioning input (face) and higher errors are found in regions that are unexplained by the conditioning input. Please note that while the synthesized video portraits slightly differ from the ground truth outside the face region, the synthesized hair and upper body are still plausible, consistent with the face region, and free of visual artifacts. For a complete analysis of these sequences, we refer to the supplemental video at the project website.

We evaluate our space-time conditioning strategy in Figure 4.13. Without space-time conditioning, the photometric error is significantly higher. The average errors over the complete sequence are 4.9 without vs. 4.5 with temporal conditioning (Barack Obama) and 5.3 without vs. 4.8 with temporal conditioning (Theresa May). In addition to a lower photometric error, space-time conditioning also leads to temporally significantly more stable video outputs. This can be seen best in the supplemental video at the project website.

We also evaluate the importance of the training set size. In this experiment, we train our rendering-to-video translation network with 500, 1000, 2000 and 4000 frames of the target sequence, see Figure 4.14. As can be expected, larger training sets produce better results, and the best results are obtained with the full training set.

We also evaluate different image resolutions by training our rendering-to-video translation network for resolutions of 256×256 , 512×512 and 1024×1024 pixels. We evaluate the quality in the self-reenactment setting, as shown in Figure 4.15. Generative networks of higher resolution are harder to train and require significantly longer training times: 10 hours for 256×256 , 42 hours for 512×512 , and 110 hours for 1024×1024 (on a Titan Xp). Therefore, we use a resolution of 256×256 pixels for most results.

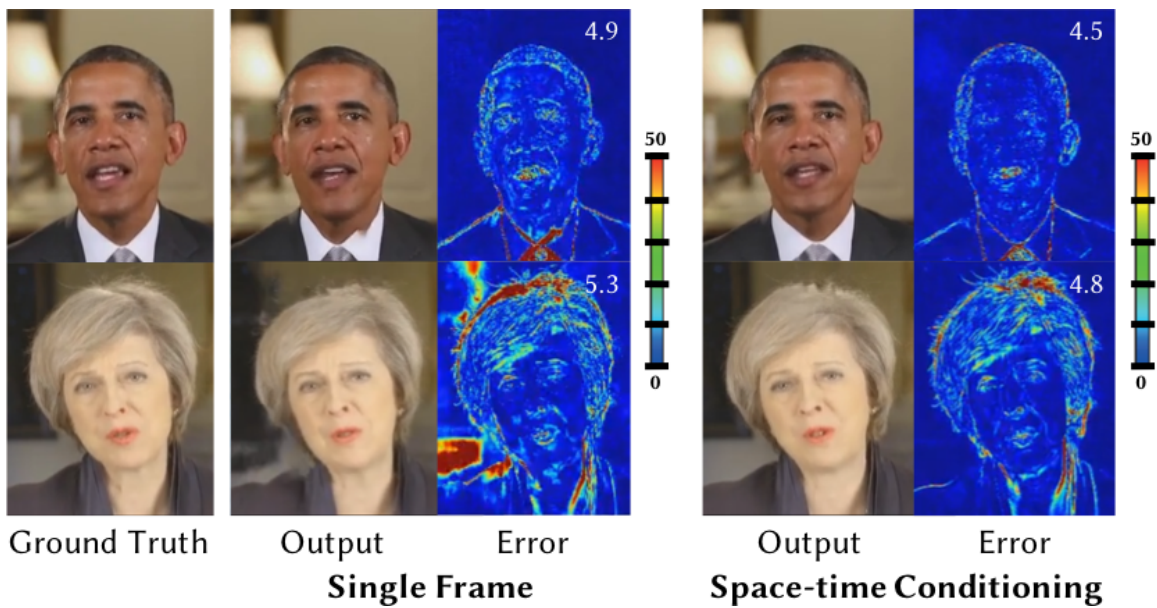


Figure 4.13: Quantitative evaluation of the influence of the proposed space-time conditioning input. The error maps show the per-pixel distance in RGB color space with each channel being in $[0,255]$; the mean photometric error is shown in the top-right. Without space-time conditioning, the photometric error is higher. Temporal conditioning adds significant temporal stability. This is best seen in the supplemental video at the project website. *Obama* video courtesy of the White House (public domain). *May* video courtesy of the UK government (Open Government Licence).

4.7.3 Comparisons to the State-of-the-Art

We compare our deep video portrait approach to current state-of-the-art video and image reenactment techniques.

Comparison to Thies et al. (2016) We compare our approach to the state-of-the-art *Face2Face* facial reenactment method of Thies et al. (2016). In comparison to *Face2Face*, our approach achieves expression transfer of similar quality. What distinguishes our approach is the capability for full-head reenactment, i.e., the ability to also transfer the rigid head pose, gaze direction, and eye blinks in addition to the facial expressions, as shown in Figure 4.16. As can be seen, in our result, the head pose and eye motion nicely matches the source sequence, while the output generated by *Face2Face* follows the head and eye motion of the original target sequence. Please see the supplemental video at the project website for the video result.

Comparison to Suwajanakorn et al. (2017) We also compare to the audio-based dubbing approach of Suwajanakorn et al. (2017), see Figure 4.17. Their *AudioToObama* approach produces accurate lip sync with visually imperceptible artifacts, but provides no direct control over facial expressions. Thus, the expressions in the output do not always perfectly match the input (box, mouth), especially for expression changes without an audio cue. Our visual dubbing approach accurately transfers the expressions from the source to the target. In addition, our approach provides more control over the

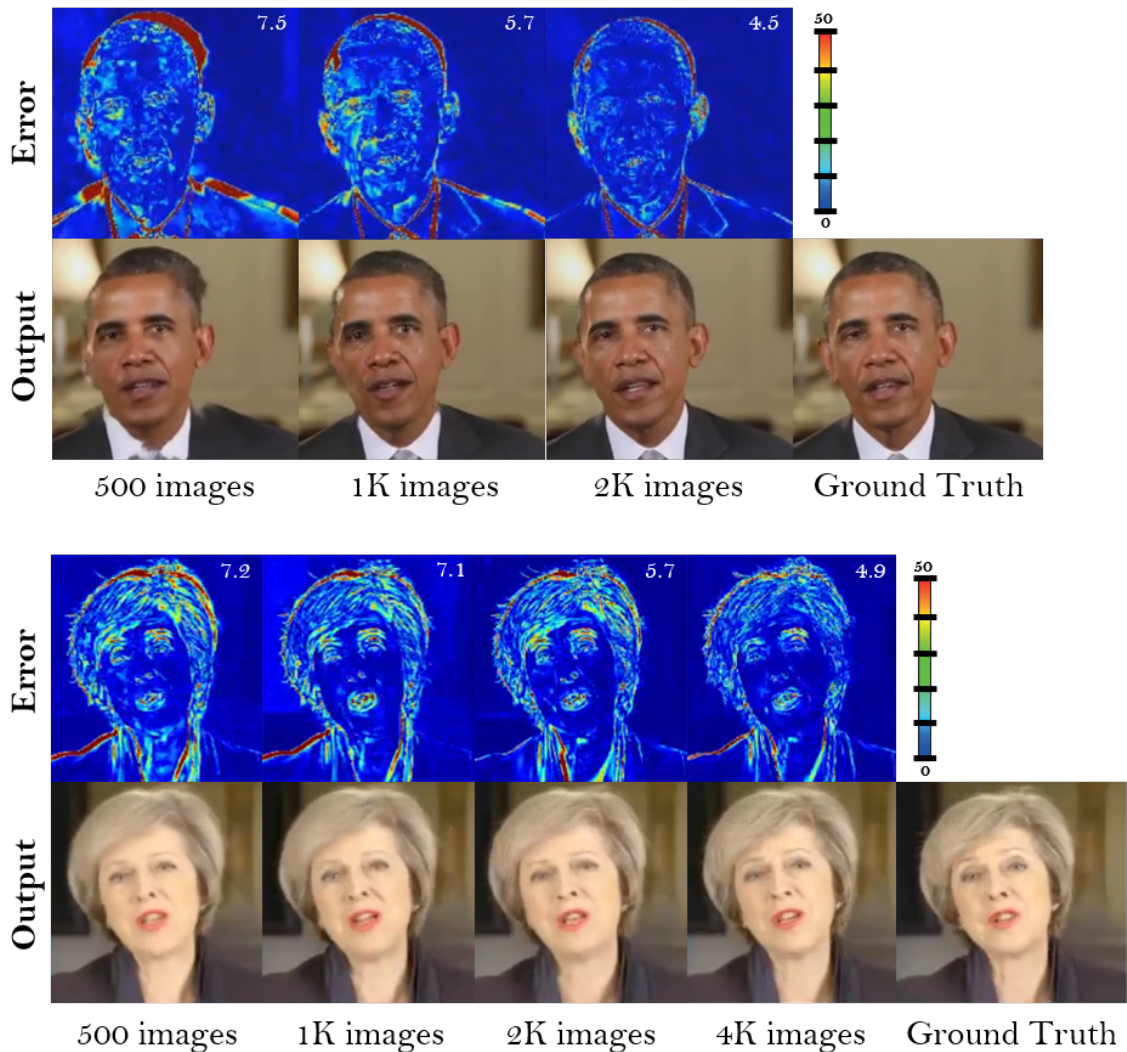


Figure 4.14: Quantitative evaluation of the training set size. We train our rendering-to-video translation network with training corpora of different sizes. The error maps show the per-pixel distance in RGB color space with each channel being in $[0,255]$; the mean photometric error is shown in the top-right. Smaller training sets have larger photometric errors, especially for regions outside of the face. For the full comparison, we refer to the supplemental video at the project website. *Obama* video courtesy of the White House (public domain). *May* video courtesy of the UK government (Open Government Licence).

target video by also transferring the eye gaze and eye blinks (box, eyes) and the general rigid head pose (arrows). While their approach is trained on a huge amount of training data (17 hours), our approach only uses a small training dataset (1.3 minutes). The differences are best visible in the supplemental video at the project website.

Comparison to Averbuch-Elor et al. (2017) We compare our approach in the full-head reenactment scenario to the image reenactment approach of Averbuch-Elor et al. (2017), see Figure 4.18. Their approach does not preserve the identity of the target actor, since they copy the teeth and mouth interior from the source to the target sequence. Our learning-based approach enables larger modifications of

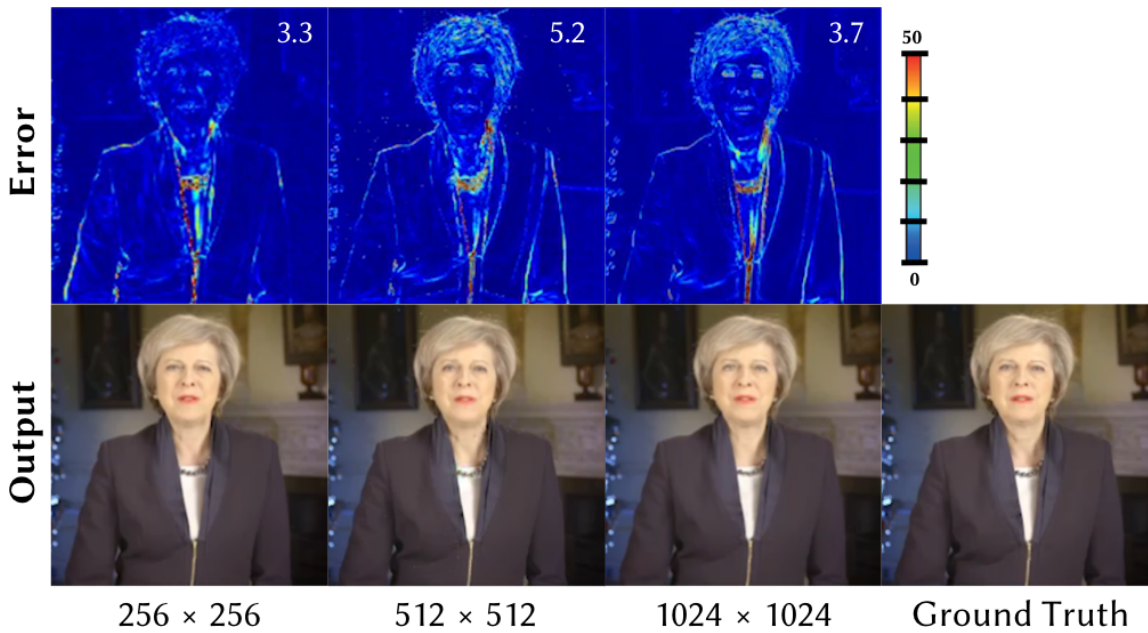


Figure 4.15: Quantitative comparison of different resolutions. We train three rendering-to-video translation networks for resolutions of 256×256 , 512×512 and 1024×1024 pixels. The error maps show the per-pixel distance in RGB color space with each channel being in $[0, 255]$; the mean photometric error is shown in the top-right. For the full comparison, see our video. *May* video courtesy of the UK government (Open Government Licence).

the head pose without apparent artifacts, while their warping-based approach significantly distorts the head and background. In addition, we enable the joint modification of the gaze direction and eye blinks; see supplemental video at the project website.

4.7.4 User Study

We conducted two extensive web-based user studies to quantitatively evaluate the realism of our results. We prepared short 5-second video clips that we extracted from both real and synthesized videos (see Figure 4.19), to evaluate three applications of our approach: self-reenactment, same-person-reenactment and visual dubbing. We opted for self-reenactment, same-person-reenactment (two speeches of Barack Obama) and visual dubbing to guarantee that the motion types in the evaluated real and synthesized video pairs are matching. This eliminates the motion type as a confounding factor from the statistical analysis, e.g., having unrealistic motions for a public speech in the synthesized videos would negatively bias the outcome of the study. Our evaluation is focused on the visual quality of the synthesized results. Most video clips have a resolution of 256×256 pixels, but some are 512×512 pixels. In our user study, we presented one video clip at a time, and asked participants to respond to the statement "*This video clip looks real to me*" on a 5-point Likert scale (1–*strongly disagree*, 2–*disagree*, 3–*don't know*, 4–*agree*, 5–*strongly agree*). Video clips are shown in a random order, and each video clip is shown exactly once to assess participants' first impression. We recruited 135 and 69 anonymous participants for our two studies, largely from North America and Europe.

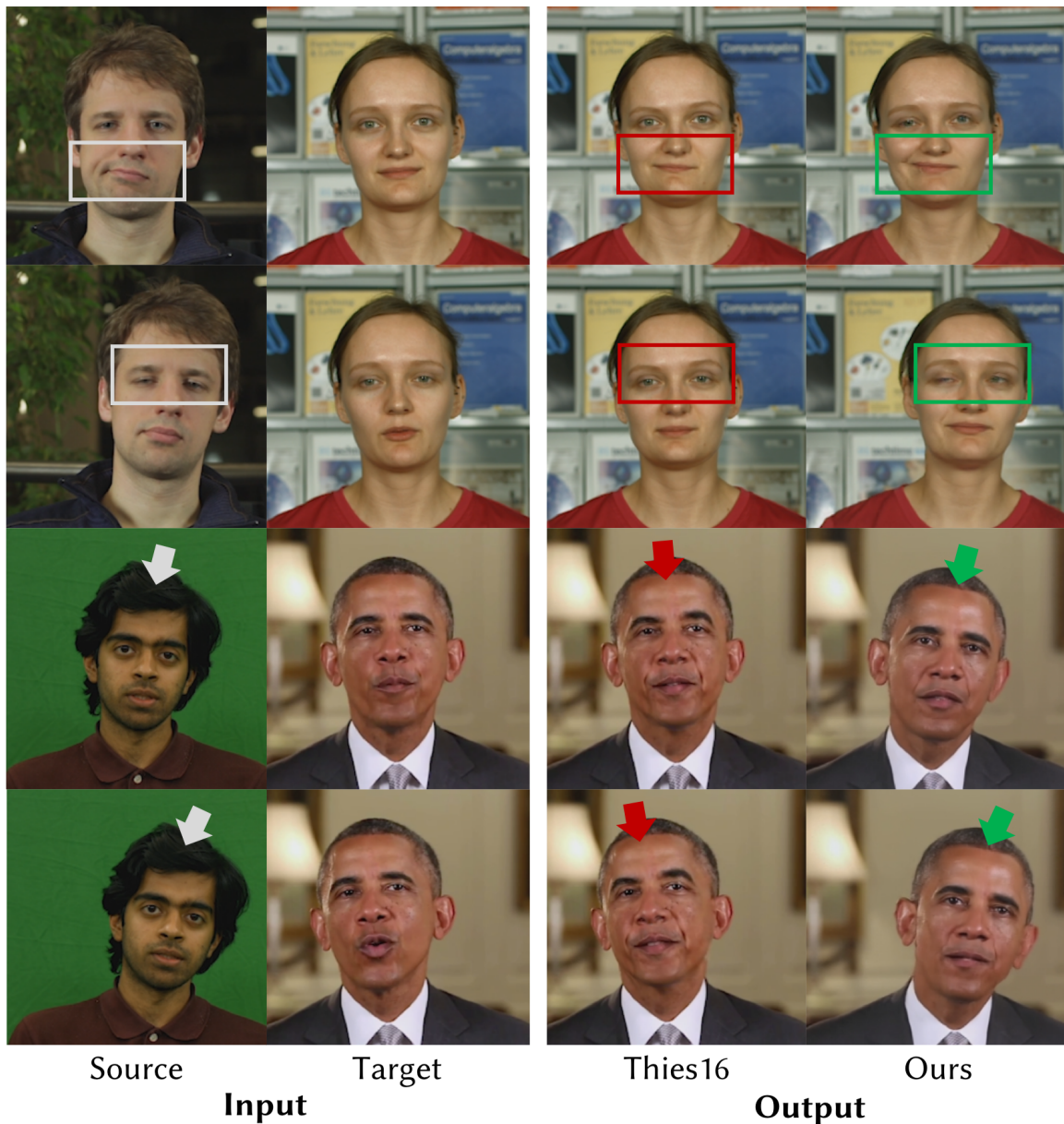


Figure 4.16: Comparison to the state-of-the-art facial reenactment approach of Thies et al. (2016). Our approach achieves expression transfer of similar quality, while also enabling full-head reenactment, i.e., it also transfers the rigid head pose, gaze direction, and eye blinks. For the video result, we refer to the supplemental video at the project website. Obama video courtesy of the White House (public domain).

The results in Table 4.1 show that only 80% of participants rated real 256×256 videos as real, i.e., (strongly) agreeing to the video looking real; it seems that in anticipation of synthetic video clips, participants became overly critical. At the same time, 50% of participants consider our 256×256 results to be real, which increases slightly to 52% for 512×512 . Our best result is the self-reenactment of Vladimir Putin at 256×256 resolution, which 65% of participants consider to be real, compared to 78% for the real video. We also evaluated partial and full reenactment by transferring a speech by Barack Obama to another video clip of himself. Table 4.2 indicates that we achieve better realism ratings with full reenactment comprising facial expressions and pose (50%) compared to transferring

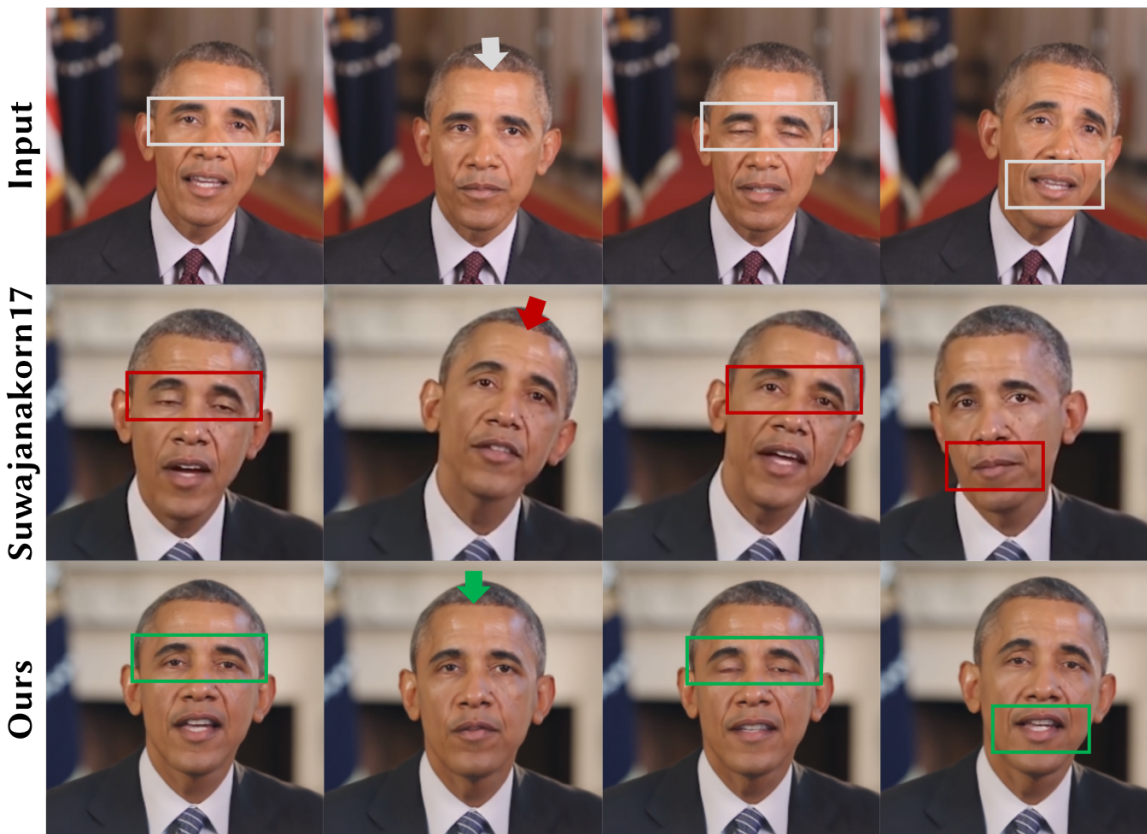


Figure 4.17: Comparison to [Suwajanakorn et al. \(2017\)](#). Their approach produces accurate lip sync with visually imperceptible artifacts, but provides no direct control over facial expressions. Thus, the expressions in the output do not always perfectly match the input (box, mouth), especially for expression changes without audio cue. Our visual dubbing approach accurately transfers the expressions from the source to the target. In addition, our approach provides more control over the target video by also transferring the eye gaze and eye blinks (box, eyes), and the rigid head pose (arrows). Since the source sequence shows more head-pose variation than the target sequence, we scaled the transferred rotation and translation by 0.5 in this experiment. For the full video sequence, we refer to the supplemental video at the project website. *Obama* video courtesy of the White House (public domain).

only facial expressions (38%). This might be because full-head reenactment keeps expressions and head motion synchronized. [Suwajanakorn et al.](#)'s speech-driven reenactment approach (2017) achieves a realism rating of 64% compared to the real source and target video clips, which achieve 70–86%. Our full-head reenactment results are considered to be at least as real as [Suwajanakorn et al.](#)'s by 60% of participants. We finally compared our dubbing results to VDub [[Garrido et al. 2015](#)] in Table 4.3. Overall, 57% of participants gave our results a higher realism rating (and 32% gave the same rating). Our results are again considered to be real by 51% of participants, compared to only 21% for VDub.

On average, across all scenarios and both studies, our results are considered to be real by 47% of the participants (1,767 ratings), compared to only 80% for real video clips (1,362 ratings). This suggests that our results already fool about 60% of the participants – a good result given the critical participant pool. However, there is some variation across our results: lower realism ratings were given for well-known personalities like Barack Obama, while higher ratings were given for instance to the unknown dubbing actors.

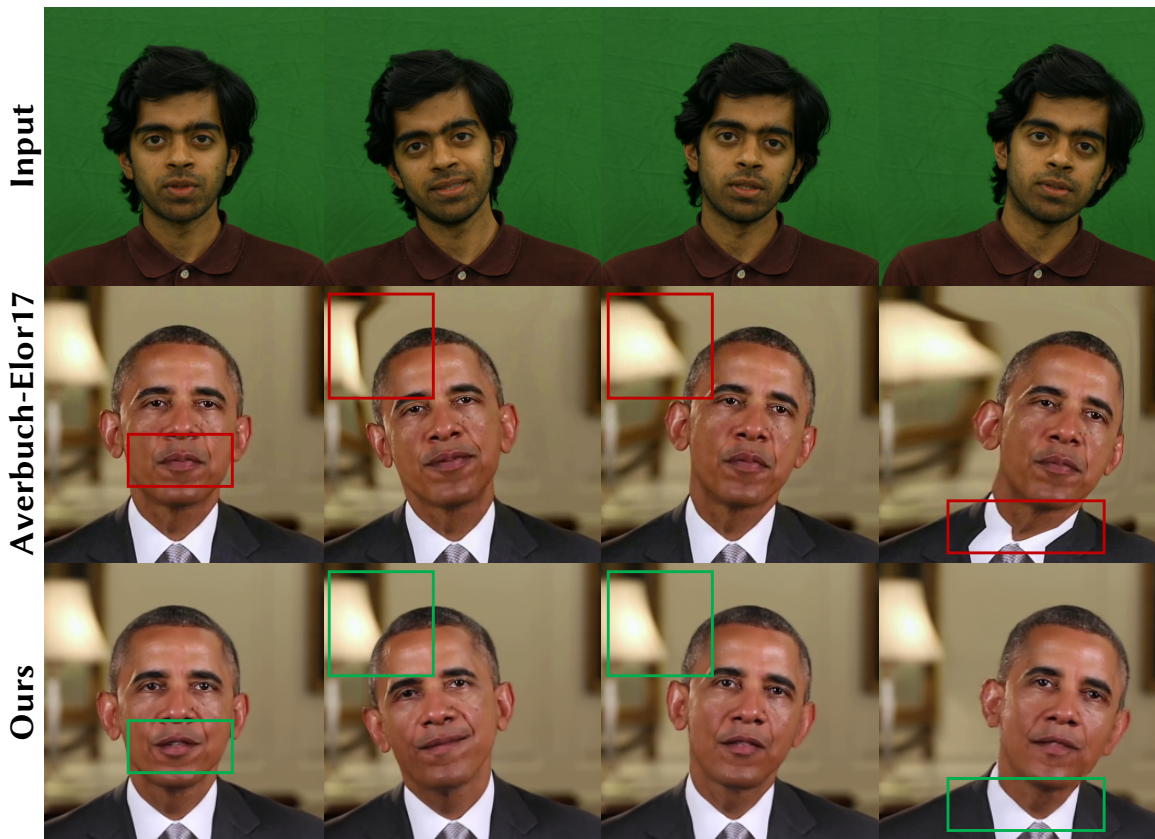


Figure 4.18: Comparison to the image reenactment approach of [Averbuch-Elor et al. \(2017\)](#) in the full-head reenactment scenario. Since their method is based on a single target image, they copy the mouth interior from the source to the target, thus not preserving the target’s identity. Our learning-based approach enables larger modifications of the rigid head pose without apparent artifacts, while their warping-based approach distorts the head and background. In addition, ours enables joint control of the eye gaze and eye blinks. The differences are most evident in the supplemental video at the project website. *Obama* video courtesy of the White House (public domain).



Figure 4.19: We performed a user study to evaluate the quality of our results and see if users can distinguish between real (top) and synthesized video clips (bottom). The video clips include self-reenactment, same-person-reenactment, and video dubbing. *Putin* video courtesy of the Kremlin (CC BY). *Obama* video courtesy of the White House (public domain). *Elizabeth II* video courtesy of the Governor General of Canada (public domain).

Table 4.1: User study results for self-reenacted videos ($n = 135$). Columns 1–5 show the percentage of ratings given about the statement "This video clip looks real to me", from 1 (*strongly disagree*) to 5 (*strongly agree*). 4+5='real'.

	res	Real videos					Our results						
		1	2	3	4	5	'real'	1	2	3	4	5	'real'
Obama	256	2	8	10	62	19	81%	13	33	11	37	6	43%
Putin	256	2	11	10	58	20	78%	3	17	15	54	11	65%
Eliabeth II	256	2	6	12	59	21	80%	6	32	20	33	9	42%
Obama	512	0	7	3	49	42	91%	9	35	13	36	8	44%
Putin	512	4	13	10	47	25	72%	2	20	15	44	19	63%
Eliabeth II	512	1	7	4	55	34	89%	7	33	10	38	13	51%
Mean	256	2	8	10	60	20	80%	7	27	15	41	9	50%
Mean	512	2	9	6	50	34	84%	6	29	12	39	13	52%

Table 4.2: User study results for expression and full head transfer between two videos of Barack Obama compared to the input videos and [Suwajanakorn et al.](#)'s approach ($n = 69$, mean of 4 clips).

	Ratings					
	1	2	3	4	5	'real'
Source video (real)	0	8	6	43	42	86%
Target video (real)	1	14	14	47	23	70%
Suwajanakorn et al. (2017)	2	20	14	47	17	64%
Expression transfer (ours)	9	37	17	29	9	38%
Full head transfer (ours)	3	31	16	37	13	50%

Table 4.3: User study results for dubbing comparison to VDub ($n = 135$).

	Garrido et al. (2015)					Our results						
	1	2	3	4	5	'real'	1	2	3	4	5	'real'
Ingmar (3 clips)	21	36	21	20	2	22%	4	21	25	42	8	50%
Thomas (3 clips)	33	36	11	16	4	20%	7	25	17	42	9	51%
Mean (6 clips)	27	36	16	18	3	21%	6	23	21	42	9	51%










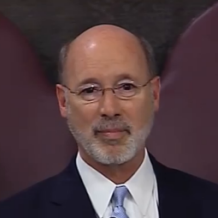




4.8 Dataset

This section describes all the used datasets, see Table 4.4 (target actors) and Table 4.5 (source actors).

4.9 Discussion

While we have demonstrated highly realistic reenactment results in a large variety of applications and scenarios in this chapter, our approach is also subject to a few limitations. Similar to all other learning-based approaches, ours works very well inside the span of the training corpus. Extreme target head poses, such as large rotations, or expressions far outside this span can lead to a degradation of the visual quality of the generated video portrait, see Figure 4.20 and the supplemental video at the project website. Since we only track the face with a parametric model, we cannot actively control

Table 4.4: Target videos: Name and length of sequences (in frames). *Malou* video courtesy of Louisa Malou (CC BY). *May* video courtesy of the UK government (Open Government Licence). *Obama* video courtesy of the White House (public domain). *Putin* video courtesy of the Kremlin (CC BY). *Reagan* video courtesy of the National Archives and Records Administration (public domain). *Elizabeth II* video courtesy of the Governor General of Canada (public domain). *Reagan* video courtesy of the National Archives and Records Administration (public domain). *Wolf* video courtesy of Tom Wolf (CC BY).

				
Ingmar 3,000	Malou 15,000	May 5,000	Obama1 2,000	Obama2 3,613
				
Putin 4,000	Elizabeth II 1,500	Reagan 6,984	Thomas 2,239	Wolf 15,000
				
DB1 8,000	DB2 18,138	DB3 6,500	DB4 30,024	

the motion of the torso or hair, or control the background. The network learns to extrapolate and finds a plausible and consistent upper body and background (including some shadows and reflections) for a given head pose. This limitation could be overcome by also tracking the body and using the underlying body model to generate an extended set of conditioning images. Currently, we are only able to produce medium-resolution output due to memory and training time limitations. The limited output resolution makes it especially difficult to reproduce fine-scale detail, such as individual teeth, in a temporally coherent manner. Yet, recent progress on high-resolution discriminative adversarial networks [Karras et al. 2018; Wang et al. 2017] is promising and could be leveraged to further increase the resolution of the generated output. On a broader scale, and not being a limitation, democratization of advanced high-quality video editing possibilities, offered by our and other methods, calls for additional care in ensuring verifiable video authenticity, e.g., through invisible watermarking.

Table 4.5: Source videos: Name and length of sequences (in frames). *Obama* video courtesy of the White House (public domain).

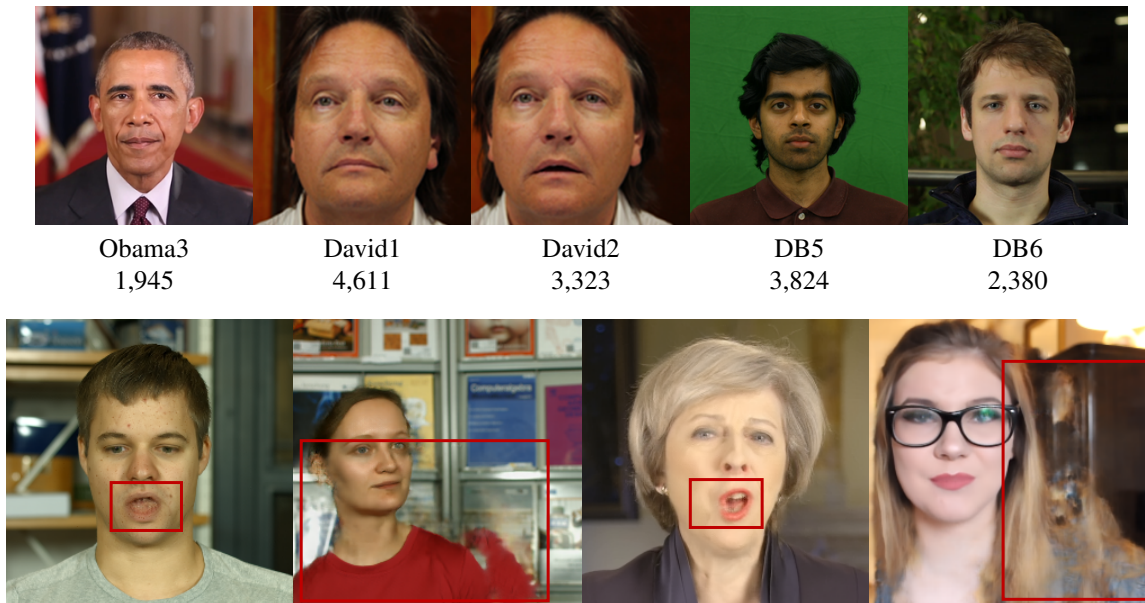


Figure 4.20: Our approach works well within the span of the training corpus. Extreme changes in head pose far outside the training set or strong changes to the facial expression might lead to artifacts in the synthesized video. This is a common limitation of all learning-based approaches. In these cases, artifacts are most prominent outside the face region, as these regions have no conditioning input. *May* video courtesy of the UK government (Open Government Licence). *Malou* video courtesy of Louisa Malou (CC BY).

4.10 Summary

In this chapter, we presented a new approach to synthesize entire photo-real video portraits of target actors in front of general static backgrounds. It is the first to transfer head pose and orientation, face expression, and eye gaze from a source actor to a target actor. The proposed method is based on a novel rendering-to-video translation network that converts a sequence of simple computer graphics renderings into photo-realistic and temporally-coherent video. This mapping is learned based on a novel space-time conditioning volume formulation. We have shown through experiments and a user study that our method outperforms prior work in quality and expands beyond their possibilities. It thus opens up a new level of capabilities in many applications, like video reenactment for virtual reality and telepresence, interactive video editing, and visual dubbing. We see our approach as a step towards highly realistic synthesis of full-frame video content under control of meaningful parameters. We hope that it will inspire future research in this very challenging field.

The algorithmic improvements proposed in this chapter can be considered as a big step towards highly realistic facial manipulation from unconstrained monocular video input, e. g., YouTube and legacy video clips. We anticipate that the method proposed in this chapter will be particularly beneficial for more sophisticated facial editing applications such as visual dubbing in the movie industry and video teleconferencing in consumer applications. To allow various application scenarios such as refocusing, tilt-shift videography and dolly zoom, we often require dynamic lens effects in the face reconstruction

and editing pipelines. Improvements in this direction are presented next in Chapter 5.

Chapter 5

Focus Editing

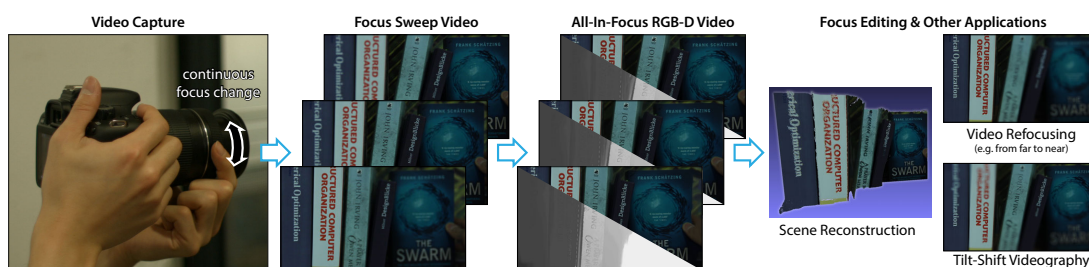


Figure 5.1: We capture focus sweep videos by continuously moving the focus plane across a scene, and then estimate per-frame depth maps and all-in-focus images, i.e., *all-in-focus RGB-D videos*. This enables a wide range of video editing applications, in particular video refocusing.

Many compelling video effects can be performed in post-processing if a video is given in the form of an all-in-focus video with per-frame depth maps and focus distances. In particular, this enables a variety of focus editing effects, such as video refocusing, which are important stylistic elements in video. Recent computational methods that allow to capture such information in an easy and robust manner modify the hardware design of the camera and its optics, or require additional hardware. Hence, they are less practical and unavailable to normal users with commodity cameras. Chapter 5 therefore presents an algorithm to capture all-in-focus RGB-D video of dynamic scenes with commodity video cameras that are unmodified and need no special calibration. Our algorithm turns defocus blur – an effect often regarded as an unwanted artifact – into a valuable signal. The input to our method is a video in which the focus plane is continuously moving back and forth during capture, and thus defocus blur is provoked and strongly visible. This can be achieved by manually turning the focus ring of the lens during recording. The core algorithmic ingredient is a new video-based depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video, and the focus distance for each frame (see Figure 5.1). The method and results presented in this chapter are based on [Kim et al. \(2016\)](#).

5.1 Introduction

A wide range of appealing video effects that would normally require specific camera control during video capture can be algorithmically produced in post-processing if per-frame scene depth is available in addition to color (RGB-D video). In particular, many widely used video effects created by cinematographers involve some form of focus manipulation like depth-of-field control. Examples are the famous dolly-zoom effect, tilt-shift imaging, or general refocusing (‘focus pulling’) to direct the visual attention of the viewer. The ability to control these effects during post-processing is in high demand by professional users and consumers alike, and becomes feasible if depth, all-in-focus video and focus distances at all frames are known. In order to capture video and depth in general, and specifically to enable post-capture focus effects, several methods were proposed that use specialized camera hardware, such as active depth cameras with special illumination [Richardt et al. 2012], light field cameras [Ng et al. 2005], or cameras with optical modifications like coded apertures [Levin et al. 2007].

We follow a different path and propose one of the first end-to-end approaches for depth estimation from – and focus manipulation in – videos captured with an unmodified commodity consumer camera. Our approach turns an often unwanted artifact, *defocus blur*, into a valuable signal. In a camera lens, light falls through an aperture onto the imaging sensor. In theory, smaller apertures produce sharp images for scenes covering a large depth range. When using a larger aperture, only scene points close to a certain focus distance project to a single point on the image sensor, and thus appear in focus. Scene points at other distances are imaged as a *circle of confusion* [Potmesil and Chakravarty 1982]. This produces a limited region of sharp focus around the focus distance that is known as *depth of field*, and outside of which the increasing *defocus blur* provides an important depth cue [Mather 1996].

The problem is that once a video is recorded with specific camera and lens settings, its depth of field is fixed and cannot easily be modified. Changing a focus effect would require recapturing of the scene. The input to our approach is a video with a continuously changing focus in which temporally changing defocus blur is purposefully provoked. This means the focus plane repeatedly sweeps across the scene, e.g., by changing lens focus manually. At first glance, this means each frame has a different unchangeable focus setting, and no frame is entirely in focus. However, we can use the information contained in the blurred video to perform space-time coherent depth, all-in-focus color, and focus distance estimation at each frame. Our approach aligns the input video frames to each other using a new defocus-preserving warping strategy that results in a *focus stack video* with a focus stack at each video frame. We next compute depth maps from each focus stack using a new depth-from-defocus formulation, and *all-in-focus* images by deconvolving the input video frames with the optimal depth-dependent defocus blur kernel at each pixel. Finally, we can render the video with a new depth of field corresponding to arbitrary camera and focus settings, which gives the user complete freedom in refocusing the input video, showing it all in focus, or applying other effects benefiting from RGB-D video.

In a nutshell, the contributions of this chapter are:

- an end-to-end system for depth estimation and focus control of videos captured with a regular

camera in post based on:

- a new hierarchical alignment scheme between video of different focus setting and dynamic scene content,
- an approach to estimate per-frame depth maps and deblurred all-in-focus color image in a space-time coherent way,
- an image-guided algorithm for focus distance initialization,
- and an optimization method for refining focus distances.

We extensively validate the individual steps of our approach and compare against related work. We show high-quality refocusing, dolly-zoom and tilt-shift results on a variety of videos captured with different cameras.

5.2 Related Work

RGB-D Video Acquisition There are many approaches for capturing RGB-D videos, but currently they all require some sort of special hardware or multiple captures from the same or different viewpoints. Active stereo sensors like the original Kinect use structured light, while time-of-flight cameras use time-modulated light, each with a collocated color camera [Richardt et al. 2012]. Moreno-Noguer et al. (2007) show how a projected dot pattern can also be used to infer scene depth from the defocus blur and attenuation of the dots.

Coded apertures enable single-shot RGB-D image capture. Levin et al.’s classic approach (2007) uses a coded aperture optimized to be distinctive at different defocus levels. Bando et al. (2008) use a colored aperture for stereo correspondence between the displaced color channels for estimating depth maps. And concentric circular apertures with different transmission spectra [Chakrabarti and Zickler 2012; Martinello et al. 2015] for example block infrared light in the outermost ring, which results in different depth-of-field in the RGB and IR color channels.

Stereo correspondence [Barron et al. 2015] or multi-view stereo approaches [Yu and Gallup 2014] require multiple views, for instance by exploiting shaky camera motions. Shroff et al. (2012) instead shift the camera’s sensor along the optical axis to change the focus within a video. They align consecutive video frames using optical flow to form a focus stack, and then apply depth from defocus to estimate depth maps and all-in-focus images. Our approach is similar, but unlike all mentioned approaches, it works with a single commodity video camera, and does not require any custom hardware.

There are also single-view techniques based on non-rigid structure-from-motion [Russell et al. 2014], which require clear (in particular out-of-plane) motion cues interpreted under strong scene priors, and learning-based depth estimation techniques [Hoiem et al. 2005; Saxena et al. 2009; Srivastava et al. 2009; Eigen and Fergus 2015].

Depth from Focus/Defocus Focus stacking combines multiple differently focused images into a single *all-in-focus* (or *extended depth of field*) image [Pertuz et al. 2013]. However, focus stacks can also be used for computing depth maps using depth-from-(de)focus techniques, which exploit (de)focus cues within the focus stack. Focus stacking is particularly popular in macro photography, where the large lens magnification of macro lenses results in a very small depth of field. By sweeping the focus plane across a scene or an object, each part of it will be sharpest in one photo, and these sharp regions are then combined into the all-in-focus image. Depth from focus additionally determines the depth of a pixel from the focus setting that produced the sharpest focus [Grossmann 1987; Nayar and Nakagawa 1994]; however, this requires a densely sampled focus stack and a highly textured scene for computing accurate depth maps. Depth from defocus, on the other hand, exploits the varying degree of defocus blur of a scene point in each image for computing depth from just a few defocused images [Pentland 1987; Subbarao and Surya 1994]. The all-in-focus image is then recovered by deconvolving the input images with the spatially-varying point spread function of the defocus blur. It should be obvious that these techniques relying on focus stacks only work well for scenes without camera or scene motion.

Suwajanakorn et al. (2015a) proposed a hybrid approach that stitches an all-in-focus image using motion-compensated focus stacking, and then optimizes for the depth map using depth-from-defocus. Their approach is completely automatic and even estimates camera parameters (up to an inherent affine ambiguity) from the input images. However, their approach is limited to reconstructing a single frame from a focus stack, and cannot easily be extended to videos, as this requires stitching per-frame all-in-focus images. Our approach is tailored for videos, not just single images.

Refocusing Images and Videos Refocusing images – and even videos – has been a long-standing, challenging problem that has seen many creative solution attempts. The defocus blur is an important depth cue that affects the perception of distances and sizes [Mather 1996; Held et al. 2010], so changing the defocus blur by refocusing is a powerful, appearance-altering effect. However, just like RGB-D video capture, all approaches suitable for refocusing videos require some sort of custom hardware, such as special lenses [Ng et al. 2005; Miao et al. 2013], coded apertures [Levin et al. 2007 etc.] or active lighting [Moreno-Noguer et al. 2007], while the remaining approaches suitable for refocusing images are difficult to extend to videos as they rely on multiple captures from the same view, for example, for depth from (de)focus [Suwajanakorn et al. 2015a].

Light field cameras capture a larger subset of the plenoptic function than a normal camera [Ng et al. 2005; Veeraraghavan et al. 2007], which enables virtual refocusing of a single light field image [Isaksen et al. 2000; Ng 2005]. However, capturing multiple views reduces the effective image resolution per view, which severely limits the resolution of refocused images. Miao et al. (2013) instead use a deformable lens between the camera sensor and lens to quickly and repeatedly sweep the focus plane across the scene, and record the resulting video at 120 Hz. They achieve a video refocusing effect by simply selecting appropriate frames from the recorded video.

All remaining image and video refocusing approaches first capture or estimate an all-in-focus RGB-D

image or video, and then virtually refocus it, for example using depth-dependent image blurring (see references in [Lee et al., 2010](#)). However, images captured by a physical camera with a finite aperture only produce a finite depth of field. These images are therefore generally not focused perfectly everywhere, but smaller apertures result in larger depth of field, and so an all-in-focus image is theoretically achieved with a pinhole aperture. Unfortunately, small apertures have poor luminous efficiency as they block most of the light before it can reach the imaging sensor, which results in noisy, poorly exposed images. Some refocusing techniques choose a small aperture that is a reasonable trade-off between defocus blur and imaging noise. The remaining just-noticeable blur can still be detected and used for estimating a depth map [[Shi et al. 2015a](#)], or it can be removed entirely [[Shi et al. 2015b](#)].

Focus information is useful in many applications that go beyond refocusing. [McGuire et al. \(2005\)](#) use differently focused cameras for video matting with general backgrounds. [Bae and Durand \(2007\)](#) magnify the defocus to simulate the shallow depth of field achieved by professional lenses with large apertures. Defocus deblurring is also related to motion deblurring [[Cho et al. 2012/e.g.](#); [Hu et al. 2014](#); [Wulff and Black 2014](#)]; both seek to remove a spatially varying blur. However, defocus and motion blur have different sources and characteristics; motion blur, for example is mostly depth-independent.

5.3 Overview

We developed a practical approach for computing coherent per-frame scene depth, all-in-focus video, and focus distances from video captured with a commodity video camera. [Figure 5.2](#) shows an overview of our approach. Our algorithm enables a variety of compelling focus editing effects during post-processing, and its ability to capture RGB-D video serves a variety of other video effects that require RGB+depth as input. The camera lens model that we use in this chapter has been introduced in [Chapter 2](#).

The input to our approach is a focus sweep video of a static or dynamic scene recorded with a standard video camera. In this video, the focus plane is swept repeatedly across the scene, for example through simple manual focus change on the lens during recording. Each input video frame therefore observes the dynamic scene at a different time and focus distance, as well as a different, purposefully provoked depth of field and defocus blur. We first segment the input video into multiple *focus ramps*, where the focus plane sweeps across the scene in one direction. The first stage of our approach ([Section 5.4.1](#)) then constructs a *focus stack video* for each of them. Focus stack videos consist of a focus stack at each video frame, by aligning adjacent video frames to the current frame using a defocus-preserving warping technique. At each frame, the focus stack video comprises multiple images with a range of approximately known focus distances (see [Section 5.4.5](#)), which are used to estimate a depth map in the second stage ([Section 5.4.2](#)) using depth-from-defocus with filtering-based regularization. The third stage ([Section 5.4.3](#)) performs spatially varying deconvolution to remove the defocus blur and produce all-in-focus images. And the fourth stage of our approach ([Section 5.4.4](#)) further minimizes the remaining error by refining the focus distances for each frame, which significantly improves the

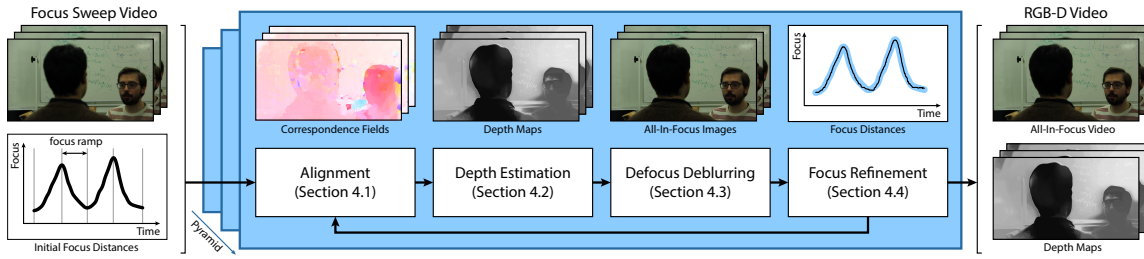


Figure 5.2: Overview of our approach, which estimates spatio-temporally coherent depth maps, all-in-focus images, and focus distances from a focus sweep video. For each video frame, we first align neighboring frames to it to construct a focus stack. We then estimate spatially and temporally consistent depth maps from the focus stacks, and compute all-in-focus images using non-blind deconvolution using the depth map. Finally, we refine the focus distances for all frames. We perform these steps in a coarse-to-fine manner and iterate until convergence.

estimated depth maps and visual quality of the all-in-focus images in the next iteration of our pipeline. Our method requires no sophisticated calibration process for focus distances, which allows it to work robustly in practical scenarios.

Video Recording The input to our approach is a video in which the focus distance is continuously changing, with the resulting change of per-frame depth of field and defocus blur. The user can simply do that by manually adjusting the focus setting of the lens so as to roughly follow a sinusoidal focus distance curve while the camera is recording, and thus to obtain several focus ramps (see Figure 5.2). Ideally, one would like to calibrate the focus distance for each frame of the video, which is not possible. Another option would be to read out the focus distance from the camera for every frame; however, in practice, this is difficult and may require low level modification of the firmware. We use the Magic Lantern¹ software for Canon EOS digital DSLR cameras to record timestamped lens information during video capture at the maximum rate of about 4 Hz. In practice, this unfortunately means that focus distance values are not measured for most frames, the timestamps may not be exactly aligned with the time of frame capture, and the recorded focus distances are quantized and so may also not be fully accurate. Furthermore, people cannot be expected to reproduce an exact curve of focus distance changes, so there is natural variation in the focus distance ramps. Therefore, our algorithm (Section 5.4) uses the sparsely recorded lens information only as a guide and explicitly optimizes for the dense correct focus distances at every frame (Section 5.4.4).

5.4 All-In-Focus RGB-D Video Recovery

Given a video \mathcal{V} with frames $\{\mathbf{V}_t\}_{t \in T}$ containing one or more focus sweeps, we formulate our algorithm as a joint optimization framework that seeks the optimal depth maps \mathbf{D}_t , all-in-focus images \mathbf{I}_t , and focus distances F_t for all video frames $t \in T$. Let us assume that $W_{s \rightarrow t}(\cdot)$ is a warping function that spatially aligns an image at time s with the image at time t while preserving its original defocus

¹<http://www.magiclantern.fm>

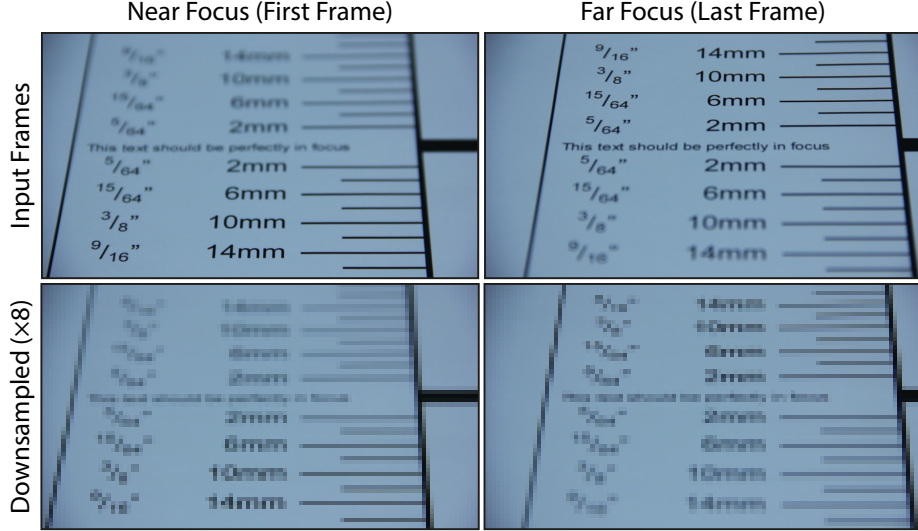


Figure 5.3: Two input frames at full resolution compared to their downsampled versions below. The downsampling effectively reduces the difference in defocus blur, which helps in correspondence finding.

blur (we explain how we compute $W_{s \rightarrow t}$ in Section 5.4.1). Then, we can construct a focus stack at each frame t by warping all input video frames to it using $\{W_{s \rightarrow t}(\mathbf{V}_s)\}_{s \in T}$ (in practice, we only warp a few keyframes, as explained later). We seek the optimal depth map \mathbf{D}_t and all-in-focus image \mathbf{I}_t at frame t , and focus distances $\{F_t\}_{t \in T}$ which best reproduce the focus stack at frame t with the defocus model in Equation 2.4. \mathbf{D}_t , \mathbf{I}_t and \mathbf{F}_t , for $t \in T$, are the unknowns our algorithm needs to compute. Therefore, the core ingredient of our joint optimization of all unknowns is a data term that penalizes the defocus model error of the focus stack at all frames:

$$E_{\text{data}} = \sum_{t \in T} \sum_{s \in T} w_{t,s} \|\Phi(\mathbf{D}_t, F_s) * \mathbf{I}_t - W_{s \rightarrow t}(\mathbf{V}_s)\|^2. \quad (5.1)$$

Here, we introduce a weighting term $w_{t,s}$ to give lower weights to pairs of frames that are further apart, and which hence need warping over longer temporal distances. In our implementation, we use a Gaussian function $w_{t,s} = \exp(-|t-s|^2/2\sigma_w^2)$ with σ_w set to 85 percent of the length of each focus ramp. Simultaneously estimating depth, deblurring the input video and optimizing focus distances from such purposefully defocused and temporally misaligned images is highly challenging, and many widely used matching criteria or invariance assumptions made by traditional correspondence finding approaches break down in this case. To solve this joint optimization problem efficiently, and to cope with the aforementioned challenges, we decompose the optimization for all the unknowns into four subproblems or *stages* that we solve iteratively: defocus-preserving alignment (Section 5.4.1), depth estimation (Section 5.4.2), defocus deblurring (Section 5.4.3), and focus refinement (Section 5.4.4). Each subproblem requires solving for a subset of the unknowns by means of minimizing a cost functional of the form given in Equation 5.1, with additional regularization terms explained in the following. We adopt a multi-scale, coarse-to-fine approach. At each resolution level, we perform three iterations of the four stages, each of which is solved for the entire length of the input video. The multi-resolution approach improves convergence, but more importantly, any focus difference between

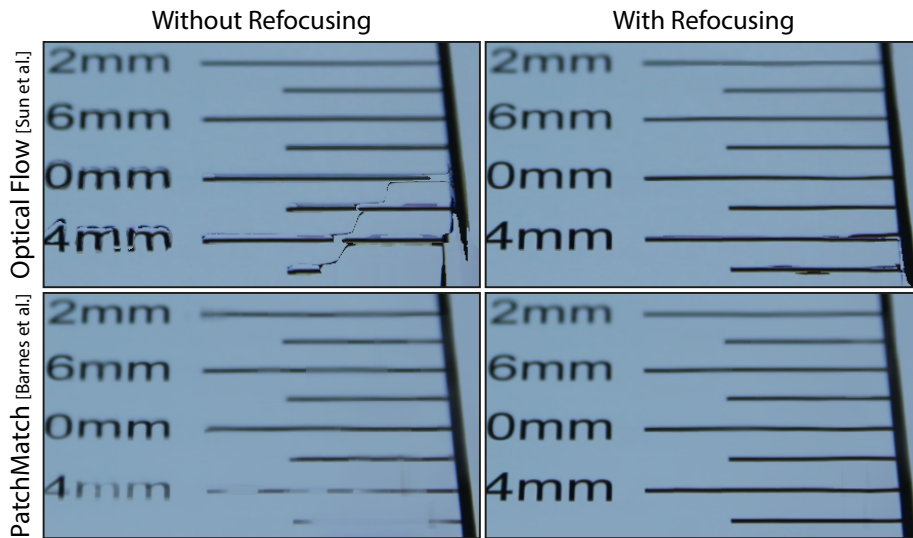


Figure 5.4: Comparison of focus stack alignment approaches, from near to far focus (see Figure 5.3). Without refocusing images to match their blur levels, both optical flow and PatchMatch fail. With refocusing, PatchMatch produces a visually better alignment than optical flow, as used by Shroff et al. (2012).

two video frames is reduced when the images are downsampled in the pyramid (see Figure 5.3). As illustrated in Figure 5.3, this insight enables us to compute reliable initial correspondence fields with less influence from different defocus blurs. Once all parameters are estimated at a coarse level, the higher level of the pyramid uses them as initialization for its iterations. The solutions to these subproblems – henceforth also called *stages* – are explained in detail in the following subsections.

5.4.1 Patch-Based Defocus-Preserving Alignment

The goal in this section is to construct a focus stack for each frame of the input video, which we achieve using patch-based, defocus-preserving image alignment. The result are the warping functions $W_{s \rightarrow t}$ for pairs (s, t) of frames, while all other unknowns (\mathbf{I} , \mathbf{D} and F) are assumed to be constant in this step. Two frames in the focus sweep video, \mathbf{V}_s and \mathbf{V}_t , generally differ in defocus blur and maybe scene or camera motion. The main challenge of the defocus-preserving alignment is therefore to compute a reliable spatial per-pixel correspondence field between them, which is robust to both complex motion as well as defocus changes between the frames.

Using standard correspondence techniques, such as optical flow, to directly warp the input video frame \mathbf{V}_s to \mathbf{V}_t is prone to failure, because the different defocus blurs in the two images are not modeled by standard matching costs. Optical flow will try to explain differences in defocus blur using flow displacements, which produces erroneous correspondences. Figure 5.4 shows an example of how the motion alignment with optical flow fails due to the different defocus blur.

The solution is to compensate any focus differences before computing correspondences [Shroff et al. 2012]. We therefore refocus the target frame \mathbf{V}_t to match the focus distance F_s of the source frame s using



Correspondences for the `BOOK` dataset
(from the first frame to the last frame).

Figure 5.5: Correspondence fields.

the refocusing operator

$$R(\mathbf{V}_t, F_s) = \Phi(\mathbf{D}_t, F_s) * \mathbf{I}_t. \quad (5.2)$$

In the first iteration at the coarsest resolution level, the refocusing operator simply returns the input frame \mathbf{V}_t unchanged, as the downsampling in the pyramid has already removed most of the defocus blur. In subsequent iterations, the refocusing operator uses the current estimates of frame t 's depth map \mathbf{D}_t and all-in-focus image \mathbf{I}_t to perform the refocusing. The embedding of this focus difference compensation and alignment process into the overarching coarse-to-fine scheme enables reliable focus stack alignment even for large scene motions and notable defocus blur differences.

In our framework, we use PatchMatch [Barnes et al. 2009] to robustly compute the warping function $W_{s \rightarrow t}$ between the source frame \mathbf{V}_s and the refocused target frame $R(\mathbf{V}_t, F_s)$. The main benefit of PatchMatch is that it can easily handle complex motions and is fairly robust to the remaining focus differences, while traditional optical flow techniques tend to fail in such cases. PatchMatch correspondences are not always geometrically correct as shown in Figure 5.5, as they exploit visually similar patches from other regions of the image which have similar defocus blur (note the yellow and purple regions on the left, which indicate vertical motion along the edge of the books). In our case, this is an advantage, as it improves the warping quality while preserving defocus blur. At the coarsest level of the pyramid, we initialize the PatchMatch search using optical flow [Sun et al. 2014] between the downsampled input frames \mathbf{V}_t and \mathbf{V}_s ; at this level focus-induced appearance differences are minimal and the flow result can be used as a guide for the finer levels. We also constrain the size of the search window to find the best matches around the initial correspondences. This encourages the estimated correspondence field to be more spatially consistent. Since the warping is computed by refocusing the target frame, the defocus blur in the source frame is preserved, which is crucial for constructing valid focus stacks from a dynamic focus sweep video.

We apply the estimated defocus-preserving warping operators $W_{s \rightarrow t}$ to create a focus stack video with per-frame focus stacks, as illustrated in Figure 5.6. However, we do not warp all frames to all others,

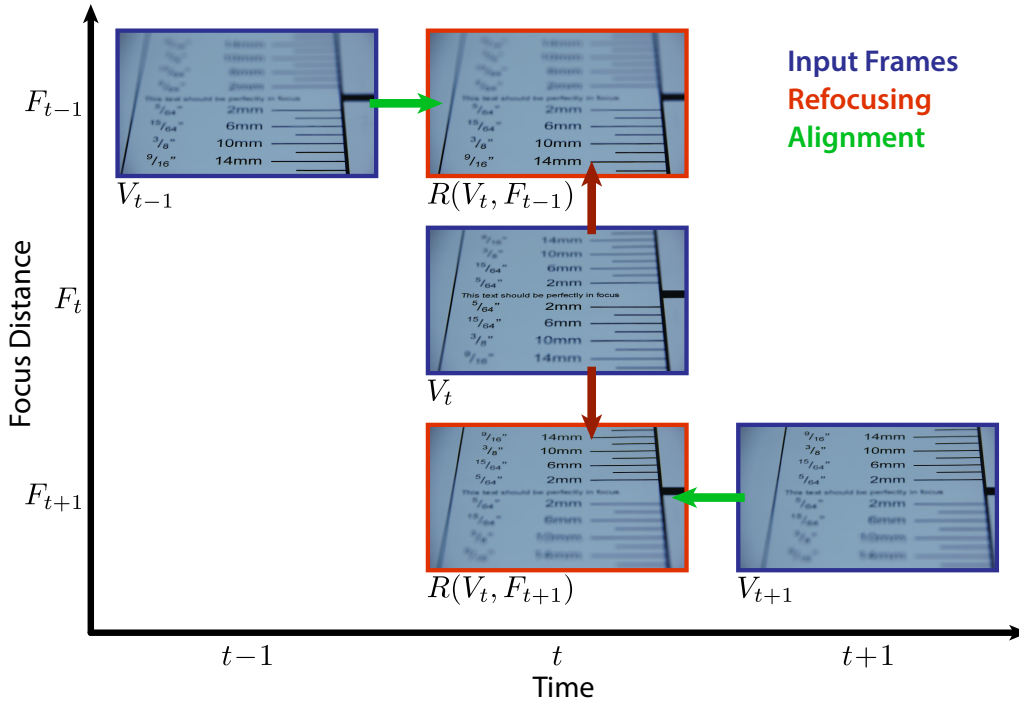


Figure 5.6: Defocus-preserving alignment. We refocus the input frame V_t (center) to match the focus distances of neighboring frames $F_{t\pm 1}$ (red arrows), and then compute correspondences (green arrows) between the neighboring frames, $V_{t\pm 1}$, and the corresponding refocused image $R(V_t, F_{t\pm 1})$.

to prevent artifacts that may be introduced by aligning temporally distant videos frames in which the scene may have drastically changed. Instead, we first segment the input video into one or more contiguous focus ramps (see Figure 5.2), $T_i \subset T$ for $i \in R$, which contain only temporally close frames. For each input video frame t , we then create a focus stack by warping the other frames in its ramp to it using our defocus-preserving alignment. This reduces the computational complexity of the alignment stage from $O(|T|^2)$ for warping all pairs of frames, to $O(|T|^2/|R|)$ for warping all pairs of frames within each ramp to each other.

5.4.2 Filtering-Based Depth Estimation

In the second stage of our approach, we estimate spatially and temporally consistent depth maps \mathbf{D}_t for all focus stacks, while keeping all other variables constant. At this point, we assume that the focus distances F_t for each frame t are known, either from the initialization (Section 5.4.5) or from focus refinement in the previous iteration of our optimization (Section 5.4.4). In this case, our data term E_{data} from Equation 5.1 measures how well the estimated depth maps fit to the defocus observations in all focus stacks, which is equivalent to depth from defocus [Pentland 1987], applied per video frame.

Depth from defocus computes depth maps from differently focused images with given focus distances by optimizing for matching defocus blur in all images. This step requires the pixel-wise alignment across each focus stack that we computed in the previous stage, to measure the fitting error. Since this error is individually penalized at each pixel, it can lead to spatial inconsistencies in the depth map.

To avoid this issue, we introduce a long-range linear Potts model. In contrast to the pairwise Potts model which compares depth values only between immediately adjacent pixels, our version performs long-range comparisons which benefit globally consistent depth estimation, yet prevents erroneous smoothing of actual features in the depth map:

$$E_{\text{smoothness}}^{\text{spatial}} = \sum_{t \in T} \sum_{\mathbf{x}} \sum_{\mathbf{y} \neq \mathbf{x}} \min(\alpha(\mathbf{x}, \mathbf{y}) |\mathbf{D}_t(\mathbf{x}) - \mathbf{D}_t(\mathbf{y})|, \tau_d), \quad (5.3)$$

where τ_d is the truncation value of the depth difference. We use the bilateral weight α between two pixels \mathbf{x} and \mathbf{y} ,

$$\alpha(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma_s^2} - \frac{\|\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})\|^2}{2\sigma_r^2}\right), \quad (5.4)$$

to encourage consistent depth estimation between nearby pixels with similar colors, where σ_s and σ_r denote the standard deviation for the spatial and range terms, respectively. We use $\sigma_s = 0.075 \times$ the image width, and $\sigma_r = 0.05$.

In addition, we want the depth maps to be temporally coherent across all frames. We minimize the discrepancy between depth maps using

$$E_{\text{smoothness}}^{\text{temporal}} = \sum_{t \in T} \sum_{s \in T \setminus t} \|\mathbf{D}_t - W_{s \rightarrow t}(\mathbf{D}_s)\|^2, \quad (5.5)$$

which encourages temporal consistency over extended depth map sequences. The total cost function for depth map estimation is defined by combining the data term from Equation 5.1 and the terms from Equations 5.3 and 5.5:

$$\underset{D}{\operatorname{argmin}} E_{\text{data}} + \lambda_{\text{ss}} E_{\text{smoothness}}^{\text{spatial}} + \lambda_{\text{ts}} E_{\text{smoothness}}^{\text{temporal}}, \quad (5.6)$$

where $\lambda_{\text{ss}} = 1$ and $\lambda_{\text{ts}} = 0.2$ are balancing weights.

The direct minimization of Equation 5.6 requires global optimization with respect to all depth images, which is computationally expensive; instead, we solve an efficient approximation of the global optimization problem. We pose the minimization task as a labeling problem, and first estimate spatially consistent depth maps for all frames by applying the filtering-based inference approach by [Krahenbuhl and Koltun \(2011\)](#), and then refine the per-frame depth computation to enforce temporal consistency [[Lang et al. 2012](#)].

We start by computing per-frame depth maps \mathbf{D}_t in three steps. First, we evaluate the data term (Equation 5.1) for a range of n pre-defined, uniformly spaced depth layers, and store the error for each pixel \mathbf{x} and depth label d in the cost volume $\mathbf{C}(\mathbf{x}, d)$. As in previous depth-from-defocus techniques [[Pentland 1987](#)], we perform this evaluation in the frequency domain, where convolution can be efficiently computed using element-wise multiplication. In the second step, we apply fast joint-bilateral filtering

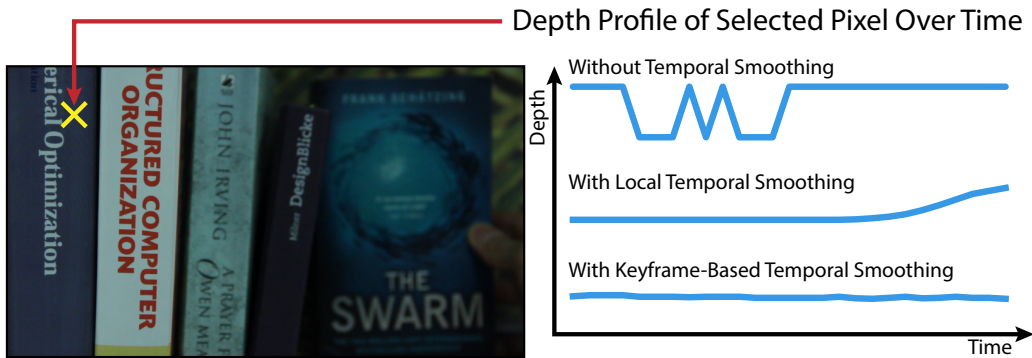


Figure 5.7: Comparison of temporal smoothing approaches. Our keyframe-based smoothing approach produces temporally more consistent depth than local smoothing of adjacent frames. (Note that the considered pixel should have constant depth in this scene.)

[Paris and Durand 2009] on each depth-cost slice individually, to minimize the long-range spatial smoothness term in Equation 5.3. We use the all-in-focus image \mathbf{I}_t as the guide image for computing the range term of the bilateral weight α in Equation 5.4. As in the previous section, we take the estimated all-in-focus image I_t from the previous iteration, and assume $\mathbf{I}_t = \mathbf{V}_t$ in the first iteration of the coarsest resolution level. This process is similar to the message-update step in the filtering-based inference approach by Krahenbuhl and Koltun (2011). In the third step, we select the spatially optimal depth for each pixel using $\mathbf{D}_t(\mathbf{x}) = \operatorname{argmin}_d \mathbf{C}(\mathbf{x}, d)$.

After computing depth maps independently from each focus stack, we apply temporal smoothing to make the depth maps consistent over time. For efficiency, we use a keyframe-based approach with a sliding temporal window. For each frame t , we align the depth maps of the previous and following two keyframes to the current depth map \mathbf{D}_t using our warping operator $W_{s \rightarrow t}$ computed on the all-in-focus images. The updated depth map \mathbf{D}_t is the Gaussian-weighted mean of aligned depth maps, with higher weight given to temporally closer frames. The used keyframes are not restricted to be from the same focus ramp as the frame t , so that temporal consistency is also enforced across focus ramp boundaries. In Figure 5.7, we show that our approach successfully produces temporally coherent depth maps, compared to the unfiltered input depth maps and also the simpler local filtering of immediately adjacent frames, as some bias remains due to the short temporal range of the filtering.

5.4.3 Defocus Deblurring

Now that we have computed the depth maps \mathbf{D}_t , we estimate the all-in-focus images \mathbf{I}_t using non-blind deconvolution with the estimated, spatially varying point spread function (PSF) corresponding to the depth-dependent circle of confusion (Equation 2.3). While the disc shape of the PSF is a good approximation of the actual shape of the camera aperture, in practice, its sharp boundary causes ringing artifacts in the deconvolution process due to zero-crossings in the frequency domain [Levin et al. 2007], see Figure 5.8. We therefore adopt the smoothness term introduced by Zhou et al. (2011) to prevent

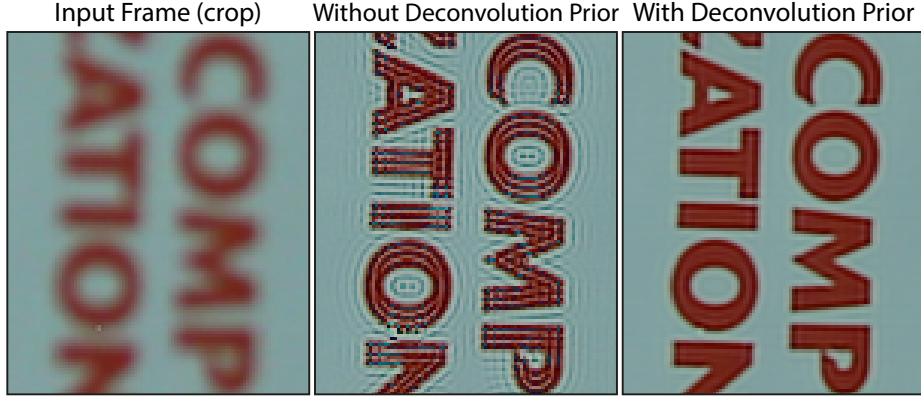


Figure 5.8: Deconvolution can result in ringing artifacts, which are successfully suppressed by our deconvolution smoothness prior.

ringing artifacts:

$$E_{\text{smoothness}}^{\text{all-in-focus}} = \sum_{t \in T} \|\mathbf{H} * \mathbf{I}_t\|^2, \quad (5.7)$$

where \mathbf{H} is an image statistics prior.

The main idea of the smoothness term is to exploit a learning approach to capture natural image statistics. We first take sample images at the same resolution as the input image from a database of natural images, and then apply the Fourier transform to the samples in order to compute the frequency distribution of the image statistics. The final image statistics of the natural image dataset \mathbf{H} is obtained by averaging the squared per-frequency modulus of all sample distributions. The smoothness term Equation 5.7 enforces the all-in-focus image \mathbf{I}_t to follow a similar frequency distribution as the learned one in \mathbf{H} . The image statistics prior \mathbf{H} only needs to be computed once for each video resolution to be processed, and can then be reused for new videos. For the details of the smoothness term, we refer the reader to Zhou et al. (2011).

The total cost function of the defocus deblurring is expressed as a combination of the data term in Equation 5.1 and the learned smoothness term in Equation 5.7:

$$\underset{I}{\operatorname{argmin}} E_{\text{data}} + \lambda_{\text{as}} E_{\text{smoothness}}^{\text{all-in-focus}}, \quad (5.8)$$

where $\lambda_{\text{as}} = 10^{-3}$ balances the two cost terms. We compute the optimal all-in-focus image \mathbf{I}_t by performing Wiener deconvolution independently on a range of n depth layers, each with a fixed, depth-dependent point spread function, and then composite the sub-images to obtain the all-in-focus image \mathbf{I}_t .

5.4.4 Focus Distance Refinement

Equations 5.6 and 5.8 solve for the optimal depth maps \mathbf{D} and all-in-focus images \mathbf{I} for the given focus distances F , and this step further refines the focus distances to reduce the cost functional E_{data} (Equation 5.1). As explained in Section 5.3, we can at best read out temporally sparse focus distance

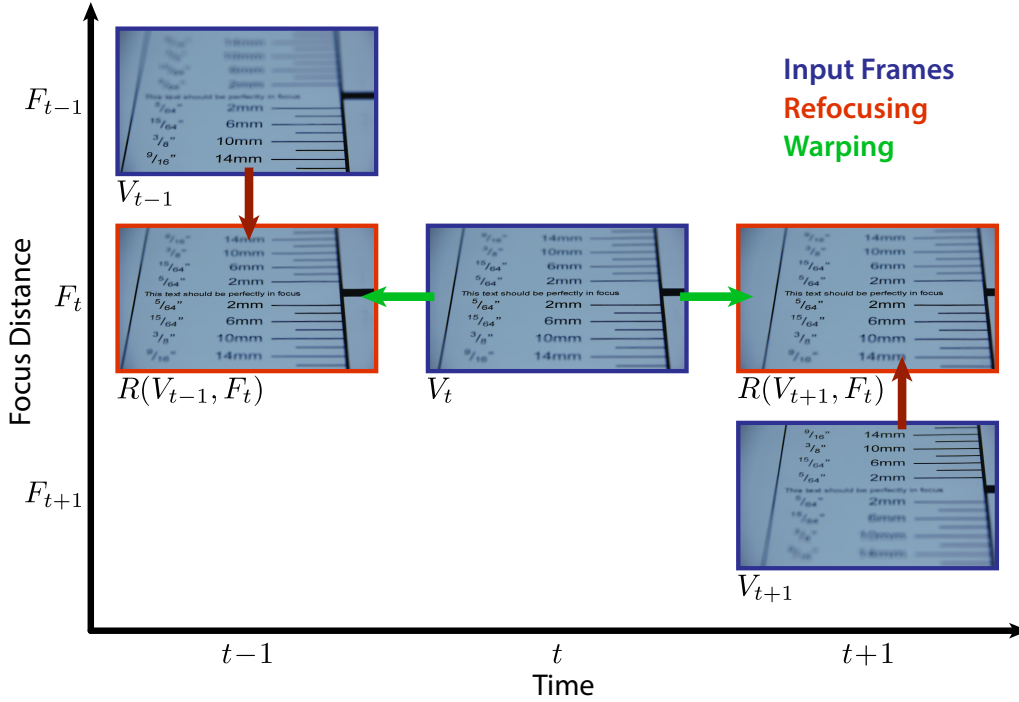


Figure 5.9: We refine focus distances by refocusing input frames to each frame t , and minimizing the difference to the frame V_t warped to each of the refocused input images (Equation 5.9).

values from the camera which are moreover subject to inaccuracies. To overcome this difficulty, we refine the focus distances for all frames in the final stage of our approach. By rearranging the terms associated with the focus distance F_t in Equation 5.1, we define the focus refinement subproblem as:

$$\operatorname{argmin}_{F_t} \sum_{s \in T} w_{s,t} \|R(\mathbf{V}_s, F_t) - W_{t \rightarrow s}(\mathbf{V}_t)\|^2, \quad (5.9)$$

where $w_{s,t}$ is the same weighting parameter as in Equation 5.1. This equation minimizes the defocus model error (Equation 2.4) at all frames by finding the optimal focus distance F_t .

We optimize Equation 5.9 by gradient descent. Since the cost function has a highly nonlinear form in terms of F_t , we compute the gradient numerically by examining the costs for focus distance $F_t \pm \delta$ with $\delta = 5$ mm. This process in practice refocuses each source frame V_s to the focus distance $F_t \pm \delta$, and compares it to the aligned target frame $W_{t \rightarrow s}(\mathbf{V}_t)$ (see Figure 5.9). We then update the focus distance F_t to the value corresponding to the new minimum cost. This computation can be performed independently for each target frame t . Once the focus distances F of all frames are refined, we use them in the next iteration of our algorithm.

We demonstrate the performance of our focus distance refinement in Figure 5.10. It improves the depth estimation as well as visual quality of the all-in-focus images by suppressing excessive edge contrasts. Because this strategy frees us from requiring artificial patterns or special hardware for the accurate calibration of focus distances, it allows for our flexible and simple acquisition of the focus sweep video.

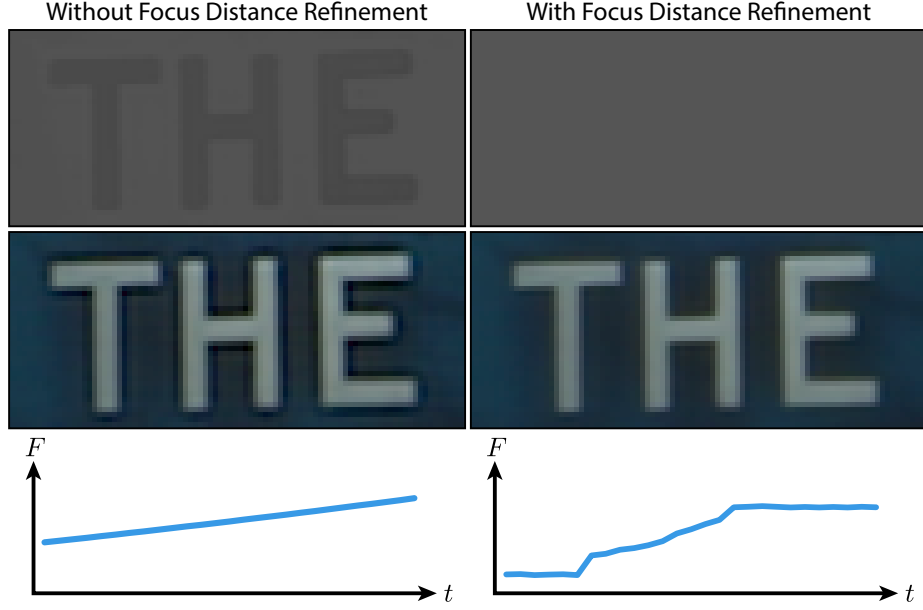


Figure 5.10: Focus distance refinement improves depth maps (top) by reducing texture copy artifacts. This also removes halos in the all-in-focus images (middle). The refined focus distances on the right also correctly reflect that the focus distance was in fact kept constant for a few frames at the beginning and end of the video.

5.4.5 Initialization and Implementation

The input to our video refocusing approach is a radiometrically linearized video with temporally changing focus distances containing one or more focus ramps, but with otherwise constant camera settings. We assume that we know camera properties such as the focal length, aperture f -number, sensor size, as well as temporally sparse readings of the camera’s focus distances for some video frames.

Focus Distance Initialization Before the start of our algorithm in Section 5.4, we compute an initial set of temporally dense focus distance values. We use the sparse timestamped focus distance readings from the Magic Lantern firmware as starting point, see Section 5.3. We then solve for the per-frame focus distances F using an energy minimization with the recorded focus data as data term, and additional smoothness and focus-consistency regularization terms:

$$\operatorname{argmin}_F E_{\text{data}}^{\text{focus}} + \lambda_{\text{fs}} E_{\text{smoothness}}^{\text{focus}} + \lambda_{\text{focus}} E_{\text{focus}}. \quad (5.10)$$

The recorded focus distances F_t^{rec} are available only for some frames $t \in T_{\text{rec}}$, so we constrain the unknown focus distances F_t at those frames to lie close to them:

$$E_{\text{data}}^{\text{focus}} = \sum_{t \in T_{\text{rec}}} \|F_t - F_t^{\text{rec}}\|^2. \quad (5.11)$$

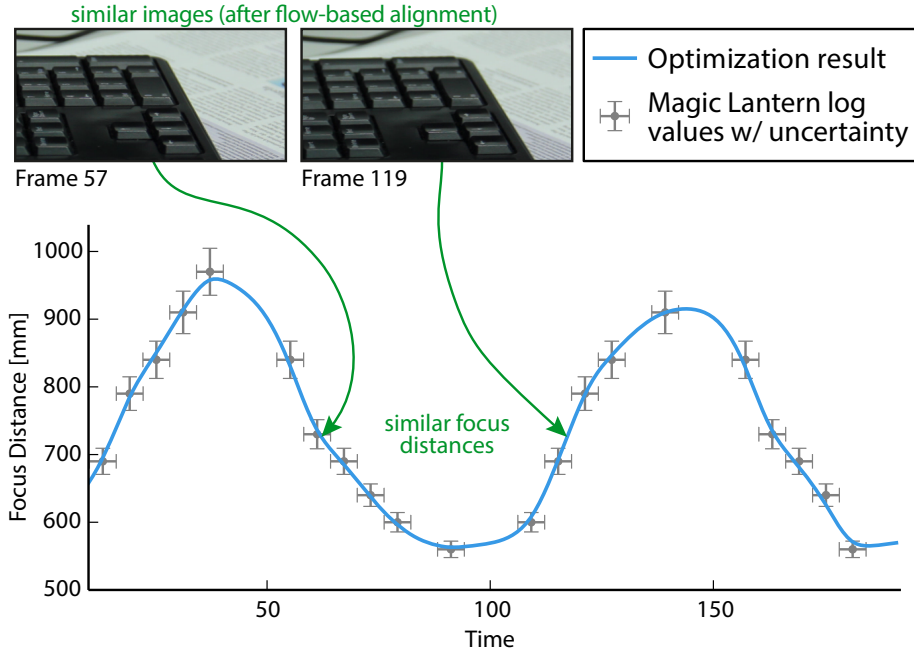


Figure 5.11: Focus distance initialization yields a smooth initial focus curve from sparse Magic Lantern data. Similar images according to $s_{t,s}$ (Equation 5.14) enforce consistent focus distances.

As the focus is assumed to change smoothly over time, we enforce this by penalizing the second derivative of the focus distances:

$$E_{\text{smoothness}}^{\text{focus}} = \sum_t \|F_{t-1} - 2F_t + F_{t+1}\|^2. \quad (5.12)$$

The focus-consistency term exploits the observation that similar focus distances result in similar depth-of-field and hence similar images, so if video frames appear very similar, then their focus distances should also be similar (see Figure 5.11):

$$E_{\text{focus}} = \sum_t \sum_{s \neq t} s_{t,s} \|F_t - F_s\|^2, \quad (5.13)$$

where $s_{t,s}$ measures the (symmetric) similarity of the input video frames V_t and V_s , so that more similar frames enforce consistency constraints more strongly. We compute the similarity using

$$s_{t,s} = \min(0, 1 - \min(d_{t,s}, d_{s,t}) / \tau_{\text{sim}}) \quad (5.14)$$

based on the image dissimilarity $d_{t,s}$ which we compute using the RMSE between input frame V_t and V_s warped to V_t using low-resolution (160×90) optical flow to compensate for camera and scene motion. The similarity threshold τ_{sim} determines which pairs of input frames result in consistency constraints and how strongly they are enforced. Typical parameter values are $\lambda_{\text{fs}} = \sqrt{10}$, $\lambda_{\text{focus}} = 0.1$ and $\tau_{\text{sim}} \in [0.01, 0.05]$.

Implementation of Video Depth-From-Defocus Algorithm To implement the method from Section 5.4, we use a multi-resolution approach with three levels to improve the convergence and visual quality of our results, as image defocus is more similar at coarser image resolutions. At each pyramid level, we perform three iterations of the stages described in Section 5.4.1, Section 5.4.2, Section 5.4.3 and Section 5.4.4. At the coarsest level, we start the first iteration assuming that the all-in-focus image I_t is the input video frame V_t , as mentioned earlier, and also initialize our PatchMatch correspondences using optical flow to provide a good starting point for our alignment computations. Between pyramid levels, we bilinearly upsample the all-in-focus images I_t , depth maps D_t and all computed flow fields. We use scale-adjusted patch sizes for PatchMatch, using 25×25 pixels at the finest level and 7×7 at the coarsest.

Computation Times Our all-in-focus RGB-D video estimation approach processes 30 video frames at 854×480 resolution in 4 hours on a 30-core 2.8 GHz processor with 256 GB memory. This runtime breaks down as follows, per video frame: 8.6 minutes for defocus-preserving alignment, 19 minutes for depth estimation, 2.4 minutes for defocus deblurring, and 5 seconds for focus distance refinement. Our MATLAB implementation is unoptimized, but parallelized over the input video frames. We believe an optimized, possibly GPU-assisted implementation would yield significant speed-ups.

5.5 Results and Evaluation

Here, we thoroughly evaluate our proposed video depth-from-defocus approach for reconstructing all-in-focus RGB-D videos. We first show qualitative results on natural, dynamic scenes with non-trivial motion, captured with static and moving video cameras. We then compare our approach against the two closest approaches, by Shroff et al. (2012) and Suwajanakorn et al. (2015a). We further evaluate the design choices made in our processing approach with an ablation study on a ground-truth dataset. Lastly, we evaluate the benefit of our focus refinement optimization.

We show all-in-focus images and depth map results on a range of datasets in Figure 5.12, and in our video. Our depth maps capture the gist of each scene, including the main depth layers and their silhouettes, and the depth gradients of slanted planes with sufficient texture. As shown in the results, our approach works for dynamic scenes, and handles a fair degree of occlusions, dis-occlusions and out-of-plane motions. It also properly reconstructs the depth and all-in-focus appearance of small objects, like the earrings in sequence TALKING2 (Figure 5.12), which is highly challenging. Note that our approach also works if scene and camera are rather static, where approaches requiring notable disparity for depth estimation, would fail, even on unblurred footage. Similar to previous depth-from-(de)focus techniques, our approach works best for textured scenes that are captured in a full focus stack. Although our depth maps are not perfect, they are temporally coherent and enable visually plausible refocusing results, as we show in Section 5.6.

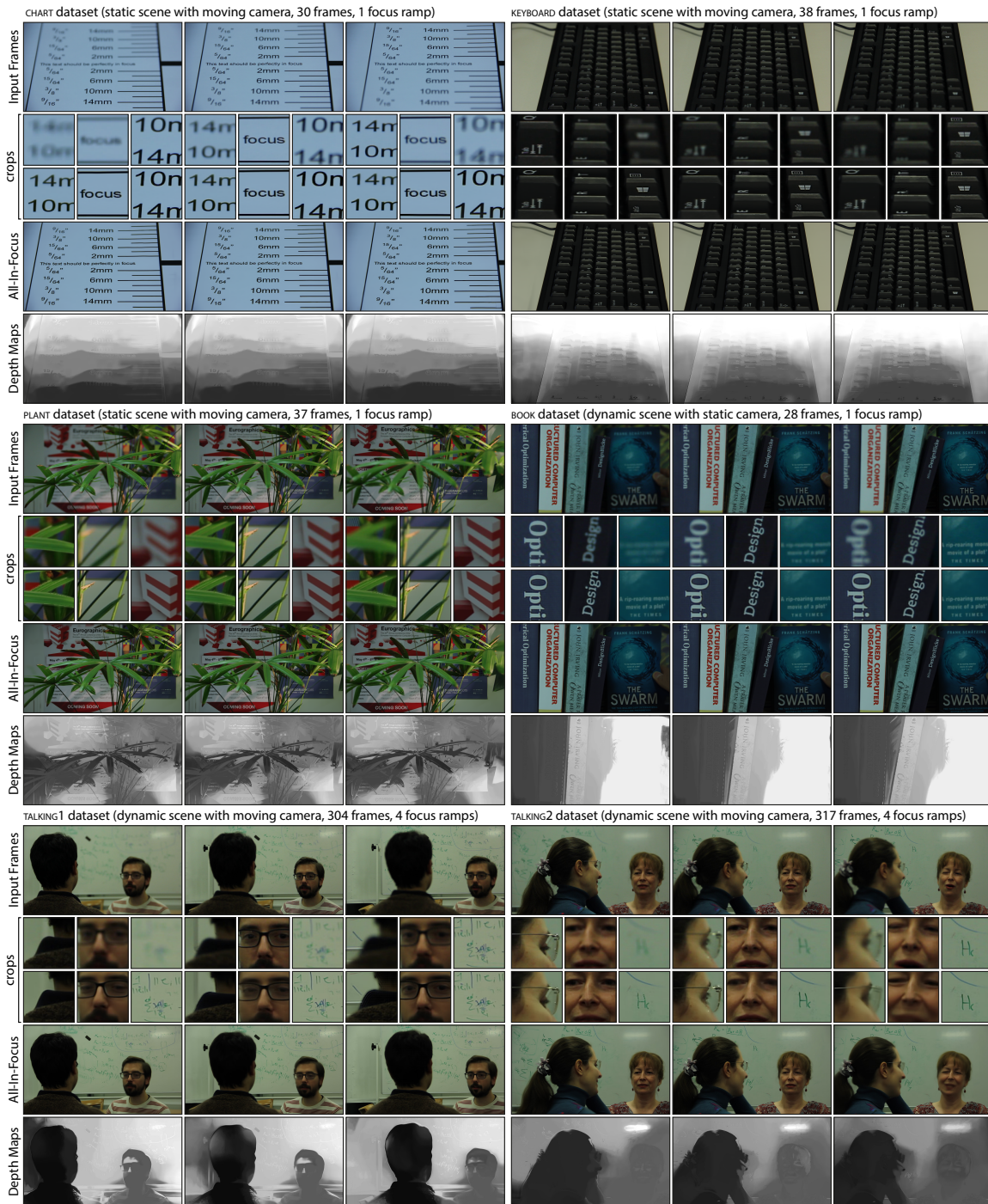


Figure 5.12: RGB-D video results. We show reconstructed all-in-focus images and depth maps for six focus sweep videos with various combinations of scene and camera motion. The image crops (top: input frame cropped, bottom: all-in-focus images cropped) focus on regions at the near, middle and far end (from left to right) of the scene’s depth range. Note that each input frame is in focus in only one of the three crops, while our all-in-focus images are in focus everywhere.

Comparison to Shroff et al. (2012) This work moves a camera’s sensor along the optical axis to compute all-in-focus RGB-D videos in an approach similar to ours. However, our approach improves on theirs in several important ways: (1) we use a commodity consumer video camera that does not require any hardware modifications like in their approach, (2) our defocus-preserving alignment finds more reliable correspondences than optical flow, (3) our depth maps are more detailed and temporally coherent, and (4) our all-in-focus images and hence refocusing results improve on theirs.

We simulate their focus stack alignment approach by replacing PatchMatch in our implementation with optical flow [Sun et al. 2014]. Figure 5.4 shows that our approach for defocus-preserving alignment achieves a visually better alignment result than using optical flow. This demonstrates that PatchMatch is more suitable for finding reliable correspondences, in particular for complex or fast motions that are common in practical scenarios. Kalantari et al. (2013) showed that a similar technique is also effective for robust correspondence finding in dynamic video for HDR video reconstruction.

Comparison to [Suwajanakorn et al. 2015a] This recent depth-from-focus technique targeted for mobile phones computes a *single* depth map with all-in-focus image from quick focus sweeps of around 30 photos of static scenes with little camera motion. Their approach first reconstructs the all-in-focus image by aligning the input photos and stitching together the sharpest regions. This will fail for videos, as dynamic scenes break their alignment strategy of concatenating the optical flows between pairs of adjacent photos. Dynamic scenes also have occlusions and disocclusions that can cause stitching artifacts in the per-frame all-in-focus images. Finally, any estimated per-frame depth maps are most likely not temporally coherent. Our approach, on the other hand, is designed to compute temporally coherent all-in-focus RGB-D videos of dynamic scenes with larger camera motions. It is our robust defocus-preserving alignment (Section 5.4.1) that enables us to construct per-frame focus stacks for dynamic scenes (moving scene and camera), and hence to compute per-frame depth maps and all-in-focus images. On top, we implement keyframe-based temporal consistency filtering (Section 5.4.2) to remove any flickering and jitter from the resulting depth maps. We visually compare the results of our approach to Suwajanakorn et al.’s approach on one of their datasets in Figure 5.13, and show additional comparisons in our video. In our approach, we use their provided camera parameters without further focus distance refinement (Section 5.4.4).

Validation of Design Choices We performed a quantitative ablation study to analyze the influence of the design choices in our algorithm. For this, we synthetically defocus 10 frames from the MPI-Sintel dataset ‘alley_1’ [Butler et al. 2012] using two focus ramps, and apply additive Gaussian noise with $\sigma = 3/255$ to simulate camera imaging noise. We then process the resulting video using our framework while disabling or replacing individual components of our approach. In Figure 5.14, we evaluate the accuracy of the estimated depth maps and all-in-focus images using the root-mean-squared error (RMSE) compared to the ground truth. Our full approach produces overall the best results. One can clearly see the importance of each component in our approach, as leaving them out significantly degrades the quality of the estimated depth maps or all-in-focus images, or both. We also evaluate how accurately each alterna-

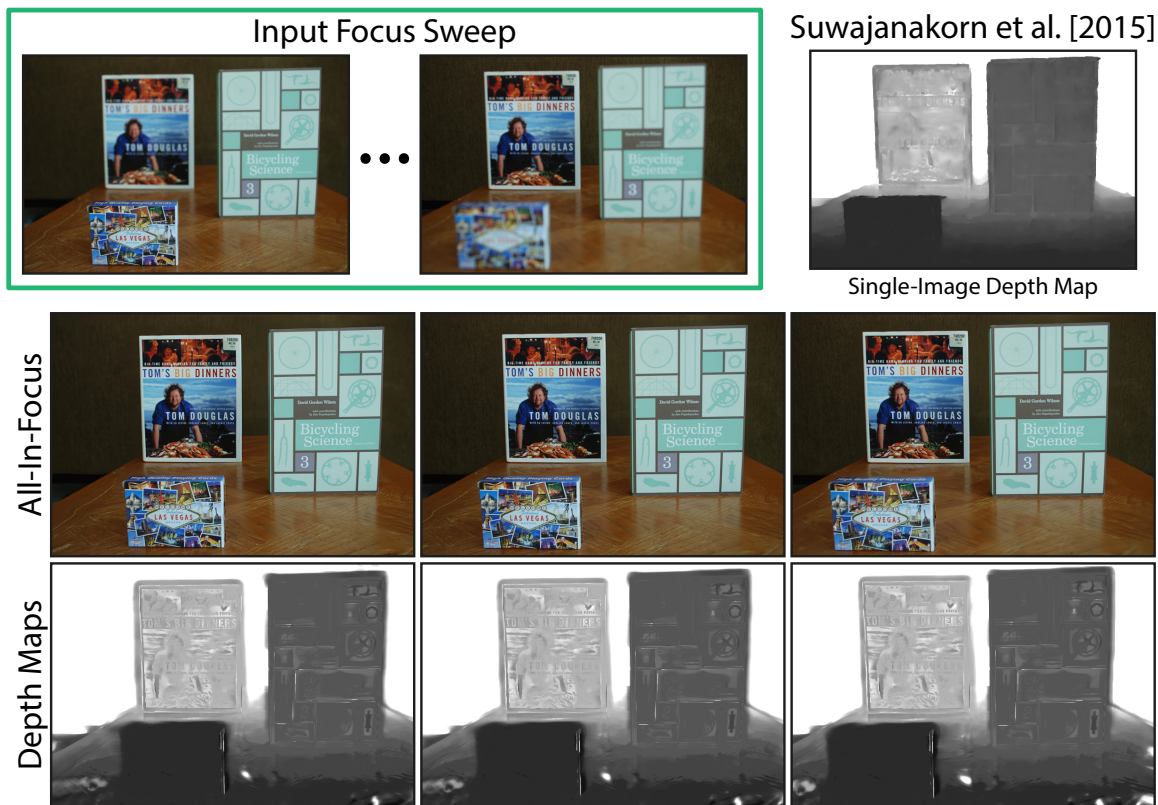


Figure 5.13: Comparison of our approach to Suwajanakorn et al.'s (2015a) on their dynamic dataset. Top: Input focus stack, focused near (left) to far (right), and Suwajanakorn et al.'s estimated depth map for the last frame only. Bottom: We reconstruct all-in-focus images and depth maps for all frames of this dynamic sequence.

tive explains the input defocus images when refocusing the all-in-focus image using the estimated depth map. Only disabling the temporal smoothing produces refocused images with lower RMSE than our full approach, but the images completely lack temporal consistency, which is not measured by RMSE.

Focus Distance Refinement Here, we investigate the contribution of the focus distance refinement (Section 5.4.4) to estimating better all-in-focus images and recovering from inaccurate initial focus distances. For this, we process the synthetically refocused ‘alley_1’ dataset with initial focus distances perturbed by varying degrees of additive Gaussian noise (but without imaging noise), with and without our focus distance refinement, and compare the all-in-focus images and estimated focus distances to the ground truth. Figure 5.15 shows that our focus distance refinement consistently reduces the errors in estimated focus distances. This in turn leads to better refocusing results for our defocus-preserving alignment (Section 5.4.1), which produces cleaner all-in-focus images and improves the overall performance of our approach.

Limitations Our approach relies on aligning all frames of a focus ramp to each other. This works well for focus ramps of up to around 30 frames, but it becomes more difficult for long ramps of around 100 frames, as more motion can happen in that time, which hence needs to be compensated. This is significantly more difficult than for example the alignment required for HDR video reconstruction

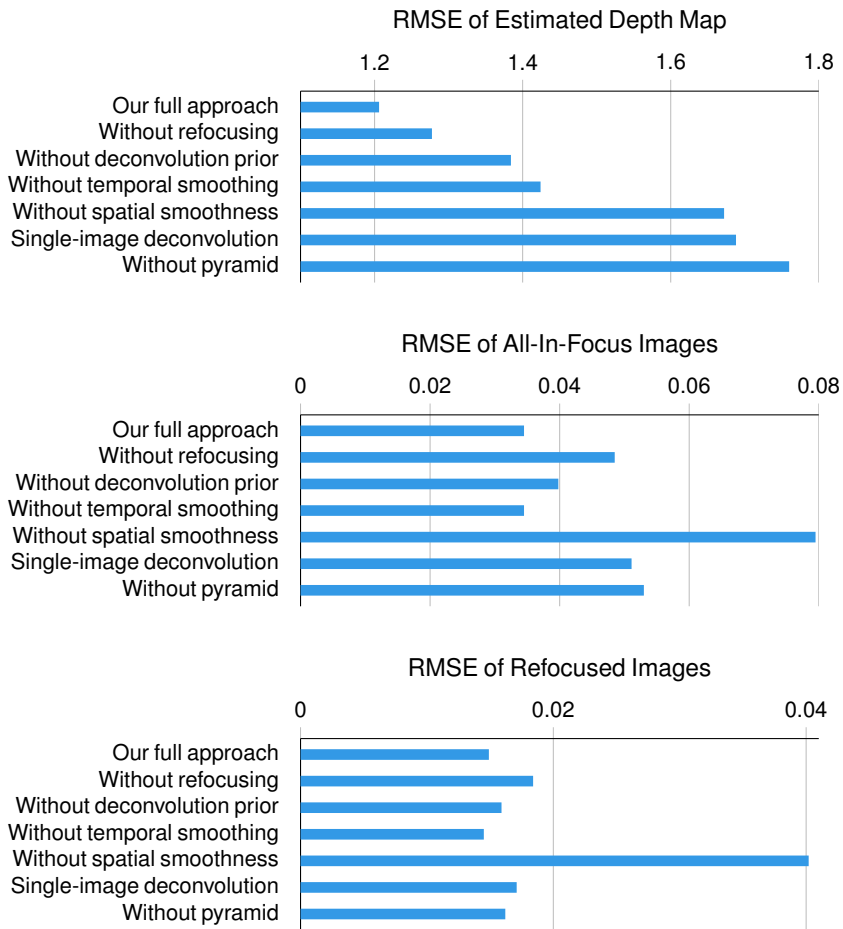


Figure 5.14: Validation of design choices for our video depth-from-defocus approach using an ablation study (lower RMSE is better). Our approach is best overall, but each components is required for achieving the best results.

[Kalantari et al. 2013], which only needs to align three subsequent frames instead of 30–100. While our alignment approach produces good results within a ramp, even for long ramps, the consistency between ramps becomes more difficult to enforce at the boundary between long focus ramps. This may sometimes lead to subtle popping artifacts in the all-in-focus video for our long sequences TALKING1 and TALKING2.

Large occlusions are also problematic as the focus stack alignment degrades in quality when part of the scene is not visible during a focus ramp, for example at the image boundaries. Similar to depth-from-defocus methods for static scenes, we also assume that the appearance of objects remains constant, at least within a certain time window, and in particular that lighting is not drastically changing in the same timespan. Additionally, untextured regions are harder to reconstruct than textured regions, and may show some temporal flickering, similar to previous depth-from-defocus methods but also passive image-based depth reconstruction approaches in general.

We employ a comparably simple defocus blur model, which uses a spatially varying convolution with a point spread function. This approach degrades near depth boundaries, as it may blur across them,

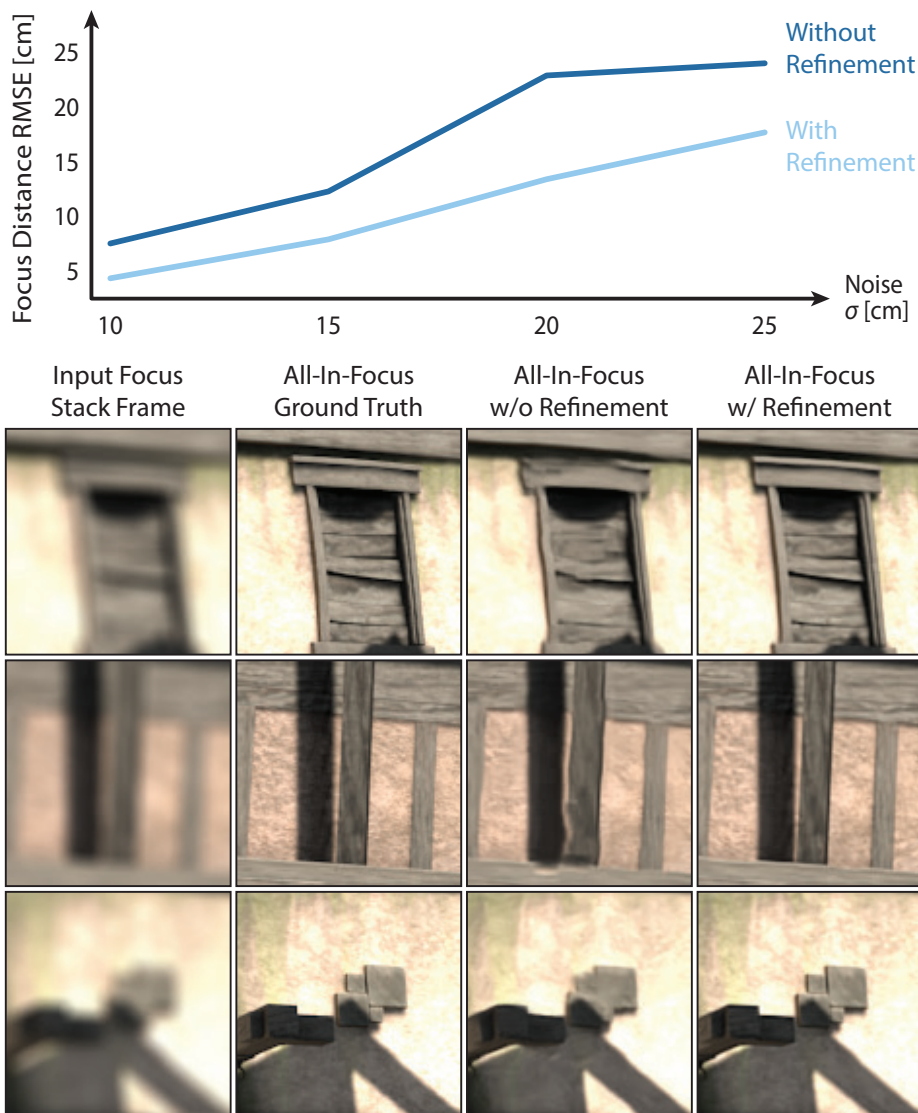


Figure 5.15: Focus distance refinement improves the focus estimates and all-in-focus images when the initial focus distances are noisy. Top: Plot of noise level versus RMSE of focus distances compared to the ground truth; note that refinement consistently reduces the error. Bottom: Crops of a single frame for noise level $\sigma = 10$ cm. Without refinement, the all-in-focus images are distorted and lack details, with refinement, the image is close to the ground truth.

which can create halos in the depth maps (see discussion in [Lee et al., 2010](#)). A possible solution are more sophisticated, multi-layer defocus blur models [[Kraus and Strengert 2007](#)], but they are harder to integrate into our optimization. Our depth maps are plausible, and enable video focus post-processing at very good quality using a standard video camera. But they may not match the quality of depth maps obtained with specialized RGB-D cameras.

5.6 Applications

Video Refocusing Given the all-in-focus images and depth maps estimated in Section 5.4, we can now freely refocus the original input video according to the user’s wishes by simply rendering the appropriate defocus blur in a post-process. For this, we use the same thin-lens defocus model as in Section 5.3, and blur each pixel’s neighborhood with the blur kernel $K(D(\mathbf{x}), F)$ corresponding to its depth $D(\mathbf{x})$ and the focus distance F of the virtual lens [Riguer et al. 2003]. This approach provides complete freedom, as the camera’s aperture, focal length and focus distance can be changed independently and arbitrarily. The user can for example change the aperture, while keeping the original focus settings, to reduce or magnify the defocus blur (see Figure 5.16), similar to Bae and Durand (2007), but for videos. The focus can also be fixed on an object of interest or follow it through the video using a ‘focus pull’, or the focus can be interactively controlled by the user using a ‘focus-follow’ function that keeps the region under the user’s mouse pointer in focus. The reconstructed focus settings can also be smoothed to correct auto-focus failures and produce a more professional-looking result.

Tilt-Shift Videography The *tilt-shift effect* is created by tilting the camera’s lens relative to its image plane which results in a slanted focus plane with a wedge-shaped depth of field that produces the iconic miniature look [Held et al. 2010]. (The purpose of lens *shift* is to correct for perspective distortions like converging parallel lines; however, it does not affect the focus plane or depth of field.) While the lens in most *view cameras* can be tilted and shifted freely thanks to the flexible bellows between lens and film, most lenses in modern cameras are fixed to be parallel to the image sensor, which prevents this effect. There are some special-purpose tilt-shift lenses for modern camera, e.g., from Canon, Nikon or Lensbaby, which can be expensive, but the tilt-shift look is baked into the recorded footage and cannot be modified after capture. We show virtual tilt-shift videography in Figure 5.16 and our video by refocusing with a tilted virtual lens [Merklinger 2010]. This provides ultimate flexibility as the desired look can be modified and tweaked interactively.

Dolly Zoom Depth maps also enable other applications such as limited novel-view synthesis. When combined with the video refocusing presented earlier, this provides the two ingredients required for a dolly zoom (or ‘Hitchcock Zoom’): a camera on a virtual dolly that moves towards or away from the scene, and a carefully controlled virtual camera zoom that keeps an object of interest at constant size (see supplemental video at the project website). Assuming thin-lens optics, this is achieved by varying the focal length f and object-to-lens distance u such that the magnification $M = f/(u - f)$ remains constant for the selected object.

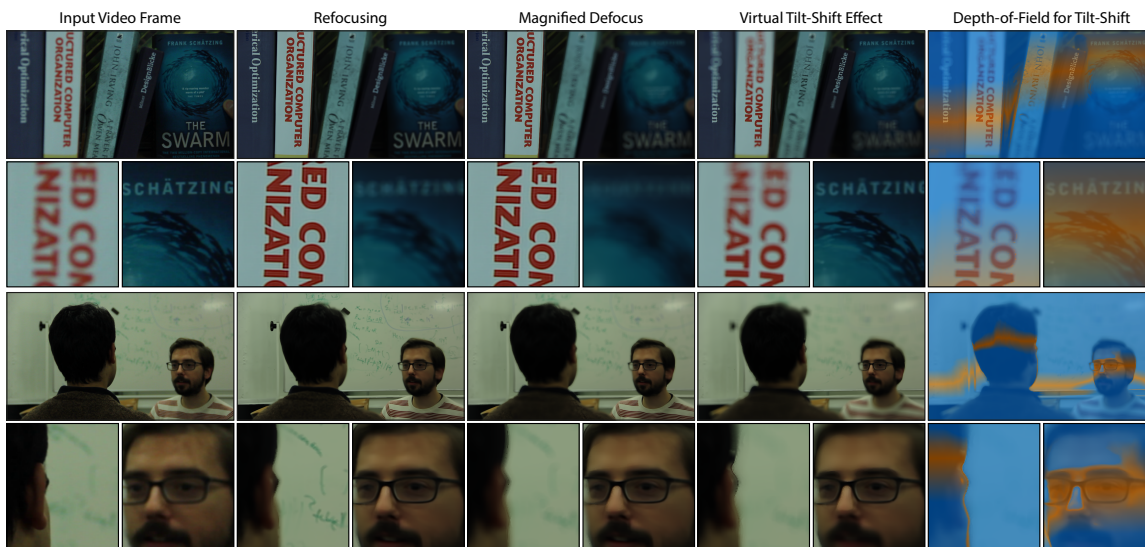


Figure 5.16: Video refocusing results. We first synthetically refocus the input video, then increase the defocus blur by increasing the aperture (smaller f -number), and finally apply a virtual tilt-shift effect, which results in a slanted focus plane. Please see our video for full results.

5.7 Summary

In this chapter, we have presented the first algorithm for space-time coherent depth-from-defocus from video. It enables reconstruction of all-in-focus RGB-D video of dynamic scenes with an unmodified commodity video camera. It takes a different view on RGB-D video computation by turning defocus blur – an effect often regarded as an unwanted artifact – into a valuable signal. From an input video with purposefully provoked defocus blur, e.g., by simply turning the lens, it enables the computation of space-time-coherent depth maps, deblurred all-in-focus video, and the focus distance for each video frame. Our end-to-end video depth-from-defocus method relies on several algorithmic contributions, including an alignment scheme robust to strongly varying focus settings, an image-based method for accurate focus distance estimation, and a space-time coherent depth estimation and deblurring approach. We have extensively evaluated our method and its components, and showed that it enables compelling focus post-processing, such as video refocusing, tilt-shift and dolly-zoom refocusing.

The algorithmic contributions proposed in this chapter greatly advance the-state-of-the-art in dynamic depth-from-defocus and allow focus editing on faces in dynamic scenes. We believe that, when combined with the contributions presented in the previous chapters, it will allow a complete facial reconstruction and editing even including additional lens effects using monocular video clips, e. g., captured on set or on mobile devices, downloaded from YouTube, or from existing videos missing focus data. There still remain some challenges concerning the highly realistic facial manipulation that need to be addressed first to generalize to face images in the wild, that is to say face images which were unseen in the training dataset. In addition, a facial manipulation technology with a high level of photorealism leads to the question how to detect and prevent malicious use of such creative applications. Chapter 6 will give an outlook to future directions in this regard.

Chapter 6

Conclusion

Digital face capture and editing technologies are essential to achieve various visual effects on virtual humans in movies. Thanks to cutting-edge advances in computer graphics and machine learning fields, modern face editing technology has become more sophisticated and photo-realistic. In practice, however, state-of-the-art production systems resort to computationally heavy model-based methods, in-studio controlled setups and skilled artists to obtain high-quality visual effects on digital face models. Therefore, improving face editing pipelines in terms of computational efficiency still remains important. Another limitation that comes with model-based face editing methods is lack of visual realism in modified face images, and thus it often requires tedious manual corrections by skilled artists in post-production. In addition, there has been less awareness about applying lens effects on faces and general scenes to face editing pipelines, which also plays an important role to better perceive visual effects.

The thesis has presented state-of-the-art algorithms to advance face reconstruction and editing pipelines towards the following goals: Real-time inverse face rendering, face editing with a high level of photorealism, and various focus editing effects. For inverse face rendering, the thesis has presented a deep convolutional neural network and a boosting framework that jointly estimates all facial rendering parameters from a single face input in real time. For face editing, the thesis has introduced a novel rendering-to-video translation neural network that enables re-animation of portrait videos with a high level of photorealism. For dynamic lens effects, the thesis has presented a video-based depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video and the focus distance for each frame from a commodity video camera. As a proof of concept, the thesis has demonstrated the proposed methods on various challenging real-world application scenarios such as face reenactment, visual dubbing, interactive face editing, postproduction, video teleconferencing, refocusing and dolly-zoom photography.

In the following, we conclude by shortly reiterating and summarizing the contributions presented throughout the thesis, and also discuss general directions of future works. In addition, we discuss some open challenges and questions about detection and verification of modified face images and videos.

6.1 Summary and Discussion

Chapter 3 presents a real-time inverse face rendering method to recover facial rendering parameters and thus high-fidelity 3D face models directly from single images. Unlike common model-based methods that fit a parametric face model to image or videos through analysis-by-synthesis optimizations, the proposed method exploits a large-scale synthetic face database annotated with rendering parameters, and finds a direct mapping function that describes the complex relationship between facial images and the rendering parameters using a deep convolutional neural network. The main idea behind this method is to update the training images with real-world face images on the fly to close the domain gap and better reflect the real-world distribution. This process is formulated in a statistical boosting framework. The reconstructed 3D face models provide the basis for advanced face editing.

Chapter 4 goes beyond model-based facial animation and presents a novel data-driven approach that enables highly realistic facial re-animation of portrait videos. To the best of our knowledge, this is, in the computer graphics community, the first that successfully modifies portrait videos in all dimensions of full 3D head position, rotation, face expression, eye gaze and eye blinking with a high level of photorealism, and thereby builds important foundations for a wide range of applications such as face reenactment, visual dubbing, interactive face editing, postproduction and video teleconferencing. The core of this approach is a novel generative neural network with a space-time architecture, that transforms synthetic renderings of a parametric face model into photo-realistic portrait videos under full control. Note that the proposed approach automatically synthesizes natural-looking hairs, an upper body and even cast shadow in the background given the target head motion and facial expression.

Finally, Chapter 5 addresses a problem that received less attention but is important for highly realistic visual effects. It introduces a video-based depth-from-defocus algorithm that computes space-time-coherent depth maps, deblurred all-in-focus video and the focus distance for each frame of a regular video camera recorded with smoothly varying focus. The main contribution of this method is hierarchical correspondence field computation in the presence of strong defocus blur. This method allows for various focus editing effects such as refocusing, tilt-shift editing and dolly-zoom photography in postproduction without requiring complex computational cameras.

Discussion Accurate 3D face modeling from single images is a challenging problem and an important step for the remaining processes of photo-realistic face editing pipelines as stated in Chapter 1. The inverse face rendering framework presented in Chapter 3 employs an off-the-shelf face landmark detector in the preprocessing step to achieve high-quality 3D face reconstruction. Strong occlusions and profile views of faces are problematic and hard to reconstruct as even state-of-the-art face landmark trackers often fail in this case.

The thesis aims to achieve highly realistic face editing that goes beyond traditional model-based approaches as stated in Chapter 1. To this end, Chapter 4 proposes a learning-based method that takes synthetic renderings as a conditional input, and translates them into highly realistic images.

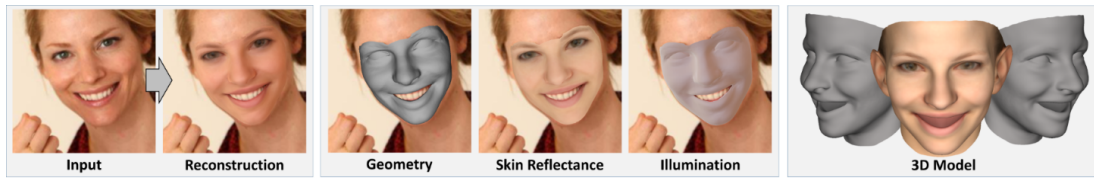


Figure 6.1: The proposed model-based deep convolutional face autoencoder [Tewari et al. 2017] enables unsupervised learning of semantic pose, shape, expression, reflectance and lighting parameters. The trained encoder predicts these parameters from a single monocular image, all at once.

Since the proposed method only constrains the face area in the synthetic renderings, it cannot actively control the motion of an upper body and hairs, or the background. In addition, it is able to produce a medium-resolution output due to the limitations of GPU memory and training time. The remaining challenges such as upper body editing and low-resolution output as well as detection of modified face images are further discussed in Section 6.3.

Focus editing with high optical accuracy is a challenging goal to achieve in postproduction as it requires special camera setups. The advances presented in Chapter 5 enable capturing and editing lens models of commodity video cameras. The proposed method employs a comparably simple focus blur model for computational efficiency. This model, however, degrades near depth boundaries, as it may blur across different depth layers, which can create halos in the estimated depth maps. Section 6.3 provides a possible solution to overcome the limitation of the simple focus model.

6.2 Alternatives

This section provides a brief summary of the two co-authored papers concerning face reconstruction, and also discusses their relevance to the proposed methods in the thesis.

6.2.1 Model-based Face Autoencoder

The deep inverse face rendering network proposed in the thesis is trained by minimizing a facial parameter loss. Alternatively, the parameter space loss can be replaced with an image-based loss. The co-authored work [Tewari et al. 2017] integrates this idea into a model-based deep convolutional autoencoder that reconstructs a 3D human face from a single in-the-wild image as shown in Figure 6.1. To this end, an expert-designed model-based decoder is introduced. The decoder takes as input a learned parameter vector by the preceded encoder, which encodes all face rendering parameters, and reconstructs an output image. In this way, the proposed autoencoder framework leads us to train on unlabeled real-world data at a large scale. The proposed decoder is also differentiable and can be trained end-to-end in an unsupervised manner. Due the loss function which is based on the pixel-wise difference of images, this method could potentially improve accuracy and robustness of the inverse face rendering framework.

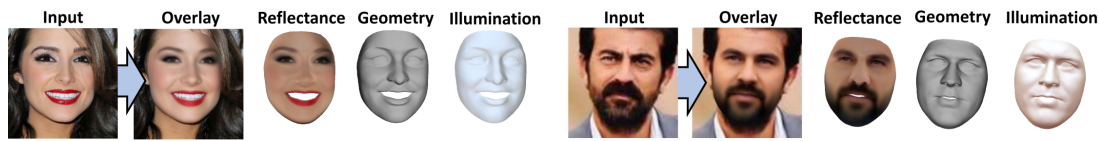


Figure 6.2: The proposed monocular reconstruction approach [Tewari et al. 2018] estimates high-quality facial geometry, skin reflectance including facial hair and incident illumination at over 250 Hz. A trainable multi-level face representation is learned jointly with the feed forward inverse rendering network. End-to-end training is based on a self-supervised loss that requires no dense ground truth.

6.2.2 Multi-level Face Model

The proposed neural rendering method represents highly realistic face models with generative adversarial networks. The co-authored work [Tewari et al. 2018] takes a different approach to model realistic face geometry and appearance as shown in Figure 6.2. Unlike other 3D face modeling approaches which resort to a strong prior in the form of a parametric face model learned from small scale datasets, the proposed method introduces a corrective layer into a model-based deep convolutional autoencoder framework [Tewari et al. 2017] to learn out-of-space geometry and appearance models over a large-scale real-world dataset. This approach is comparable to the state-of-the-art methods in terms of reconstruction quality, however better generalizes to diverse identities and face shapes seen in real-world face images, and even runs at over 250 Hz.

6.3 Future Work and Outlook

This section discusses other remaining aspects and open challenges which are not covered in this thesis in detail. This includes strong occlusions by head poses, upper body editing, low-resolution outputs and simple focus models. In addition, we discuss detection and verification of face images modified with a high level of photorealism.

6.3.1 Challenges

The proposed method for face reconstruction assumes no severe occlusions in face areas to make the initial face tracking step tractable and robust. However, faces in real-world images often exhibit occlusions, for instance due to head poses, hairs, beards, hands and glasses. As a consequence, it could fail accurate face tracking and thus the whole face editing pipeline. This issue could be addressed by further improving face landmark detection algorithms. A recent work by Zhao et al. (2018) shows that the accuracy of occluded face tracking can be improved by an additional neural network trained on the dataset augmented with artificial occlusions. This alternative solution could be applied even to the proposed face editing method that does perform less well on occluded faces.

The proposed neural rendering framework based on conditional generative adversarial networks excludes the synthetic rendering of an upper body from conditioning inputs. Even though it is capable of generating plausible results by the correlation between head poses and an upper body, full control

over the upper body could be achieved by providing the synthetic renderings of the upper body as an additional conditioning input. With the same principle, individual editing of hairs and backgrounds could become feasible.

Similar to most deep learning-based methods, the proposed method for face editing is only able to produce medium-resolution images due to the limitations of GPU memory and training time. A naive approach that changes the size of the convolution kernels to produce high-resolution images leads to blurry outputs or lack fine-scale details. Recently, promising approaches for high-resolution outputs are proposed based on hierarchical neural network models [Wang et al. 2017; Karras et al. 2018]. This could be leveraged to further increase the resolution of the generated output.

The halo artifacts across depth boundaries presented in some of the focus editing applications need to be improved as well. As already discussed by Lee et al. (2010), a possible solution would be to apply a more sophisticated multi-layered defocus blur model [Kraus and Strengert 2007] to the proposed focus editing framework.

6.3.2 Detection and Verification

Besides many creative use cases and its great potential, face editing technology could be misused and applied to modified videos with malicious intent. This raises a question about the authenticity of video contents that people consume every day, especially when there is no proof of origin, and asks for a lot more attention to develop verification systems to help us to spot such modifications.

It is important to note that the understanding of the algorithms and principles behind state-of-the-art video editing tools, as the thesis conducts it, is also the key to develop the detection and verification technologies. The methods for video editing rest on very similar principles to detect video modifications. The face editing approach introduced in Chapter 4 is based on a conditional generative adversarial network that consists of two subnetworks: a generator and a discriminator. These two networks are jointly trained for opposing objectives. The goal of the generator is to produce videos that are indistinguishable from real images. On the other hand, the goal of the discriminator is to spot the synthetically generated video. During training, the aim is to maintain an equilibrium between both networks, i. e., the discriminator should only be able to win in half of the cases, and thus both networks become more sophisticated at their tasks. Note that detection, which can be formulated as a binary classification problem, is in general an easier problem than image generation, which means that it will always be possible to train a highly accurate detector given any specific image forgery approach. Despite the fact that the video modifications become increasingly imperceptible to the human eye, the thesis also conducted several experiments along those lines that show an algorithm can always train very effective discriminators to detect such modifications. An example of such a network that is able to clearly detect such modifications using the principles of the proposed method in Chapter 4 is shown in Figure 6.3.

Many techniques for video authentication already exist, and should be continuously improved alongside new video editing tools. We believe that, in research communities, the development of new editing



Figure 6.3: Detection of modified images: A neural network can reliably detect the images modified by the proposed face editing method. The modified areas, which are detected by the neural network, are highlighted in red.

technologies has to be flanked by detection and verification techniques.

6.4 Closing Remarks

The state-of-the-art approaches used for face reconstruction and editing in movie and game productions are computationally expensive and require tremendous efforts to achieve photorealistic visual effects. To address the limitations, the thesis has made scientific contributions that enable real-time inverse face rendering at high fidelity, highly realistic face editing, and various focus editing effects from a single video input throughout Chapters 3–5.

Despite the advances, there are still many challenges that need to be addressed as discussed in Section 6.3. Among others, detection and verification of digitized face models have recently drawn the attention from the computer graphics and machine learning communities. Interesting works have already been carried out by training on a large-scale corpus of real and modified images using a convolutional neural network [Rössler et al. 2018], or analyzing the frequency of eye blinking using recurrent neural networks [Li et al. 2018]. The contributions of the thesis can be potentially extended in this direction, generating a high-quality training dataset to improve such detection and verification systems. More importantly, a methodical insight should be provided into the development of creative and advanced face editing technologies to root such misuse out.

We hope that the thesis motivates further research on learning-based face reconstruction and editing frameworks, as well as detection and verification of modified face images. We also believe that the thesis will inspire other computer graphics areas to approach open challenges with machine learning-based algorithms [Kim et al. 2019; Liu et al. 2019].

Bibliography

- ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCHE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y. AND ZHENG, X. (2015): TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems., Software available from tensorflow.org [⟨URL: https://www.tensorflow.org/⟩](https://www.tensorflow.org/) 60
- ALDRIAN, O. AND SMITH, W. A. P. (2013): Inverse Rendering of Faces with a 3D Morphable Model. *IEEE TPAMI*, 35 (5), 1080–1093, ISSN 0162–8828 32, 33
- ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, J.-Y., MA, W.-C., WANG, C.-C. AND DEBEVEC, P. (2010): The Digital Emily Project: Achieving a Photorealistic Digital Actor. *IEEE Computer Graphics and Applications*, 30 (4), 20–31, ISSN 0272–1716 16, 36, 56
- AVERBUCH-ELOR, H., COHEN-OR, D., KOPF, J. AND COHEN, M. F. (2017): Bringing Portraits to Life. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 36 (6), 196:1–13 [⟨URL: http://cs.tau.ac.il/~averbuch1/portraitslife/⟩](http://cs.tau.ac.il/~averbuch1/portraitslife/), ISSN 0730–0301 15, 16, 50, 52, 69, 73
- BAE, S. AND DURAND, F. (2007): Defocus Magnification. *Computer Graphics Forum (Eurographics)*, 26 (3), 571–579 [⟨URL: http://people.csail.mit.edu/soonmin/dof/⟩](http://people.csail.mit.edu/soonmin/dof/) 83, 101
- BANDO, Y., CHEN, B.-Y. AND NISHITA, T. (2008): Extracting depth and matte using a color-filtered aperture. *ACM TOG (SIGGRAPH Asia)*, 27 (5), 134:1–9, ISSN 0730–0301 81
- BARNES, C., SHECHTMAN, E., FINKELSTEIN, A. AND GOLDMAN, D. B. (2009): PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM TOG (SIGGRAPH)*, 28 (3), 24 87
- BARRON, J. T., ADAMS, A., SHIH, Y. AND HERNÁNDEZ, C. (2015): Fast Bilateral-Space Stereo for Synthetic Defocus. In *CVPR* 16, 81
- BEELEER, T. AND BRADLEY, D. (2014): Rigid Stabilization of Facial Expressions. *ACM Trans. Graph.* 33 (4), 44:1–44:9 16

- BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W. AND GROSS, M. (2011): High-quality Passive Facial Performance Capture Using Anchor Frames. *ACM Trans. Graph.* 30 (4), 75:1–75:10 16
- BHAGAVATULA, C., ZHU, C., LUU, K. AND SAVVIDES, M. (2017): Faster Than Real-time Facial Alignment: A 3D Spatial Transformer Network Approach in Unconstrained Poses. In *ICCV* 35
- BLANZ, V., SCHERBAUM, K., VETTER, T. AND SEIDEL, H.-P. (2004): Exchanging Faces in Images. *Computer Graphics Forum (Eurographics)*, 23 (3), 669–676, ISSN 1467–8659 16, 52
- BLANZ, V. AND VETTER, T. (1999): A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, ISBN 0–201–48560–5, 187–194 16, 26, 33, 34, 36, 37, 52, 55, 56
- BOOTH, J., ANTONAKOS, E., PLOUMPIS, S., TRIGEORGIS, G., PANAGAKIS, Y. AND ZAFEIRIOU, S. (2017): 3D Face Morphable Models “in-the-wild”. In *CVPR* 33
- BOOTH, J., ROUSSOS, A., PONNIAH, A., DUNAWAY, D. AND ZAFEIRIOU, S. (2018): Large Scale 3D Morphable Models. *IJCV*, 126 (2), 233–254, ISSN 1573–1405 33, 52
- BOUAZIZ, S., WANG, Y. AND PAULY, M. (2013): Online Modeling for Realtime Facial Animation. *ACM Trans. Graph.* 32 (4), 40:1–40:10 27
- BRADSKI, G. AND KAEHLER, A. (2013): *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. 2nd edition., ISBN 1449314651, 9781449314651 24
- BREGLER, C., COVELL, M. AND SLANEY, M. (1997): Video Rewrite: Driving Visual Speech with Audio. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, ISBN 0–89791–896–7, 353–360 52
- BULAT, A. AND TZIMIROPOULOS, G. (2017): How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In *ICCV* [URL: https://www.adrianbulat.com/face-alignment/](https://www.adrianbulat.com/face-alignment/) 35
- BUTLER, D. J., WULFF, J., STANLEY, G. B. AND BLACK, M. J. (2012): A Naturalistic Open Source Movie for Optical Flow Evaluation. In *ECCV* [URL: http://sintel.is.tue.mpg.de/](http://sintel.is.tue.mpg.de/), ISBN 978–3–642–33782–6 97
- CAO, C., BRADLEY, D., ZHOU, K. AND BEELER, T. (2015): Real-time High-fidelity Facial Performance Capture. *ACM ToG*, 34 (4), 46:1–9, ISSN 0730–0301 34, 50, 52
- CAO, C., HOU, Q. AND ZHOU, K. (2014a): Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM Transactions on Graphics (SIGGRAPH)*, 33 (4), 43:1–10, ISSN 0730–0301 50
- CAO, C., WENG, Y., LIN, S. AND ZHOU, K. (2013): 3D Shape Regression for Real-time Facial Animation. *ACM ToG*, 32 (4), 41:1–10, ISSN 0730–0301 34

- CAO, C., WENG, Y., ZHOU, S., TONG, Y. AND ZHOU, K. (2014b): FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE TVCG*, 20 (3), 413–425, ISSN 1077–2626 36, 41, 44, 47, 52, 56
- CAO, C., WU, H., WENG, Y., SHAO, T. AND ZHOU, K. (2016): Real-time Facial Animation with Image-based Dynamic Avatars. *ACM Transactions on Graphics (SIGGRAPH)*, 35 (4), 126:1–12, ISSN 0730–0301 50
- CARREIRA, J., AGRAWAL, P., FRAGKIADAKI, K. AND MALIK, J. (2016): Human Pose Estimation With Iterative Error Feedback. In *CVPR* [URL: https://people.eecs.berkeley.edu/~carreira/ief/](https://people.eecs.berkeley.edu/~carreira/ief/) 34
- CHAKRABARTI, A. AND ZICKLER, T. (2012): Depth and Deblurring from a Spectrally-Varying Depth-of-Field. In *ECCV*, ISBN 978–3–642–33714–7, 648–661 81
- CHANG, Y.-J. AND EZZAT, T. (2005): Transferable Videorealistic Speech Animation. In *Symposium on Computer Animation (SCA)*, 143–151 52
- CHEN, Q. AND KOLTUN, V. (2017): Photographic Image Synthesis with Cascaded Refinement Networks. In *International Conference on Computer Vision (ICCV)* [URL: http://cqf.io/ImageSynthesis/](http://cqf.io/ImageSynthesis/), 1520–1529 53, 59
- CHO, S., WANG, J. AND LEE, S. (2012): Video deblurring for hand-held cameras using patch-based synthesis. *ACM TOG (SIGGRAPH)*, 31 (4), 64:1–9, ISSN 0730–0301 83
- CHRYSOS, G. G., ANTONAKOS, E., SNAPE, P., ASTHANA, A. AND ZAFEIRIOU, S. (2017): A Comprehensive Performance Evaluation of Deformable Face Tracking “In-the-Wild”. *IJCV preprints*, ISSN 1573–1405 35
- COOTES, T. F., EDWARDS, G. J. AND TAYLOR, C. J. (2001): Active appearance models. *IEEE TPAMI*, 23 (6), 681–685, ISSN 0162–8828 33, 34
- CRISPELL, D. AND BAZIK, M. (2017): Pix2face: Direct 3D Face Model Estimation. In *ICCV Workshops* 34
- DALE, K., SUNKAVALLI, K., JOHNSON, M. K., VLASIC, D., MATUSIK, W. AND PFISTER, H. (2011): Video face replacement. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 30 (6), 130:1–10 [URL: http://gvi.seas.harvard.edu/paper/video-face-replacement/](http://gvi.seas.harvard.edu/paper/video-face-replacement/), ISSN 0730–0301 52
- DOU, P., SHAH, S. K. AND KAKADIARIS, I. A. (2017): End-to-end 3D face reconstruction with deep neural networks. In *CVPR* 34
- DUONG, C. N., LUU, K., QUACH, K. G. AND BUI, T. D. (2016): Deep Appearance Models: A Deep Boltzmann Machine Approach for Face Modeling., arXiv:1607.06871 [URL: https://arxiv.org/abs/1607.06871](https://arxiv.org/abs/1607.06871) 34

- EIGEN, D. AND FERGUS, R. (2015): Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV* [〈URL: http://www.cs.nyu.edu/~deigen/dnl/〉](http://www.cs.nyu.edu/~deigen/dnl/) 81
- EZZAT, T., GEIGER, G. AND POGGIO, T. (2002): Trainable Videorealistic Speech Animation. *ACM Transactions on Graphics (SIGGRAPH)*, 21 (3), 388–398, ISSN 0730–0301 53
- FORSYTH, D. A. AND PONCE, J. (2012): *Computer Vision: A Modern Approach*. 2nd edition. 23, 24
- FRIED, O., SHECHTMAN, E., GOLDMAN, D. B. AND FINKELSTEIN, A. (2016): Perspective-aware Manipulation of Portrait Photos. *ACM Transactions on Graphics (SIGGRAPH)*, 35 (4), 128:1–10, ISSN 0730–0301 52
- FYFFE, G., JONES, A., ALEXANDER, O., ICHIKARI, R. AND DEBEVEC, P. (2014): Driving High-Resolution Facial Scans with Video Performance Capture. *ACM Transactions on Graphics*, 34 (1), 8:1–14, ISSN 0730–0301 52
- GANIN, Y., KONONENKO, D., SUNGATULLINA, D. AND LEMPITSKY, V. (2016): DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation. In *European Conference on Computer Vision (ECCV)*, 311–326 53
- GARRIDO, P. (2017): High-quality Face Capture, Animation and Editing from Monocular Video. Ph. D thesis, Saarland University 24, 26
- GARRIDO, P., VALGAERTS, L., REHMSEN, O., THORMAEHLEN, T., PÉREZ, P. AND THEOBALT, C. (2014): Automatic Face Reenactment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN 1063–6919, 4217–4224 52
- GARRIDO, P., VALGAERTS, L., SARMADI, H., STEINER, I., VARANASI, K., PÉREZ, P. AND THEOBALT, C. (2015): VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track. *Computer Graphics Forum* 34 (2) 16, 53, 64, 65, 72, 74
- GARRIDO, P., VALGAERTS, L., WU, C. AND THEOBALT, C. (2013): Reconstructing detailed dynamic face geometry from monocular video. *ACM ToG*, 32 (6), 158:1–10 [〈URL: http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/〉](http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/), ISSN 0730–0301 25, 34, 41, 46
- GARRIDO, P., ZOLLHÖFER, M., CASAS, D., VALGAERTS, L., VARANASI, K., PÉREZ, P. AND THEOBALT, C. (2016): Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM ToG*, 35 (3), 28:1–15 [〈URL: http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig/〉](http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig/) 25, 31, 32, 34, 43, 44, 46, 52, 53, 57
- GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. AND BENGIO, Y. (2014): Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* 27, 53

- GROSSMANN, P. (1987): Depth from focus. *Pattern Recognition Letters*, 5 (1), 63–69, ISSN 0167–8655 82
- GUO, Y., ZHANG, J., CAI, J., JIANG, B. AND ZHENG, J. (2017): 3DFaceNet: Real-time Dense Face Reconstruction via Synthesizing Photo-realistic Face Images., arXiv:1708.00980 (URL: <https://arxiv.org/abs/1708.00980>) 34
- HE, K., ZHANG, X., REN, S. AND SUN, J. (2016): Deep Residual Learning for Image Recognition. In *CVPR* (URL: <http://arxiv.org/abs/1512.03385>) 38, 42, 43
- HELD, R. T., COOPER, E. A., O'BRIEN, J. F. AND BANKS, M. S. (2010): Using blur to affect perceived distance and size. *ACM TOG*, 29 (2), 19:1–16, ISSN 0730–0301 16, 82, 101
- HINTON, G. E. AND SALAKHUTDINOV, R. (2006): Reducing the Dimensionality of Data with Neural Networks. *Science*, 313 (5786), 504–507, ISSN 0036–8075 53
- HOIEM, D., EFROS, A. A. AND HEBERT, M. (2005): Automatic photo pop-up. *ACM TOG (SIGGRAPH)*, 24 (3), 577–584, ISSN 0730–0301 81
- HU, L., SAITO, S., WEI, L., NAGANO, K., SEO, J., FURSUND, J., SADEGHI, I., SUN, C., CHEN, Y.-C. AND LI, H. (2017): Avatar Digitization from a Single Image for Real-time Rendering. *ACM ToG*, 36 (6), 195:1–14, ISSN 0730–0301 50
- HU, Z., XU, L. AND YANG, M.-H. (2014): Joint Depth Estimation and Camera Shake Removal from Single Blurry Image. In *CVPR* 83
- HUANG, G. B., RAMESH, M., BERG, T. AND LEARNED-MILLER, E. (2007): Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst (07-49). – Technical report 41, 47, 48
- HUANG, X., ZHANG, S., WANG, Y., METAXAS, D. N. AND SAMARAS, D. (2004): A Hierarchical Framework For High Resolution Facial Expression Tracking. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPR Workshops '04, 22 16
- ICHIM, A. E., BOUAZIZ, S. AND PAULY, M. (2015): Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM ToG*, 34 (4), 45:1–14, ISSN 0730–0301 34, 50, 52
- IOFFE, S. AND SZEGEDY, C. (2015): Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* 28
- ISAKSEN, A., MCMILLAN, L. AND GORTLER, S. J. (2000): Dynamically Reparameterized Light Fields. In *SIGGRAPH*, ISBN 1–58113–208–5, 297–306 16, 82
- ISOLA, P., ZHU, J.-Y., ZHOU, T. AND EFROS, A. A. (2017): Image-to-Image Translation with Conditional Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (URL: <https://phillipi.github.io/pix2pix/>), ISSN 1063–6919, 5967–5976 51, 53, 59

- JACKSON, A. S., BULAT, A., ARGYRIOU, V. AND TZIMIROPOULOS, G. (2017): Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression. In *ICCV* [⟨URL: http://aaronspence.co.uk/papers/jackson2017recon/⟩](http://aaronspence.co.uk/papers/jackson2017recon/) 34, 44, 45, 46, 47
- JIA, Y., SHELFHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S. AND DARRELL, T. (2014): Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the International Conference on Multimedia* [⟨URL: http://caffe.berkeleyvision.org/⟩](http://caffe.berkeleyvision.org/), ISBN 978-1-4503-3063-3, 675-678 38
- JIANG, L., ZHANG, J., DENG, B., LI, H. AND LIU, L. (2017): 3D Face Reconstruction with Geometry Details from a Single Image., arXiv:1702.05619 [⟨URL: https://arxiv.org/abs/1702.05619⟩](https://arxiv.org/abs/1702.05619) 34
- JIN, X. AND TAN, X. (2017): Face alignment in-the-wild: A Survey. *Computer Vision and Image Understanding*, 162, 1-22, ISSN 1077-3142 35
- KAJIYA, J. T. (1986): The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 143-150 25
- KALANTARI, N. K., SHECHTMAN, E., BARNES, C., DARABI, S., GOLDMAN, D. B. AND SEN, P. (2013): Patch-based High Dynamic Range Video. *ACM TOG (SIGGRAPH Asia)*, 32 (6), 202:1-8 [⟨URL: http://www.ece.ucsb.edu/~psen/PaperPages/HDRVideo/⟩](http://www.ece.ucsb.edu/~psen/PaperPages/HDRVideo/), ISSN 0730-0301 97, 99
- KARRAS, T., AILA, T., LAINE, S. AND LEHTINEN, J. (2018): Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations (ICLR)* 53, 75, 107
- KEMELMACHER-SHLIZERMAN, I. AND SEITZ, S. M. (2011): Face reconstruction in the wild. In *ICCV*, ISSN 1550-5499, 1746-1753 32
- KEMELMACHER-SHLIZERMAN, I. (2013): Internet-Based Morphable Model. In *International Conference on Computer Vision (ICCV)*, ISSN 1550-5499, 3256-3263 52
- KEMELMACHER-SHLIZERMAN, I., SANKAR, A., SHECHTMAN, E. AND SEITZ, S. M. (2010): Being John Malkovich. In *European Conference on Computer Vision (ECCV)*, ISBN 978-3-642-15549-9, 341-353 52
- KEMELMACHER-SHLIZERMAN, I., SHECHTMAN, E., GARG, R. AND SEITZ, S. M. (2011): Exploring photobios. *ACM Transactions on Graphics (SIGGRAPH)*, 30 (4), 61:1-10, ISSN 0730-0301 52
- KIM, H., ELGHARIB, M., ZOLLHÖFER, M., SEIDEL, H.-P., BEELER, T., RICHARDT, C. AND THEOBALT, C. (2019): Neural Style-Preserving Visual Dubbing. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2019)*, 38 (6), 178:1-13 108

- KIM, H., GARRIDO, P., TEWARI, A., XU, W., THIES, J., NIESSNER, M., PÉREZ, P., RICHARDT, C., ZOLLHÖFER, M. AND THEOBALT, C. (2018a): Deep Video Portraits. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2018)*, 37 (4), 163:1–14 [50](#)
- KIM, H., RICHARDT, C. AND THEOBALT, C. (2016): Video Depth-From-Defocus. In *International Conference on 3D Vision*, 370–379 [79](#)
- KIM, H., ZOLLHÖFER, M., TEWARI, A., THIES, J., RICHARDT, C. AND THEOBALT, C. (2018b): InverseFaceNet: Deep Monocular Inverse Face Rendering. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition* [32](#)
- KINGMA, D. P. AND BA, J. (2015): Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)* [60](#)
- KLAUDINY, M., MCDONAGH, S., BRADLEY, D., BEELER, T. AND MITCHELL, K. (2017): Real-Time Multi-View Facial Capture with Synthetic Training. *Computer Graphics Forum (Proceedings of Eurographics)*, 36 (2), 325–336, ISSN 1467–8659 [34](#)
- KLEHM, O., ROUSSELLE, F., PAPAS, M., BRADLEY, D., HERY, C., BICKEL, B., JAROSZ, W. AND BEELER, T. (2015): Recent Advances in Facial Appearance Capture. *Computer Graphics Forum*, 34 (2), 709–733, ISSN 1467–8659 [33](#)
- KRAHENBUHL, P. AND KOLTUN, V. (2011): Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS* [〈URL: http://graphics.stanford.edu/projects/densecrf/〉](http://graphics.stanford.edu/projects/densecrf/) [89](#), [90](#)
- KRAUS, M. AND STRENGERT, M. (2007): Depth-of-Field Rendering by Pyramidal Image Processing. *Computer Graphics Forum (Eurographics)*, 26 (3), 645–654, ISSN 1467–8659 [100](#), [107](#)
- KRIZHEVSKY, A., SUTSKEVER, I. AND HINTON, G. E. (2012): ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS* [27](#), [38](#), [42](#), [43](#)
- KULKARNI, T. D., WHITNEY, W., KOHLI, P. AND TENENBAUM, J. B. (2015): Deep Convolutional Inverse Graphics Network. In *NIPS*, 2539–2547 [33](#)
- LAINE, S., KARRAS, T., AILA, T., HERVA, A., SAITO, S., YU, R., LI, H. AND LEHTINEN, J. (2017): Production-level Facial Performance Capture Using Deep Convolutional Neural Networks. In *Proceedings of the Symposium on Computer Animation (SCA)*, ISBN 978–1–4503–5091–4, 10:1–10 [34](#)
- LANG, M., WANG, O., AYDIN, T. O., SMOLIC, A. AND GROSS, M. (2012): Practical temporal consistency for image-based graphics applications. *ACM TOG (SIGGRAPH)*, 31 (4), 34:1–8, ISSN 0730–0301 [89](#)

- LASSNER, C., PONS-MOLL, G. AND GEHLER, P. V. (2017): A Generative Model of People in Clothing. In *International Conference on Computer Vision (ICCV)* (URL: <http://files.is.tuebingen.mpg.de/classner/gp/>), 853–862 [53](#)
- LECUN, Y., BENGIO, Y. ET AL. (1995): Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361 (10), 1995 [27](#)
- LECUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W. AND JACKEL, L. D. (1989): Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1 (4), 541–551 [29](#)
- LEE, S., EISEMANN, E. AND SEIDEL, H.-P. (2010): Real-time lens blur effects and focus control. *ACM TOG (SIGGRAPH)*, 29 (4), 65:1–7, ISSN 0730–0301 [83](#), [100](#), [107](#)
- LEVIN, A., FERGUS, R., DURAND, F. AND FREEMAN, W. T. (2007): Image and depth from a conventional camera with a coded aperture. *ACM TOG (SIGGRAPH)*, 26 (3), 70, ISSN 0730–0301 [16](#), [80](#), [81](#), [82](#), [90](#)
- LEWIS, J. P., ANJYO, K., RHEE, T., ZHANG, M., PIGHIN, F. AND DENG, Z. (2014): Practice and Theory of Blendshape Facial Models. In LEFEBVRE, S. AND SPAGNUOLO, M., EDITORS: *Eurographics 2014 - State of the Art Reports*, 199–218 [26](#), [27](#)
- LI, C., ZHOU, K. AND LIN, S. (2014a): Intrinsic Face Image Decomposition with Human Face Priors. In FLEET, D., PAJDLA, T., SCHIELE, B. AND TUYTELAARS, T., EDITORS: *ECCV Volume 8693*, ISBN 978–3–319–10601–4, 218–233 [32](#), [33](#)
- LI, C., ZHOU, K. AND LIN, S. (2015a): Simulating Makeup Through Physics-Based Manipulation of Intrinsic Image Layers. In *CVPR*, 4621–4629 [33](#)
- LI, H., TRUTOIU, L., OLSZEWSKI, K., WEI, L., TRUTNA, T., HSIEH, P.-L., NICHOLLS, A. AND MA, C. (2015b): Facial Performance Sensing Head-mounted Display. *ACM Transactions on Graphics (SIGGRAPH)*, 34 (4), 47:1–9, ISSN 0730–0301 [50](#)
- LI, H., WEISE, T. AND PAULY, M. (2010): Example-based Facial Rigging. *ACM Trans. Graph.* 29 (4), 32:1–32:6 [27](#)
- LI, H., YU, J., YE, Y. AND BREGLER, C. (2013a): Realtime Facial Animation with On-the-fly Correctives. *ACM Trans. Graph.* 32 (4), 42:1–42:10 [27](#)
- LI, H., YU, J., YE, Y. AND BREGLER, C. (2013b): Realtime Facial Animation with On-the-fly Correctives. *ACM ToG*, 32 (4), 42:1–10, ISSN 0730–0301 [27](#), [34](#)
- LI, K., DAI, Q., WANG, R., LIU, Y., XU, F. AND WANG, J. (2014b): A Data-Driven Approach for Facial Expression Retargeting in Video. *IEEE Transactions on Multimedia*, 16 (2), 299–310, ISSN 1520–9210 [52](#)

- LI, T., BOLKART, T., BLACK, M. J., LI, H. AND ROMERO, J. (2017a): Learning a model of facial shape and expression from 4D scans. *ACM ToG*, 36 (6), 194:1–17 (URL: <http://flame.is.tue.mpg.de/>), ISSN 0730–0301 33
- LI, X., DONG, Y., PEERS, P. AND TONG, X. (2017b): Modeling Surface Appearance from a Single Photograph Using Self-augmented Convolutional Neural Networks. *ACM ToG*, 36 (4), 45:1–11, ISSN 0730–0301 32
- LI, Y., CHANG, M.-C., FARID, H. AND LYU, S. (2018): In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking. *arXiv preprint arXiv:1806.02877* 108
- LIU, K. AND OSTERMANN, J. (2011): Realistic facial expression synthesis for an image-based talking head. In *International Conference on Multimedia and Expo (ICME)*, ISSN 1945–7871 53
- LIU, L., XU, W., ZOLLHÖFER, M., KIM, H., BERNARD, F., HABERMANN, M., WANG, W. AND THEOBALT, C. (2019): Neural Rendering and Reenactment of Human Actor Videos. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2019)*, 38 (5), 139:1–14 108
- LIU, M.-Y., BREUEL, T. AND KAUTZ, J. (2017): Unsupervised Image-to-Image Translation Networks. In *Advances in Neural Information Processing Systems* 53
- LIU, Z., SHAN, Y. AND ZHANG, Z. (2001): Expressive Expression Mapping with Ratio Images. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, ISBN 1–58113–374–X, 271–276 50, 52
- LIU, Z., LUO, P., WANG, X. AND TANG, X. (2015): Deep Learning Face Attributes in the Wild. In *ICCV*, 3730–3738 41, 42, 43, 44, 47
- LU, J., SUNKAVALLI, K., CARR, N., HADAP, S. AND FORSYTH, D. (2016): A Visual Representation for Editing Face Images., *arXiv:1612.00522* (URL: <https://arxiv.org/abs/1612.00522>) 33
- MA, L., SUN, Q., JIA, X., SCHIELE, B., TUYTELAARS, T. AND VAN GOOL, L. (2017): Pose Guided Person Image Generation. In *Advances in Neural Information Processing Systems* 53
- MARTINELLO, M., WAJS, A., QUAN, S., LEE, H., LIM, C., WOO, T., LEE, W., KIM, S.-S. AND LEE, D. (2015): Dual Aperture Photography: Image and Depth from a Mobile Camera. In *ICCP* 81
- MATHER, G. (1996): Image Blur as a Pictorial Depth Cue. *Phil. Trans. R. Soc. B*, 263 (1367), 169–172 16, 80, 82
- MCGUIRE, M., MATUSIK, W., PFISTER, H., HUGHES, J. F. AND DURAND, F. (2005): Defocus video matting. *ACM TOG (SIGGRAPH)*, 24 (3), 567–576, ISSN 0730–0301 83
- MEDINA, J. ALABORT-I AND ZAFEIRIOU, S. (2017): A Unified Framework for Compositional Fitting of Active Appearance Models. *IJCV*, 121 (1), 26–64, ISSN 1573–1405 34

- MERKLINGER, H. M. (2010): Focusing the View Camera. 1st edition. [⟨URL: http://www.trenholm.org/hmmerk/#FVC⟩](http://www.trenholm.org/hmmerk/#FVC) 101
- MIAU, D., COSSAIRT, O. AND NAYAR, S. K. (2013): Focal sweep videography with deformable optics. In *ICCP* 82
- MIRZA, M. AND OSINDERO, S. (2014): Conditional Generative Adversarial Nets., arXiv:1411.1784 [⟨URL: https://arxiv.org/abs/1411.1784⟩](https://arxiv.org/abs/1411.1784) 53
- MORENO-NOGUER, F., BELHUMEUR, P. N. AND NAYAR, S. K. (2007): Active refocusing of images and videos. *ACM TOG (SIGGRAPH)*, 26 (3), 67, ISSN 0730–0301 16, 81, 82
- MÜLLER, C. (1966): Spherical harmonics., *Lecture Notes in Mathematics* 17 37
- NAIR, V. AND HINTON, G. E. (2010): Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814 29
- NAIR, V., SUSSKIND, J. AND HINTON, G. E. (2008): Analysis-by-Synthesis by Learning to Invert Generative Black Boxes. In KŮRKOVÁ, V., NERUDA, R. AND KOUTNÍK, J., EDITORS: *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 971–981 33, 34
- NAYAR, S. K. AND NAKAGAWA, Y. (1994): Shape from focus. *PAMI*, 16 (8), 824–831, ISSN 0162–8828 82
- NG, R. (2005): Fourier Slice Photography. *ACM TOG (SIGGRAPH)*, 24 (3), 735–744, ISSN 0730–0301 82
- NG, R., LEVOY, M., BRÉDIF, M., DUVAL, G., HOROWITZ, M. AND HANRAHAN, P. (2005): Light Field Photography with a Hand-held Plenoptic Camera. Stanford University (2005-02). – CSTR 16, 80, 82
- OLSZEWSKI, K., LI, Z., YANG, C., ZHOU, Y., YU, R., HUANG, Z., XIANG, S., SAITO, S., KOHLI, P. AND LI, H. (2017): Realistic Dynamic Facial Textures from a Single Image using GANs. In *International Conference on Computer Vision (ICCV)*, 5439–5448 50, 53
- OLSZEWSKI, K., LIM, J. J., SAITO, S. AND LI, H. (2016): High-fidelity Facial and Speech Animation for VR HMDs. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 35 (6), 221:1–14, ISSN 0730–0301 50
- PARIS, S. AND DURAND, F. (2009): A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach. *IJCV*, 81, 24–52 90
- PENG, X., FERIS, R. S., WANG, X. AND METAXAS, D. N. (2016): A Recurrent Encoder-Decoder Network for Sequential Face Alignment. In *ECCV* 35

- PENTLAND, A. P. (1987): A New Sense for Depth of Field. *PAMI*, 9 (4), 523–531, ISSN 0162–8828 82, 88, 89
- PERTUZ, S., PUIG, D., GARCIA, M. A. AND FUSIELLO, A. (2013): Generation of All-in-Focus Images by Noise-Robust Selective Fusion of Limited Depth-of-Field Images. *Transactions on Image Processing*, 22 (3), 1242–1251 (URL: http://www.sayonics.com/research/focus_fusion.html), ISSN 1057–7149 82
- PIOTRASCHKE, M. AND BLANZ, V. (2016): Automated 3D Face Reconstruction from Multiple Images Using Quality Measures. In *CVPR*, 3418–3427 34
- POTMESIL, M. AND CHAKRAVARTY, I. (1982): Synthetic Image Generation with a Lens and Aperture Camera Model. *ACM TOG*, 1 (2), 85–108, ISSN 0730–0301 24, 80
- RADFORD, A., METZ, L. AND CHINTALA, S. (2016): Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)* 53
- RAMAMOORTHY, R. AND HANRAHAN, P. (2001a): An efficient representation for irradiance environment maps. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (URL: <http://www.cs.berkeley.edu/~ravir/papers/envmap/>), ISBN 1–58113–374–X, 497–500 56
- RAMAMOORTHY, R. AND HANRAHAN, P. (2001b): A signal-processing framework for inverse rendering. In *SIGGRAPH*, ISBN 1–58113–374–X, 117–128 32, 33, 37
- RICHARDSON, E., SELA, M. AND KIMMEL, R. (2016): 3D Face Reconstruction by Learning from Synthetic Data. In *3DV*, 460–469 32, 34, 44, 46, 52
- RICHARDSON, E., SELA, M., OR-EL, R. AND KIMMEL, R. (2017): Learning Detailed Face Reconstruction from a Single Image. In *CVPR*, ISSN 1063–6919, 5553–5562 32, 34, 44, 45, 46, 47, 52
- RICHARDT, C., STOLL, C., DODGSON, N. A., SEIDEL, H.-P. AND THEOBALT, C. (2012): Coherent Spatiotemporal Filtering, Upsampling and Rendering of RGBZ Videos. *Computer Graphics Forum (Eurographics)*, 31 (2), 247–256 (URL: <http://www.mpi-inf.mpg.de/resources/rgbz-camera/>) 80, 81
- RIGUER, G., TATARCHUK, N. AND ISIDORO, J. (2003): Real-Time Depth of Field Simulation. In ENGEL, W. F., EDITOR: *Shader²*. – chapter 4 101
- RONNEBERGER, O., FISCHER, P. AND BROX, T. (2015): U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ISBN 978–3–319–24574–4, 234–241 53, 58

- RÖSSLER, A., COZZOLINO, D., VERDOLIVA, L., RIESS, C., THIES, J. AND NIESSNER, M. (2018): FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces. *arXiv preprint arXiv:1803.09179* 108
- ROTH, J., TONG, Y. T. AND LIU, X. (2017): Adaptive 3D Face Reconstruction from Unconstrained Photo Collections. *IEEE TPAMI*, 39 (11), 2127–2141, ISSN 0162–8828 34, 52
- RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., BERG, A. C. AND FEI-FEI, L. (2015): ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115 (3), 211–252 38
- RUSSELL, C., YU, R. AND AGAPITO, L. (2014): Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes. In *ECCV* (URL: <http://www.homepages.ucl.ac.uk/~ucabryu/VideoPopup2.html>), ISBN 978–3–319–10583–3 81
- SARAGIH, J. M., LUCEY, S. AND COHN, J. F. (2011a): Deformable Model Fitting by Regularized Landmark Mean-Shift. *IJCV*, 91 (2), 200–215, ISSN 1573–1405 35, 38
- SARAGIH, J. M., LUCEY, S. AND COHN, J. F. (2011b): Real-time avatar animation from a single image. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 117–124 56
- SAXENA, A., SUN, M. AND NG, A. (2009): Make3D: Learning 3D Scene Structure from a Single Still Image. *PAMI*, 31 (5), 824–840, ISSN 0162–8828 81
- SCHÖNBORN, S., EGGER, B., MOREL-FORSTER, A. AND VETTER, T. (2017): Markov Chain Monte Carlo for Automated Face Image Analysis. *IJCV*, 123 (2), 160–183, ISSN 1573–1405 34, 35
- SELA, M., RICHARDSON, E. AND KIMMEL, R. (2017): Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. In *ICCV*, 1585–1594 32, 34, 46, 47, 52, 53
- SHEN, J., ZAFEIRIOU, S., CHRYSOS, G. G., KOSSAIFI, J., TZIMIROPOULOS, G. AND PANTIC, M. (2015): The First Facial Landmark Tracking in-the-Wild Challenge: Benchmark and Results. In *ICCV Workshops*, 1003–1011 41, 47, 48
- SHI, F., WU, H.-T., TONG, X. AND CHAI, J. (2014): Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM ToG*, 33 (6), 222:1–13, ISSN 0730–0301 34, 52
- SHI, J., TAO, X., XU, L. AND JIA, J. (2015a): Break Ames Room Illusion: Depth from General Single Images. *ACM TOG (SIGGRAPH Asia)*, 34 (6), 225:1–11, ISSN 0730–0301 83
- SHI, J., XU, L. AND JIA, J. (2015b): Just Noticeable Defocus Blur Detection and Estimation. In *CVPR* (URL: <http://shijianping.me/jnb/>) 83
- SHROFF, N., VEERARAGHAVAN, A., TAGUCHI, Y., TUZEL, O., AGRAWAL, A. AND CHELLAPPA, R. (2012): Variable focus video: Reconstructing depth and video for dynamic scenes. In *ICCP* 81, 86, 95, 97

- SHU, Z., YUMER, E., HADAP, S., SUNKAVALLI, K., SHECHTMAN, E. AND SAMARAS, D. (2017): Neural Face Editing with Intrinsic Image Disentangling. In *CVPR* 33
- SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I. AND SALAKHUTDINOV, R. (2014): Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15 (1), 1929–1958 29
- SRIVASTAVA, S., SAXENA, A., THEOBALT, C., THRUN, S. AND NG, A. Y. (2009): i23 – Rapid Interactive 3D Reconstruction from a Single Image. In *Vision, Modeling, and Visualization* 81
- STEGER, A. AND TIMOFTE, R. (2016): Failure Detection for Facial Landmark Detectors. In CHEN, C.-S., LU, J. AND MA, K.-K., EDITORS: *Proceedings of ACCV Workshops*, ISBN 978–3–319–54427–4, 361–376 35
- SUBBARAO, M. AND SURYA, G. (1994): Depth from defocus: A spatial domain approach. *IJCV*, 13 (3), 271–294, ISSN 0920–5691 82
- SUMNER, R. W. AND POPOVIĆ, J. (2004): Deformation Transfer for Triangle Meshes. *ACM Transactions on Graphics (SIGGRAPH)*, 23 (3), 399–405 (URL: <http://people.csail.mit.edu/sumner/research/deftransfer/>), ISSN 0730–0301 56
- SUN, D., ROTH, S. AND BLACK, M. J. (2014): A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them. *IJCV*, 106 (2), 115–137, ISSN 0920–5691 87, 97
- SUWAJANAKORN, S., HERNANDEZ, C. AND SEITZ, S. M. (2015a): Depth From Focus With Your Mobile Phone. In *CVPR* 82, 95, 97, 98
- SUWAJANAKORN, S., KEMELMACHER-SHLIZERMAN, I. AND SEITZ, S. M. (2014): Total Moving Face Reconstruction. In FLEET, D., PAJDLA, T., SCHIELE, B. AND TUYTELAARS, T., EDITORS: *ECCV Volume 8692*, ISBN 978–3–319–10592–5, 796–812 34, 52
- SUWAJANAKORN, S., SEITZ, S. M. AND KEMELMACHER-SHLIZERMAN, I. (2015b): What Makes Tom Hanks Look Like Tom Hanks. In *International Conference on Computer Vision (ICCV)*, 3952–3960 16, 52
- SUWAJANAKORN, S., SEITZ, S. M. AND KEMELMACHER-SHLIZERMAN, I. (2017): Synthesizing Obama: Learning Lip Sync from Audio. *ACM Transactions on Graphics (SIGGRAPH)*, 36 (4), 95:1–13, ISSN 0730–0301 15, 50, 53, 68, 72, 74
- TAIGMAN, Y., POLYAK, A. AND WOLF, L. (2017): Unsupervised Cross-Domain Image Generation. In *International Conference on Learning Representations (ICLR)* 53
- TEWARI, A., ZOLLHÖFER, M., GARRIDO, P., BERNARD, F., KIM, H., PÉREZ, P. AND THEOBALT, C. (2018): Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. 106

- TEWARI, A., ZOLLHÖFER, M., KIM, H., GARRIDO, P., BERNARD, F., PÉREZ, P. AND THEOBALT, C. (2017): MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *ICCV* (URL: http://gvv.mpi-inf.mpg.de/projects/MZ/Papers/arXiv2017_FA/page.html), 3735–3744 34, 35, 42, 44, 46, 47, 52, 105, 106
- THIES, J., ZOLLHÖFER, M., NIESSNER, M., VALGAERTS, L., STAMMINGER, M. AND THEOBALT, C. (2015): Real-time Expression Transfer for Facial Reenactment. *ACM ToG*, 34 (6), 183:1–14, ISSN 0730–0301 25, 27, 34, 50
- THIES, J., ZOLLHÖFER, M., STAMMINGER, M., THEOBALT, C. AND NIESSNER, M. (2016): Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *CVPR* (URL: http://people.mpi-inf.mpg.de/~mzollhoefer/Papers/CVPR2016_FF/page.html), 2387–2395 15, 16, 32, 33, 36, 44, 45, 50, 52, 53, 57, 68, 71
- THIES, J., ZOLLHÖFER, M., STAMMINGER, M., THEOBALT, C. AND NIESSNER, M. (2018): FaceVR: Real-Time Facial Reenactment and Eye Gaze Control in Virtual Reality. *ACM Transactions on Graphics* 52
- THOMAS, D. AND TANIGUCHI, R. I. (2016): Augmented Blendshapes for Real-Time Simultaneous 3D Head Modeling and Facial Motion Capture. In *CVPR*, 3299–3308 34
- TRAN, A. T., HASSNER, T., MASI, I. AND MEDIONI, G. (2017): Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network. In *CVPR* (URL: <https://www.openu.ac.il/home/hassner/projects/CNN3DMM/>), 1493–1502 34, 44, 45, 46, 47, 52
- VALGAERTS, L., WU, C., BRUHN, A., SEIDEL, H.-P. AND THEOBALT, C. (2012): Lightweight binocular facial performance capture under uncontrolled lighting. *ACM ToG*, 31 (6), 187:1–11, ISSN 0730–0301 25, 41, 44, 45
- VEERARAGHAVAN, A., RASKAR, R., AGRAWAL, A., MOHAN, A. AND TUMBLIN, J. (2007): Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM TOG (SIGGRAPH)*, 26 (3), 69, ISSN 0730–0301 82
- VLASIC, D., BRAND, M., PFISTER, H. AND POPOVIĆ, J. (2005): Face Transfer with Multilinear Models. *ACM Transactions on Graphics (SIGGRAPH)*, 24 (3), 426–433, ISSN 0730–0301 50, 52
- WANG, C., ZHENG, H., YU, Z., ZHENG, Z., GU, Z. AND ZHENG, B. (2017): Discriminative Region Proposal Adversarial Networks for High-Quality Image-to-Image Translation., arXiv:1711.09554 (URL: <https://arxiv.org/abs/1711.09554>) 75, 107
- WANG, T.-C., LIU, M.-Y., ZHU, J.-Y., TAO, A., KAUTZ, J. AND CATANZARO, B. (2018): High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Conference on Computer Vision and Pattern Recognition (CVPR)* 53

- WANG, Y., HUANG, X., LEE, C.-S., ZHANG, S., LI, Z., SAMARAS, D., METAXAS, D., ELGAMMAL, A. AND HUANG, P. (2004): High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. *Computer Graphics Forum*, 23 (3), 677–686 16
- WEISE, T., BOUAZIZ, S., LI, H. AND PAULY, M. (2011): Realtime Performance-based Facial Animation. *ACM ToG*, 30 (4), 77:1–10, ISSN 0730–0301 27, 34, 50
- WEISE, T., LI, H., VAN GOOL, L. AND PAULY, M. (2009): Face/Off: Live Facial Puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09*, 7–16 16
- WIKIPEDIA: Convolutional Neural Network. ⟨URL: https://en.wikipedia.org/wiki/Convolutional_neural_network⟩ 28
- WOOD, E., BALTRUŠAITIS, T., MORENCY, L.-P., ROBINSON, P. AND BULLING, A. (2018): GazeDirector: Fully articulated eye gaze redirection in video. *Computer Graphics Forum (Eurographics)* 37 (2), ISSN 1467–8659 52
- WU, C., BRADLEY, D., GROSS, M. AND BEELER, T. (2016): An Anatomically-Constrained Local Deformation Model for Monocular Face Capture. *ACM ToG*, 35 (4), 115:1–12 52
- WU, C., WILBURN, B., MATSUSHITA, Y. AND THEOBALT, C. (2011): High-quality Shape from Multi-view Stereo and Shading Under General Illumination. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11*, 969–976 25
- WULFF, J. AND BLACK, M. J. (2014): Modeling Blurred Video with Layers. In *ECCV* , ISBN 978–3–319–10598–7 83
- YI, Z., ZHANG, H., TAN, P. AND GONG, M. (2017): DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. In *International Conference on Computer Vision (ICCV)*, 2868–2876 53
- YU, F. AND GALLUP, D. (2014): 3D Reconstruction from Accidental Motion. In *CVPR* 16, 81
- YU, Y., DEBEVEC, P., MALIK, J. AND HAWKINS, T. (1999): Inverse global illumination: recovering reflectance models of real scenes from photographs. In *SIGGRAPH* , ISBN 0–201–48560–5, 215–224 33
- ZEILER, M. D. (2012): ADADELTA: An Adaptive Learning Rate Method., arXiv:1212.5701 ⟨URL: <https://arxiv.org/abs/1212.5701>⟩ 38
- ZHANG, Z. (2000): A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (11), 1330–1334 24
- ZHAO, F., FENG, J., ZHAO, J., YANG, W. AND YAN, S. (2018): Robust LSTM-Autoencoders for Face De-Occlusion in the Wild. *IEEE Transactions on Image Processing*, 27 (2), 778–790, ISSN 1057–7149 47, 106

- ZHOU, C., LIN, S. AND NAYAR, S. K. (2011): Coded Aperture Pairs for Depth from Defocus and Defocus Deblurring. *IJCV*, 93 (1), 53–72, ISSN 0920–5691 90, 91
- ZHU, J.-Y., PARK, T., ISOLA, P. AND EFROS, A. A. (2017): Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *International Conference on Computer Vision (ICCV)* (URL: <https://junyanz.github.io/CycleGAN/>), 2242–2251 53
- ZHU, X., LEI, Z., LIU, X., SHI, H. AND LI, S. Z. (2016): Face Alignment Across Large Poses: A 3D Solution. In *CVPR*, 146–155 35, 39
- ZOLLHÖFER, M., THIES, J., GARRIDO, P., BRADLEY, D., BEELER, T., PÉREZ, P., STAMMINGER, M., NIESSNER, M. AND THEOBALT, C. (2018): State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Computer Graphics Forum* 37 (2), ISSN 1467–8659 51