Learning Image-Based VR Facial Animation and Face Reenactment

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

vorgelegt von

Andre Rochow

aus

Frechen, Deutschland

Bonn, April 2025

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

Gutachter/Betreuer:	Prof. Dr. rer. nat. Sven Behnke
Gutachter:	Prof. Dr. rer. nat. Volker Blanz

Tag der Promotion:	24.11.2025
Erscheinungsjahr:	2025

Abstract

The field of facial animation deals with the manipulation of facial representations, such as images or meshes, primary with the objective of generating a natural and consistent animation sequence. This thesis focuses on images and presents methods for Virtual Reality (VR) facial animation and face reenactment. Facial animation in virtual reality environments (VR facial animation) is essential for applications that necessitate clear visibility of the user's face and the ability to convey emotional signals and expressions. The primary challenge is to reconstruct the complete face of an individual utilizing a head-mounted display (HMD). In our case, all information relevant for the animation is obtained from one mouth camera mounted below the HMD and two eye cameras inside the HMD. The principal use case for our methods is to animate the face of an operator who controls our robotic avatar system at the ANA Avatar XPRIZE competition.

For the semifinals, we initially propose a real-time capable pipeline with very fast adaptation for specific operators. The method can be trained on talking-head datasets and generalizes to unseen operators, while requiring only a quick enrollment step, during which two short videos are captured. The first video is a sequence of source images from the operator without the VR headset which contain all the important operator-specific appearance information. During inference, we then use the operator keypoint information extracted from a mouth camera and two eye cameras to estimate the target expression, to which we map the appearance of a source still image. In order to enhance the mouth expression accuracy, we dynamically select an auxiliary expression frame from the captured sequence. This selection is done by learning to transform the current mouth keypoints into the source image space, where the alignment can be determined accurately.

Based on this method, we propose an extension that was used in the ANA Avatar XPRIZE finals. We significantly improve the temporal consistency and animation accuracy. In addition, we are able to represent a much broader range of facial expressions by resolving kepoint ambiguities occurring in our method used in the semifinals. Purely keypoint-driven animation approaches struggle with the complexity of facial movements. We present a hybrid method that uses both keypoints and direct visual guidance from a mouth camera. Instead of using only one source image, multiple source images are selected with the intention to cover different facial expressions. We employ an attention mechanism to determine the importance of each source image. To resolve keypoint ambiguities and animate a broader range of mouth expressions, we propose to inject visual mouth camera information into the latent space. We enable training on large-scale talking-head datasets by simulating the mouth camera input with its perspective differences and facial deformations.

We then approach the task of face reenactment, which involves transferring the head motion and facial expressions from a facial driving video to the appearance of a source image, which may be of a different person (cross-reenactment). Most existing methods are CNN-based and estimate optical flow from the source image to the current driving frame. After deforming the source image into the driving frame, it is inpainted and refined to produce the output animation. We propose a transformer-based encoder for computing a set-latent representation of the source image(s). We then predict the output color of a query pixel using a transformer-based decoder, which is conditioned with keypoints and a facial expression vector extracted from the driving frame. Latent representations of the source person are learned in a self-supervised manner that factorize their appearance, head pose, and facial expressions. Thus, they are perfectly suited for cross-reenactment. In contrast to most related work, our method naturally extends to multiple source images and can thus adapt to person-specific facial dynamics. We also propose data augmentation and regularization schemes that are necessary to prevent overfitting and support generalizability of the learned representations. We evaluate our approach in a randomized user study. The results indicate superior performance compared to previous state-of-the-art methods in terms of motion transfer quality and temporal consistency. Finally, we demonstrate in a separate experiment that the method can be adapted for the VR facial animation task, while simultaneously reducing the preprocessing time significantly in comparison to our previous approaches.

Acknowledgments

I would like to express my gratitude to Prof. Dr. Sven Behnke, who provided me with valuable supervision and feedback throughout the thesis process. I would also like to thank Prof. Dr. Volker Blanz for serving as the second reviewer of my thesis. I want to acknowledge the positive work environment provided by my colleagues. Particular appreciation is attributed to my colleague Max Schwarz for our insightful discussions, his constructive feedback, and his assistance in paper writing. Additionally, I would like to thank Michael Schreiber for his contributions to the construction and maintenance of crucial hardware components. My colleagues Christian Lenz and Michael Schreiber kindly allowed me to use their facial images, which was important to show and compare facial animation results. Finally, I would like to express my deepest gratitude to my family and friends.

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2070 - 390732324, the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

Contents

List of Figures

List of Tables

Acronyms

2D	two-dimensional
3D	three-dimensional
6 DoF	six degrees of freedom
AdaIN	Adaptive Instance Normalization
AI	artificial intelligence
AKD	Average Keypoint Distance
CNN	convolutional neural network
fps	frames per second
GPU	graphics processing unit
HMD	head-mounted display
ID	identity
IR	infrared
LPIPS	Learned Perceptual Image Patch Similarity
MLP	multi-layer perceptron
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
VAE	variational autoencoder
VR	Virtual Reality

1

Introduction

The face is the human body's primary way of communicating emotion and expression when interacting with others. Consequently, it is not surprising that facial animation has become a prominent field within the domain of generative artificial intelligence (AI). In general, facial animation deals with the manipulation of facial representations, such as images or meshes, that correspond to existing or non-existing identities. One particularly publicized application of facial animation techniques is the generation of so-called "deepfake" videos that show generated animations of persons without their consent. In this thesis, we explicitly disassociate ourselves from such misuse of technology and focus on beneficial use cases such as VR avatars.

There are different variants of facial animation, such as face reenactment or talking-head synthesis, which differ mainly in the input and objective. The objective of talking-head synthesis (Ma et al., 2023; Richard et al., 2021b; Wang et al., 2023a;

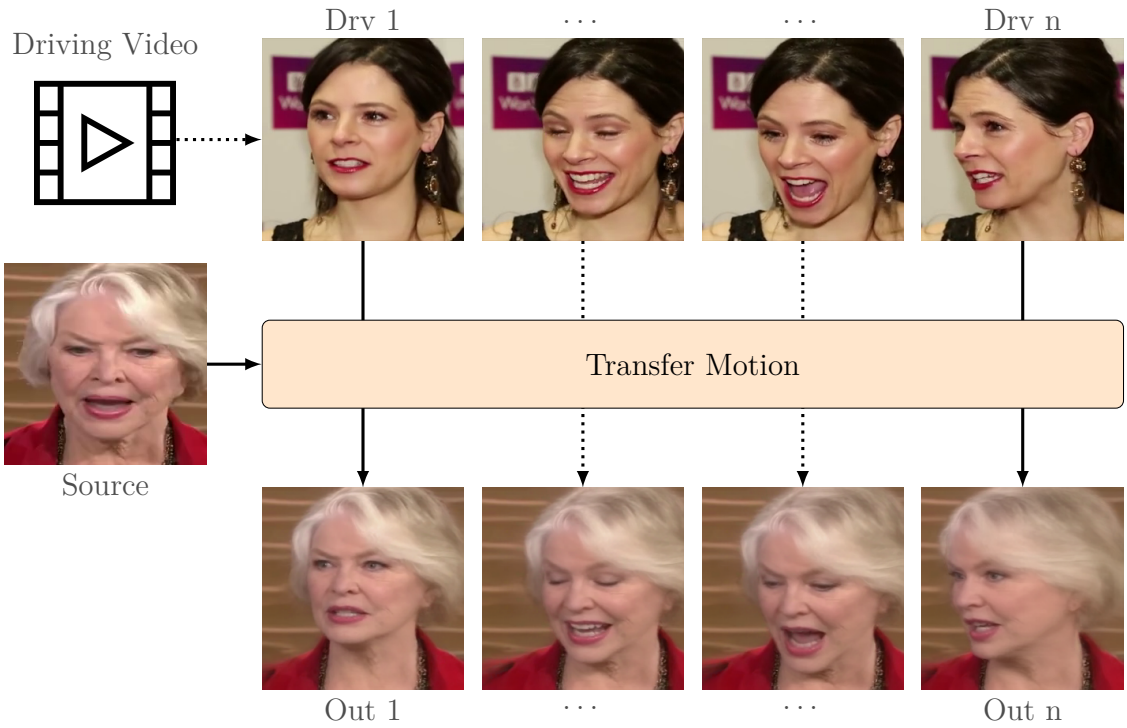

Figure 1.1: Face reenactment. The outputs (bottom) are predicted using the appearance of the source image (left) and the motion (head pose and expression) of the driving frames (top). Outputs are generated by our method from Chapter 4. Images extracted from the VoxCeleb test set (Nagrani et al., 2017).

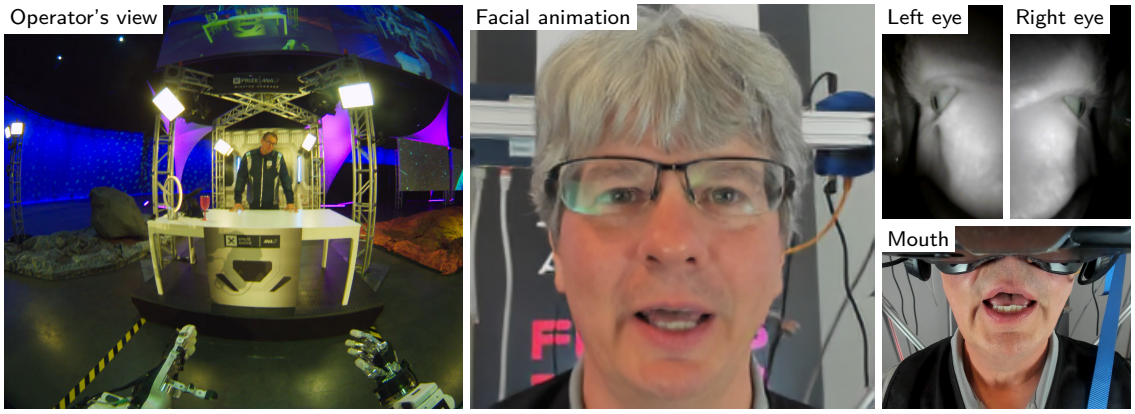

Figure 1.2: VR facial animation. The images show the operator's view displayed on the HMD screen (left) and the facial animation (center) reconstructed from the mouth image (bottom right, cropped for visualization) from a camera below the HMD and the eye camera inputs (top right). Our method from Chapter 3 was used to generate the animation.

Wang et al., 2023b; Wang et al., 2023c; Wang et al., 2021a; Wiles et al., 2018; Xu et al., 2023; Zhou et al., 2019, 2021) is to make speech, emotions, and head pose controllable. Mouth movement is reconstructed from audio or text. Face reenactment (Hong et al., 2022; Pang et al., 2023; Siarohin et al., 2019a,b; Wang et al., 2021b; Zhao and Zhang, 2022, Chapter 4), which is a motion transfer task, involves the transfer of head motion and facial expressions from a driving video to the appearance of a source image (see Fig. 1.1). When the source image and the driving video are of the same person, this is referred to as self-reenactment. As the motion information extracted from the driving video is often significantly smaller in memory than the pixel color information, a particularly popular use case of self-reenactment is low-bandwidth video conferencing (Wang et al., 2021b). Rather than transferring the entire video stream, only the motion information and static source image(s) are transferred.

Another related task is VR facial animation (Lombardi et al., 2018; Richard et al., 2021a; Schwartz et al., 2020; Wei et al., 2019, Chapter 2, Chapter 3), whereby the driving information must be extracted from a person who is wearing an HMD. In such cases, the driving information is often extracted from a mouth camera that is mounted below the HMD and two eye cameras integrated into the HMD (see Fig. 1.2). An innovative use case is three-dimensional (3D) videoconferencing, which enables to communicate with each other in VR. In this context, the animated output should be a 3D representation such as a mesh (Lombardi et al., 2018) to ensure a satisfactory VR experience.

In this thesis, we developed VR facial animation methods for the ANA Avatar XPRIZE competition[1], in which our team NimbRo (at AIS of the University of Bonn), successfully participated. In this competition, a previously unknown operator was required to teleoperate our avatar robot (Lenz et al., 2025; Pätzold et al., 2023; Schwarz et al., 2023, 2021) to complete specific tasks remotely. The avatar system,

1 https://www.xprize.org/prizes/avatar

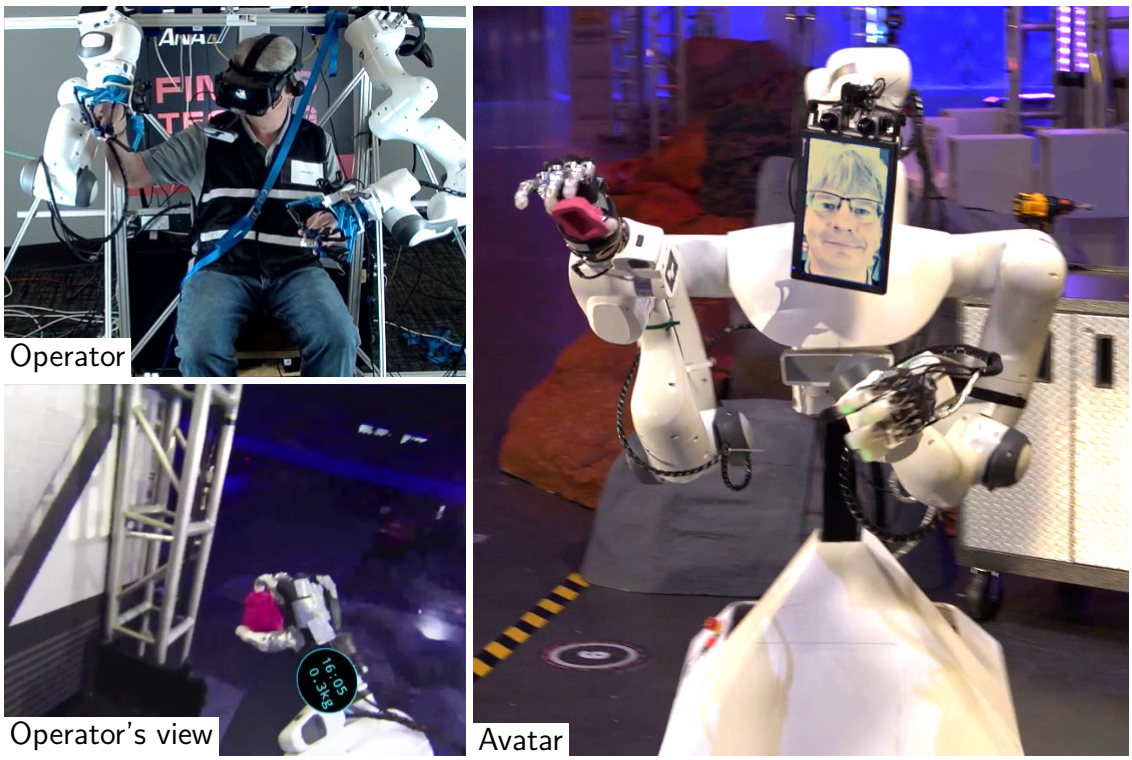

Figure 1.3: Avatar system used at the ANA Avatar XPRIZE competition finals in Long Beach, CA. Modified from Schwarz et al. (2023), ©2023 IEEE, with permissions.

including the operator station, the avatar robot, and the operator's view is displayed in Fig. 1.3. Since we decided to use an HMD for enhanced immersion, it was necessary to animate the face of this operator, because the ability to understand the operator's emotions and gestures, as well as the ability to identify and feel the presence of an operator, was a crucial scoring criterion. The animated face of the operator is displayed on a monitor mounted on the avatar robot (see Fig. 1.3). Consequently, it is sufficient to synthesize a two-dimensional (2D) animation of the operator.

The VR facial animation methods proposed by (Lombardi et al., 2018; Richard et al., 2021a; Schwartz et al., 2020; Wei et al., 2019) require extensive subject-specific data capture (with and without the HMD) and subject-specific training. Given the limited preparation time during the competition when a new operator is assigned, it is necessary to develop an approach that can be utilized with minimal preparation time for unseen operators.

In general, both VR facial animation and face reenactment can be considered motion transfer tasks. Furthermore, VR facial animation can be regarded as a specific variant or subproblem of face reenactment, where the driving information is extracted from different sensors on the HMD, such as cameras capturing the mouth and eyes and tackers to determine the head pose. In contrast, standard face reenactment (Siarohin et al., 2019b) extracts driving information from a single facial driving frame. In our VR facial animation methods, the driving information is derived from the video stream of a mouth camera positioned below the HMD and

two eye cameras (one for each eye) mounted within the HMD (see Fig. 1.2). This poses a significant challenge during training, since it is not possible to capture ground truth data consisting of perfectly corresponding images from the HMD cameras and complete face images without the HMD being worn. We refer to this as the alignment problem.

This thesis presents approaches for VR facial animation that were developed with the objective of enabling the animation of previously unknown operators. The methods were designed with the specific aim of meeting the requirements of the ANA Avatar XPRIZE competition. Additionally, we present a method for face reenactment, where the driving frame can be of the same or a different person, thereby enabling self- and cross-reenactment, respectively.

1.1 Tasks and Key Contribution

This thesis addresses two related tasks: VR facial animation (**T1**) and face reenactment (**T2**). The key contributions of this thesis are:

1. A VR facial animation pipeline that does not necessitate operator-specific training and only requires a quick enrollment step to animate a new operator (addressing task **T1**).

2. A video reconstruction training regime, that allows training on publicly available talking-head datasets, while generalizing to VR facial animation during inference, and thus bypassing the alignment problem (addressing task **T1**).

3. An extension of this approach which processes multiple source images and directly utilizes visual mouth camera information to significantly improve temporal consistency and animation accuracy while representing a broader range of facial expressions (addressing task **T1**).

4. A transformer-based face reenactment method that enables animation from factorized features for appearance, head pose, and facial expression (addressing task **T2**).

The first key contribution (Chapter 2) is a VR facial animation pipeline that enables the utilization for the ANA Avatar XPRIZE competition. Besides eye calibration data, the only operator-specific data required are two short videos of the operator reading the same sentence with and without the HMD being worn. Subsequently, one fixed source image is selected from the source video without the HMD. The driving information is then processed through a keypoint bottleneck. For inference, the two mentioned videos of the operator are employed to facilitate the learning of a keypoint mapping that projects lower-face keypoints, that are visible in the HMD mouth camera, to the source image space. Similarly, we transform eye tracking information (extracted from images of eye cameras inside the HMD) into a predefined eye coordinate system in the source image. This enables the construction

of driving keypoints, based on the source image keypoints, mapped mouth camera keypoints, and eye tracking results. Our motion transfer module follows a flow-based deformation and refinement strategy (Siarohin et al., 2019a,b). We first deform the source image into the driving keypoints using the optical flow predicted by a convolutional motion network, and then refine it to produce the output animation. To improve the animation accuracy, we guide the animation of the mouth region using a retrievable expression frame that has keypoints comparable to those in the mouth camera image. During inference, the expression frame is retrieved from the previously captured source video (i.e., the video of the operator without the HMD).

The second key contribution (Chapter 2) is a training regime for VR facial animation using video reconstruction on publicly available talking-head datasets, such as the VoxCeleb dataset (Nagrani et al., 2017), which contain a large variety of individuals in many different videos. This enables our method to generalize to unseen operators during inference, unlike previous VR facial animation approaches proposed by Lombardi et al. (2018), Richard et al. (2021a), Schwartz et al. (2020), and Wei et al. (2019) which require extensive subject-specific training. In order to leverage a talking-head dataset, we need to train self-reenactment with a source image and driving frame extracted from the same video. This is possible since our inference pipeline is designed to encode driving information solely through keypoints. Moreover, during training the expression frame is mimicked by randomly selecting a frame close to the driving frame in the video sequence.

The third key contribution (Chapter 3) addresses limitations of the first method. While the keypoint bottleneck allows for training on talking-head datasets, it also results in keypoint ambiguities. This occurs when different facial expressions are represented by the same keypoints, which significantly constrains the range of facial expressions that can be animated. We resolve keypoint ambiguities by directly utilizing visual features of the mouth camera image during inference. The mouth camera image is warped into the lower-face keypoints of the source image using the barycentric coordinates of the Delaunay triangulation. During training on talking-head datasets we mimic the presence of a mouth camera image, utilizing the lower facial region of the driving frame itself, augmented with various types of noise to prevent overfitting. Moreover, the visual mouth camera guidance predicts a mask, which gates the deformed source image features, thereby introducing another information bottleneck. Training is mainly performed on the VoxCeleb dataset (Nagrani et al., 2017). However, during finetuning, we also train the network with samples from a small dataset of manually annotated complete face and HMD mouth camera image pairs with a roughly matching lower-face expression. The visual mouth camera guidance reduces temporal inconsistencies that are generated by the abrupt change of the retrieved expression frame. To further improve temporal consistency and animation accuracy, the number of source images is increased to five, and an attention mechanism is employed to weight and aggregate the deformed source image features. Here, the weighting of the attention mechanism is smoothly driven by the keypoints detected in the mouth camera stream. Our method can be used either in a mode

with five fixed source images (maximizing temporal consistency) or with the last source image treated as a retrievable dynamic image. In the latter case, the use of multiple source images effectively mitigates the negative impact of the dynamic image on temporal consistency. Another benefit of utilizing multiple source images is that it enables the network to adapt to specific facial dynamics exhibited by the source person.

The fourth key contribution (Chapter 4) addresses the face reenactment task, where all driving information is encoded in a single driving frame displaying an arbitrary person's head. A multitude of related methods, including our proposed VR facial animation pipelines (Chapters 2 and 3), are based on convolutional neural networks (CNNs), and estimate optical flow to deform source images or features into the target head pose and expression (Hong et al., 2022; Siarohin et al., 2019a,b; Wang et al., 2021b; Zhao and Zhang, 2022). This is then refined to produce the output animation. Motivated by Scene Representation Transformers (SRT) (Sajjadi et al., 2022b), where a set-latent scene representation is learned for performing geometry-free novel view synthesis, we present an implicit transformer-based method, namely FSRT, that encodes the appearance of a person, as given by a single or multiple source images, into a set of latent vectors. Given a driving frame we extract facial keypoints, and a latent expression vector. While keypoints encode all pose information, the expression vector encodes facial expressions of the target animation. Using the source image appearance encoded in the set-latent representation, each output pixel can be sampled by attending to the set-latents, while conditioning on the extracted keypoints, the latent expression vector, and the desired pixel location. In contrast to SRT our representation is not just encoding a static scene, but a very high dimensional scene including the source person appearance, facial dynamics and head motion. Our approach enables the factorization into three components: appearance features, head-pose features, and facial expression features. This allows for the manipulation of each component separately, as well as performing cross-reenactment between individuals. Our method outperforms previous state-of-the-art techniques in terms of cross-reenactment, as shown by the results of a user study. To our knowledge, we are the first to address the face reenactment problem with a transformer-based architecture. Since the expression vectors are decoupled from head pose and appearance information, they can be extracted from a different person (cross-reenactment) even with a different head pose.

We then demonstrate, through a separate experiment, that a modified FSRT model, designated as VR-FSRT, can be effectively employed for VR facial animation, while solely being trained on talking-head datasets. Instead of optimizing a keypoint mapping, utilized to estimate the target lower-face keypoints to which we warp visual mouth camera information (as in our VR facial animation method from Chapter 3), VR-FSRT directly extracts a latent expression vector from the mouth camera image. This vector is then used to condition the decoder when inferring the output image. Compared to the time-consuming process of capturing two videos, a single source image is sufficient, significanty reducing the operator-specific preprocessing time from

15 minutes to approximately 3 minutes compared to the other VR facial animation methods we proposed.

1.2 Publications

The main part of this thesis has already been published in the following peer-reviewed conference papers:

Andre Rochow, Max Schwarz, Michael Schreiber, and Sven Behnke (2022). "VR Facial Animation for Immersive Telepresence Avatars." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2167–2174. DOI: 10.1109/IROS47612.2022.9981892

Andre Rochow, Max Schwarz, and Sven Behnke (2023). "Attention-based VR facial animation with visual mouth camera guidance for immersive telepresence avatars." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1276–1283. DOI: 10.1109/IROS55552.2023.10342522

Andre Rochow, Max Schwarz, and Sven Behnke (2024). "FSRT: Facial Scene Representation Transformer for Face Reenactment from Factorized Appearance, Head-pose, and Facial Expression Features." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7716–7726. DOI: 10.1109/CVPR52733.2024.00737

The following peer-reviewed publications describe important components of our robotic avatar system, and our performance at the ANA Avatar XPRIZE competition. They are related to this thesis as the main motivation for the development of VR facial animation methods was the visualization of the operator's face on the display mounted to the avatar robot. Note that they are cited as external references:

Max Schwarz, Christian Lenz, Raphael Memmesheimer, Bastian Pätzold, Andre Rochow, Michael Schreiber, and Sven Behnke (2023). "Robust immersive telepresence and mobile telemanipulation: NimbRo Avatar wins ANA Avatar XPRIZE Finals." In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. DOI: 10.1109/Humanoids57100.2023.10375179

Bastian Pätzold, Andre Rochow, Michael Schreiber, Raphael Memmesheimer, Christian Lenz, Max Schwarz, and Sven Behnke (2023). "Audio-based roughness sensing and tactile feedback for haptic perception in telepresence." In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1387–1392. DOI: 10.1109/SMC53992.2023.10394062

Christian Lenz, Max Schwarz, Andre Rochow, Bastian Pätzold, Raphael Memmesheimer, Michael Schreiber, and Sven Behnke (2025). "NimbRo wins ANA Avatar XPRIZE immersive telepresence competition: human-centric evaluation and lessons learned." In: *International Journal of Social Robotics* 17.3, pp. 337–361. DOI: 10.1007/s12369-023-01050-9

1.3 Outline

Scientific contributions are presented in Chapters 2 to 4.

In Chapter 2 we present our first VR facial animation method that we used for the ANA Avatar XPRIZE semifinals in Miami, FL.

In Chapter 3 the VR facial animation method is extended. We enhance the range of facial expressions that can be displayed and significantly improve temporal consistency and the motion transfer accuracy. The method was utilized at the ANA Avatar XPRIZE competition finals in Long Beach, CA, which our team won.

Finally, in Chapter 4 we delve into the face reenactment task. We propose an robust method that outperforms previous state-of-the-art methods in cross-reenactment (i.e., when the source image and the driving frame are of different persons).

Each chapter of this thesis is written to be mostly self-contained, so that they can be read separately. However, some chapters reference content and methodology of previous chapters. Especially, the method presented in Chapter 3 builds on the architecture and pipeline of the method presented in Chapter 2.

2

VR Facial Animation for Immersive Telepresence Avatars

Preface

This chapter is adapted from the following publication:
Andre Rochow, Max Schwarz, Michael Schreiber, and Sven Behnke (2022). "VR Facial Animation for Immersive Telepresence Avatars." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2167–2174. DOI: 10.1109/IROS47612.2022.9981892, ©2022 IEEE, presented in Kyoto, Japan.

Statement of Personal Contribution

The author of this thesis is the main author of the publication Rochow et al. (2022) and significantly contributed to all aspects of the publication and this chapter. These contributions particularly include proposing, conceptualizing, designing, and implementing the method, conducting a comprehensive literature review, evaluating the method, analyzing and interpreting the results, and active involvement in the ANA Avatar XPRIZE competition. Additionally, the author of this thesis was responsible for training the model and was the primary contributor to writing the paper and creating the visualizations.

Facial animation is an important task in visual computing. A popular setting is face reenactment, where a facial source image and a facial driving frame, which may be of different persons, are provided. The resulting image should have the appearance of the source image person, but the head pose and facial expression of the driving frame person. Generally, face reenactment methods are trained on talking-head datasets. At inference time, the objective is to utilize arbitrary driving videos to animate the source image person.

A special case in virtual reality is VR facial animation, where the user wears an HMD and is thus not fully visible. Driving information has to be captured by sensors mounted on the headset and is typically incomplete. Limited and occluded information together with large perspective offsets makes VR facial animation exceptionally challenging. Furthermore, many HMDs cause deformations in even the visible areas which particularly limits mouth movements. The alignment problem between mouth camera images and images without the presence of an HMD is one of the biggest challenges for generating training samples. VR facial animation has applications in computer games, immersive telepresence, or 3D videoconferencing in VR. VR allows virtual immersion in another world. When interacting virtually with a VR user, it is often desirable to perceive all their characteristics and facial expressions.

Especially in recent years, deep learning techniques have made great progress in generative modeling (Goodfellow et al., 2014) and rendering (Mildenhall et al., 2020). In this chapter, we demonstrate a deep learning-based approach to the VR facial animation task.

The immediate motivation for this chapter were the semifinals of the ANA Avatar XPRIZE competition[1], where judges interact through avatar systems developed by the participant teams. Here, a specific imposition for a facial animation system is that adaption to the operator has to be finished in one hour. This time has to be shared with the operator training time, so that available time for adaptations is even shorter. Our robotic avatar system allows an operator to directly perceive the world through the eyes of the avatar. We animate the operator on a display which is attached via a six degrees of freedom (6 DoF) arm to the avatar. For more information about our avatar system we refer to Schwarz et al. (2021). Inspired by the method of Siarohin et al. (2019b), our pipeline is trained with the VoxCeleb dataset (Nagrani et al., 2017) to animate a face from a source image, driven by keypoints from a target (driving) frame. We demonstrate how the pipeline can be adapted to generalize to the VR facial animation sub-problem. To enhance the modeling capabilities, we select a third frame, i.e. the expression frame. Using this frame, we directly embed operator-specific expression information into the animation process. We propose an inference pipeline that, given the mouth image captured by a camera below the HMD (mouth camera), dynamically and automatically chooses the optimal expression frame via keypoint-driven image retrieval. Our pipeline runs in real time and can be adapted to an unknown operator with only 15 min preprocessing.

1 https://www.xprize.org/prizes/avatar

Figure 2.1: Operator interacting through the NimbRo Avatar system with a human recipient at the ANA Avatar XPRIZE semifinals. Top: Operator at remote site. Bottom: Two different facial expressions of the operator animated with our method. See also the supplementary video[2] for an animated version.

In addition to the full real-time VR facial animation pipeline, our contributions include:

1. A method allowing fast adaptation to new operators,
2. a training regime that allows offline training on large-scale talking-head datasets,
3. a fast approximate approach to solve the alignment problem between facial image sequences captured with and without an HMD, allowing retrieval of matching frames,
4. an efficient eye tracking method for challenging camera perspectives, trainable in under a minute, and
5. methods improving temporal consistency at inference.

2.1 Related Work

Facial Image Animation. Facial Image Animation is a long-standing problem in computer graphics and aims to generate facial images with controllable expressions. Wiles et al. (2018) propose a method that learns the sampling coordinates from a source image to an embedded image and then from the embedded image to a driving frame using a separate network. Whereas other warping-based methods exist (Siarohin et al., 2019a,b; Wang et al., 2019; Zhao et al., 2021a), there are also indirect approaches (Choi et al., 2018; Pumarola et al., 2018; Zakharov et al., 2019) that rely on generative modeling to perform facial expression synthesis. Zakharov et al. (2019) learn an embedding vector from few source images which then modulate a keypoint-driven generator network via Adaptive Instance Normalization (AdaIN) (Huang and Belongie, 2017), which has demonstrated to be especially well-suited to perform style transfer. For VR facial animation, those approaches, however, must be adapted to work with incomplete driving frames, e.g. captured from mouth and eye cameras.

Keypoint-Driven Face Reenactment. Siarohin et al. (2019a) use deep neural networks to decouple appearance and motion information. They combine the appearance extracted from a source image and the motion derived from the driving video. Their pipeline is separated into a keypoint detector, a dense motion network, and a generator network. Based on this architecture, Siarohin et al. (2019b) encode motion based on keypoint displacements combined with local affine transformations that allow modeling more complex motions compared to using keypoint displacements alone. More recently, Zhao et al. (2021a) proposed an encoder/decoder dense motion network that employs AdaIN (Huang and Belongie, 2017) to transfer source face keypoint geometry to the encoder and driving face keypoint geometry to the decoder. They separate the dense motion network into a global branch and multiple local branches that have a limited visibility—to focus on generating a more accurate motion for the eyes and the mouth area. Furthermore, they investigate how to improve the temporal alignment of the keypoint detector as proposed by Bulat and Tzimiropoulos (2017), which is also the default choice in our approach. Our method is based on Siarohin et al. (2019b), but faces a much more difficult problem, where the driving information comes from multiple input images (eye and mouth cameras) captured from perspectives different from that of the source image camera. To address these issues, we forgo using local affine transformations (Siarohin et al., 2019b) and rely on a keypoint-based image retrieval to simulate the lower face region more precisely.

VR Facial Animation. VR facial animation can be regarded as a special case of face reenactment, where large parts of the driving face are occluded by an HMD. Typically, driving information from multiple sensors is processed and combined. Thies et al. (2018) use image retrieval to obtain the most similar source views.

2 https://www.ais.uni-bonn.de/videos/IROS_2022_Rochow

They use blending to combine the retrieved images to a photo-realistic output. Several methods (Lombardi et al., 2018; Richard et al., 2021a; Wei et al., 2019) render a virtual avatar based on operator-specific geometry. Lombardi et al. (2018) propose to learn a variational autoencoder (VAE) with an encoder that predicts a viewpoint-independent latent variable and a decoder that can be conditioned with extrinsic and intrinsic variables controlling the camera pose, speech, identity, and gaze. Wei et al. (2019) extend this idea and generate ground truth data with expression-preserving style transfer networks, which map between the HMD camera domain and the avatar domain. More recently, Richard et al. (2021a) predict facial coefficients that parameterize an avatar face model with audio and gaze information only. This is especially useful if the lower face region is occluded as well, e.g. with a medical mask. In contrast to these methods, our pipeline does not assume pretrained personalized parametric face models and can therefore be adapted to a specific operator with much less preparation time.

Note that additional related work, which was published after our VR facial animation methods from this chapter and Chapter 3, is described in Sec. 4.5.1.

Neural Rendering. Gafni et al. (2021) propose to learn facial animation using dynamic neural radiance fields (Mildenhall et al., 2020). Whereas they generate impressive results, their pipeline must be trained for a specific person, which is impracticable in our application due to the long training time.

2.2 Method

2.2.1 *Avatar System and Modified VR Headset*

Our robotic avatar system is described in (Schwarz et al., 2021). Briefly put, it allows a human operator to immersive themselves into a remote robot and to interact and cooperate with humans at the remote site.

For visualization on the operator side, we use the *Valve Index* VR headset. In order to capture the mouth expression, we attach a *Logitech Brio* webcam below the HMD (see Fig. 2.2). Furthermore, we allow for eye tracking by mounting two additional cameras and infrared (IR) LEDs inside the HMD.

2.2.2 *Basic Expression Mapping Pipeline*

We propose a pipeline that allows to map the appearance from a source image I_S to the expression and the head pose (motion) present in the driving frame, which may be present as keypoints only (see Figs. 2.3 and 2.4).

Similar to Siarohin et al. (2019b), we separate our pipeline into a keypoint detector $\mathcal{K}$, a motion network $\mathcal{M}$, and an image generator $\mathcal{G}$. The motion network $\mathcal{M}$ uses the keypoints extracted by $\mathcal{K}$ to generate a deformation grid $\mathcal{M}_{S \leftarrow D}$ which warps the source image I_S to the head pose and expression of the driving frame I_D.

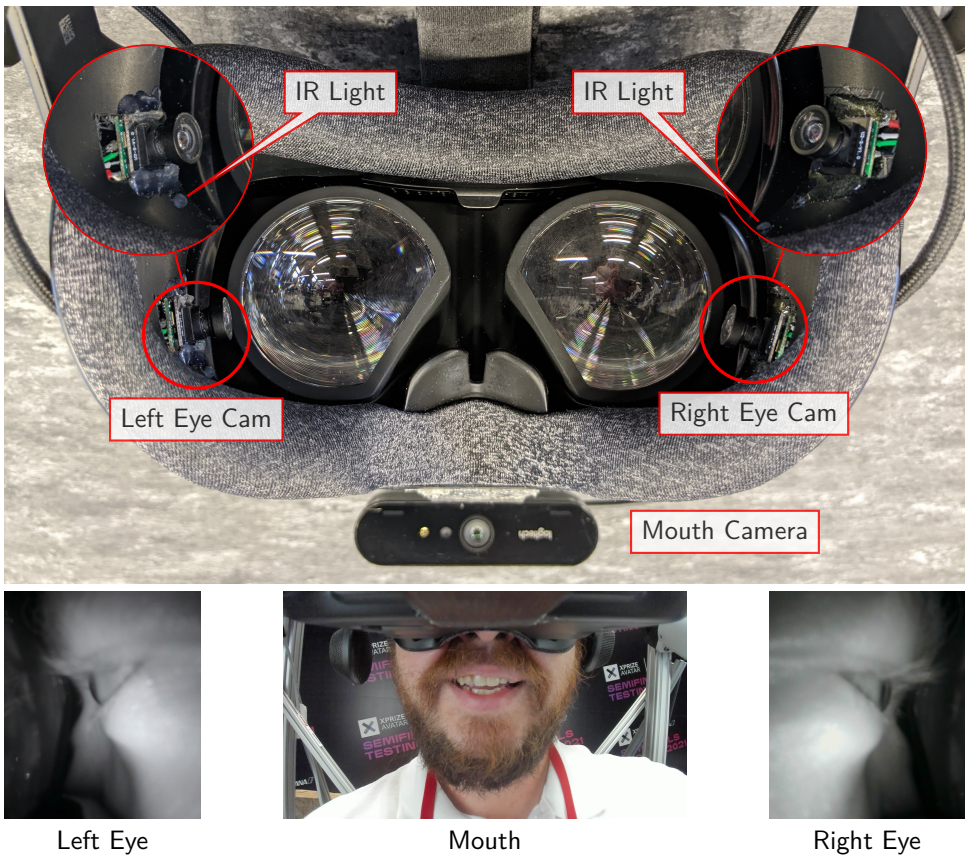

Figure 2.2: The modified Valve Index VR headset. We attached three additional cameras to capture the eyes and the mouth expression of the operator. The inside of the VR headset is lit using IR LEDs. We show the corresponding camera views at the bottom.

Solely warping is insufficient to generate a realistic output, though. To address this issue, we add a generator network $\mathcal{G}$ which then creates the final output image I_O, given the initial motion estimate ${}^{D}I_S$. For precise architectural information, we refer to Siarohin et al. (2019b).

Note that there are approaches that forego using a source image by embedding appearance features directly in the network weights (Gafni et al., 2021; Wei et al., 2019), however, making only implicit use of the appearance as encoded in a given source image allows us to generalize to unseen operators.

Keypoint Detector Network. Our keypoint detector $\mathcal{K}$ extracts keypoints from the source image I_S and from the driving frame I_D. We obtain two sequences of l keypoints, respectively:

$$\mathcal{K}(I_S) = [\, k_S^{(1)}, k_S^{(2)}, \ldots, k_S^{(l)} \,] \text{ and} \tag{2.1}$$

$$\mathcal{K}(I_D) = [\, k_D^{(1)}, k_D^{(2)}, \ldots, k_D^{(l)} \,]. \tag{2.2}$$

Unlike Siarohin et al. (2019b), we separate our keypoint detector into two models:

(i) One model $\mathcal{K}_{\mathcal{VR}}$ for the lower facial expression (detecting the mouth keypoints and one chin keypoint), and

(ii) a global keypoint detector $\mathcal{K}_{\mathcal{F}}$ that has access to images of complete faces, which detects all other keypoints (eye, head pose, etc.).

$\mathcal{K}_{\mathcal{VR}}$ and $\mathcal{K}_{\mathcal{F}}$ are both based on an Hourglass network (Newell et al., 2016).

The global keypoint detector $\mathcal{K}_{\mathcal{F}}$ is trained to detect keypoints, which are primarily encoding the head pose and information about the operator's eyes. We obtain these keypoints using the keypoint detector extracted from a pretrained First Order Motion Model (Siarohin et al., 2019b). This keypoint detector was trained in a self-supervised manner. Additionally, we remove one keypoint from the output of $\mathcal{K}_{\mathcal{F}}$, which captures the position of the lower lip. This keypoint is not required because several mouth keypoints are already extracted by $\mathcal{K}_{\mathcal{VR}}$.

In contrast, $\mathcal{K}_{\mathcal{VR}}$ is trained in a supervised manner using annotated images from the VoxCeleb dataset (Nagrani et al., 2017). We annotate VoxCeleb images by cropping the face and extracting keypoints using the method proposed by Bulat and Tzimiropoulos (2017). However, we are only interested in the keypoints of the lower facial region which are visible in our mouth camera (see Fig. 2.2). These lower-face keypoints extracted by $\mathcal{K}_{\mathcal{VR}}$ are referred to as the VR keypoints. In order to simulate the lower-face image I_M during training, we crop a random quadratic region with the only constraint that all lower-face keypoints must fit into this region. The cropped region is then resized to 128×128 pixels. We therefore implicitly train $\mathcal{K}_{\mathcal{VR}}$ to extract keypoints in partially visible faces—as captured at inference time by the mouth camera.

For simplification, we define

$$\mathcal{K}(I) := \mathcal{K}_{\mathcal{VR}}(I) \oplus \mathcal{K}_{\mathcal{F}}(I) \tag{2.3}$$

to be the concatenation $\oplus$ of both keypoint sequences.

Motion Network. The motion network $\mathcal{M}$ is also based on an Hourglass architecture and produces a deformation of the source frame appearance, represented with the driving frame's facial expression and head pose. Similar to Siarohin et al. (2019b), we first create for each keypoint $k_S^{(j)} \in \mathcal{K}(I_S)$ a shifted source image that aligns $k_S^{(j)}$ with the corresponding driving keypoint $k_D^{(j)} \in \mathcal{K}(I_D)$. These l shifted versions are then fed into $\mathcal{M}$ together with the heatmap representation of the driving keypoints. The motion network then predicts a deformation grid $\mathcal{M}_{S \leftarrow D}$ which can be used to sample the source image deformed to the driving keypoints ${}^{D}I_S$. For a broader explanation we refer to Siarohin et al. (2019b).

To enhance the modeling capabilities, Siarohin et al. (2019b) propose to use local affine transformations for each keypoint instead of just shifting. However, this assumes the existence of a complete driving frame. In contrast, our driving frames

are separated into a mouth camera image and two eye camera images. Particularly, these images are captured from views that are significantly different from the target view. Thus, we rely on a motion network which only processes shifted source images and keypoint heatmaps. This allows us to create imaginary driving frame keypoints with perspective corrections and arbitrary head poses (see Sec. 2.2.3).

Generator Network. The basic generator network has an encoder-decoder architecture and predicts the output image I_O, given the deformed source image ${}^{D}I_S$.

2.2.3 *Inference with VR Headset*

In order to animate an operator controlling our avatar, we first need to capture a source image of them without any occlusions. The major challenge during inference is that we do not have access to a driving frame corresponding to a complete face image. Instead, we have to work with a mouth camera and two cameras capturing the eyes. The basic inference pipeline is illustrated in the non-blue area of Figure 2.3.

To obtain valid driving keypoints to which we can map the source image's appearance, we construct the keypoints $\hat{k}_D^{(j)} \in \mathcal{K}(\hat{I}_D)$ of an imaginary driving frame $\hat{I}_D$. The constructed keypoints $\mathcal{K}(\hat{I}_D)$ should encode (i) the same (frontal facing) head pose as in the source image, (ii) the gaze direction and eye openness of the operator, and (iii) the lower-face expression which is captured by the mouth camera.

Taking (i)-(iii) into account, we therefore build the driving keypoints (as illustrated by "Construct Driving Frame" in Figure 2.3) from

$$\mathcal{K}(\hat{I}_D) = \Pi_S(\, \mathcal{K}_{\mathcal{VR}}(I_M)\,) \oplus \rho(\, \mathcal{K}_{\mathcal{F}}(I_S), \hat{k}_{eye}\,), \tag{2.4}$$

where I_M is the mouth camera image, $\Pi_S(\cdot)$ (see Eq. (2.5)) maps each mouth camera VR keypoint $k_M^{(j)} \in \mathcal{K}_{\mathcal{VR}}(I_M)$ into the space of source image I_S, and $\rho(\cdot)$ replaces the eye keypoints detected in I_S with the modified values $\hat{k}_{eye}$ in order to include the operator's current gaze direction and eye openness. Furthermore, keypoints which encode the head pose are simply copied from the source image. This is sufficient in our use case, since we move the avatar's head display following the operator's head motions (see Figure 2.1).

2.2.4 *Learned Keypoint Mapping* Π_S

Constructing the keypoints of our imaginary driving frame $\mathcal{K}(\hat{I}_D)$ requires mapping keypoints from the mouth camera image to the source image space. A naïve approach would be to simply perform a common translation and scale adjustment on all keypoints together (Schwarz et al., 2021), however, this does not consider perspective differences. One could also attempt to estimate the transformation matrix that maps from the mouth camera to the source camera. This would force us to estimate exact

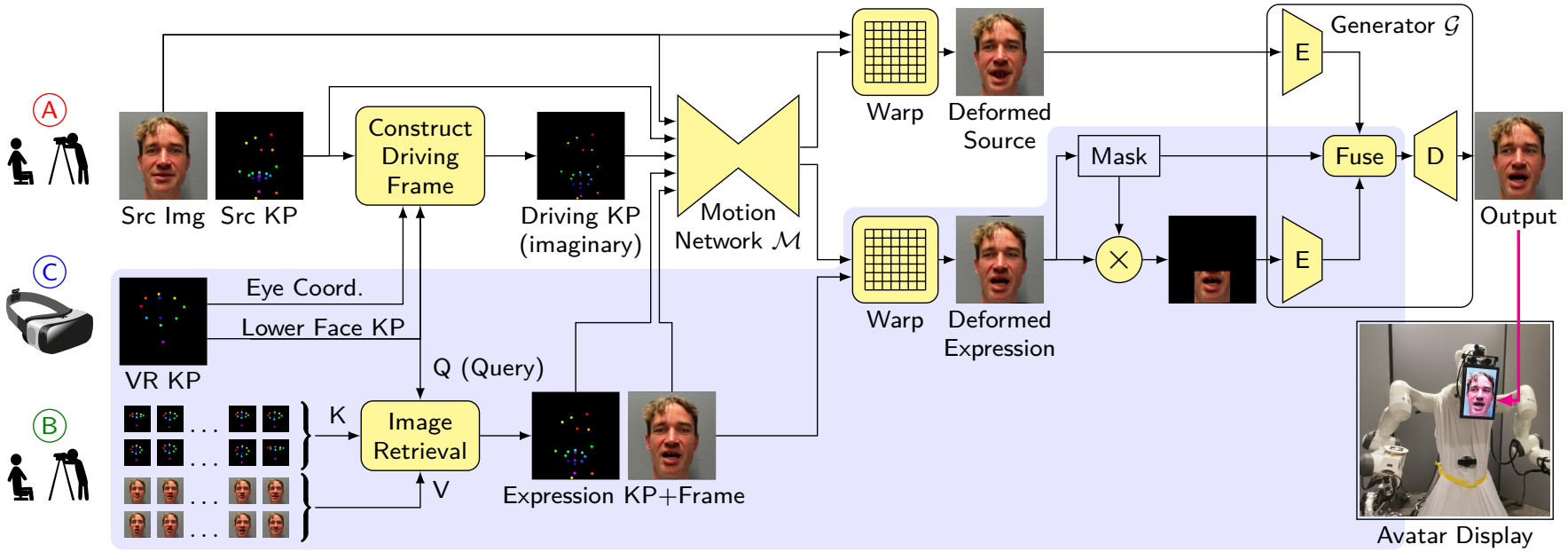

Figure 2.3: Inference pipeline for VR facial animation. We select a still image from a portrait video of the operator shot before the run as source image (A). The remaining frames from the source video are used as a key-value storage of keypoints and corresponding frames (B). The live keypoints from the HMD mouth camera (C) are then approximately projected to the space of each frame in the storage, to retrieve the closest keypoints and corresponding frame (i.e., the expression frame). The source keypoints, a constructed set of driving keypoints, and the retrieved expression keypoints then enter the motion network $\mathcal{M}$, which estimates warping vectors that deform the source image and the expression frame to match the driving keypoints. The two deformed images then enter an encoder-decoder architecture that fuses them and generates the output image.

depth in the mouth image, which is not very robust. Furthermore, there are facial distortions caused by the weight of the HMD, which would not be modeled by such an approach.

Instead, we propose to learn a separate homogeneous transformation matrix ${}^{S}T_{M}^{(j)}$ that maps from each VR keypoint $k_M^{(j)} \in \mathcal{K}_{\mathcal{VR}}(I_M)$ to the corresponding keypoint $\hat{k}_D^{(j)} \in \mathcal{K}_{\mathcal{VR}}(\hat{I}_D)$ that aligns with the head pose of the source image. Estimating this transformation is not trivial, since the VR keypoints exhibit a high variance and learning such a mapping requires corresponding mouth and source image pairs. We therefore capture not just a single source image of the operator, but a whole source video sequence including mouth movements and a similar second video from the mouth camera of the HMD worn by the operator. Note that finding corresponding pairs is challenging since we cannot capture both videos simultaneously.

Because establishing correspondences manually is expensive, we solve this alignment problem approximately by (i) capturing two videos with roughly the same mouth expressions and (ii) iteratively refining the learned keypoint transformations based on the currently associated pairs. This iterative process has the following steps:

1. Extract the keypoints of all source and mouth camera video frames.
2. Initialize each homogeneous transformation ${}^{S}T_{M}^{(j)}$ from the mean scale difference between the two sets of keypoints.
3. Map the keypoints of each mouth image into each source frame space.
4. For each mapped keypoint sequence, search for the best corresponding source frame, yielding N pairs of images.
5. Optimize each ${}^{S}T_{M}^{(j)}$ to minimize the Euclidean distance of the current keypoint pairs and goto step 3) unless the maximum number of 1000 iterations is met.

Note that the optimization runs in approx. 10 s for a reasonable number of 250 calibration images. In order to create robustness against head movements while capturing the source video and changes of the VR headset relative to the operator, we define the mapping in a coordinate system relative to the centroid of I_M and I_S:

$$\Pi_S(\, k_M^{(j)}\,) = {}^{S}T_{M}^{(j)}\,(\, k_M^{(j)} - \overline{\mathcal{K}_{\mathcal{VR}}}(I_M)\,) + \overline{\mathcal{K}_{\mathcal{VR}}}(I_S), \tag{2.5}$$

where $\overline{\mathcal{K}_{\mathcal{VR}}}(I_S), \overline{\mathcal{K}_{\mathcal{VR}}}(I_M)$ are the mean values of the VR keypoints in the source image and the mouth camera image, respectively. As demonstrated in Figure 2.7, this normalization also generates robustness to switching the operator, who is controlling our system, at inference time.

2.2.5 *Extended Inference Pipeline with Image Retrieval*

Whereas the head pose and gaze direction can be encoded well using only keypoints, it is highly challenging to generate proper mouth expressions using only one source image and few keypoints. To address this issue, we capture not just a single source image but multiple source frames in a video. This allows us to dynamically change the source image to the one which is closest to the projected VR keypoints of the current mouth image. In such an approach, the pipeline would then need to apply only small corrections to the facial expression and could, thus, generate more accurate mouth expressions. However, in this naïve setup, non-negligible flickering effects would appear whenever we change the source image.

Instead, we propose to train a modified generator network that allows to decode the information of not just one source image but also the mouth region information of a second source image, which should be closer to the target. Therefore, the primary source image I_S always remains constant while the second source image, which we call the expression frame I_E, can change arbitrarily (see Figure 2.3). This has a positive effect on temporal consistency and counteracts flickering during source image changes significantly.

Following standard information retrieval, we compare the current query

$$Q = \mathcal{K}_{\mathcal{VR}}(I_M)$$

with all keys

$$K = [\,\mathcal{K}_{\mathcal{VR}}(I_{S_1}), \ldots, \mathcal{K}_{\mathcal{VR}}(I_{S_n})\,]$$

via

$$\sum_j ||\Pi_{S_i}(Q^{(j)}) - K_i^{(j)}||_2,\ K_i \in K$$

to retrieve the optimal index of the (image, keypoints) tuples

$$V = [\,(\,I_{S_1}, \mathcal{K}(I_{S_1})\,), \ldots, (\,I_{S_n}, \mathcal{K}(I_{S_n})\,)\,].$$

We modify the pipeline and generator network accordingly:

(i) Use the motion network to generate a deformed image of both the source image and the expression frame $({}^{D}I_S, {}^{D}I_E)$.

(ii) Split the generator into two separate encoder networks, where the first encoder $\mathcal{G}_S^{Enc}$ extracts the source image features

$${}^{D}f_S = \mathcal{G}_S^{Enc}({}^{D}I_S) \tag{2.6}$$

and the second encoder $\mathcal{G}_E^{Enc}$ extracts the expression frame features

$${}^{D}f_E = \mathcal{G}_E^{Enc}(m \odot {}^{D}I_E), \tag{2.7}$$

where $m \in \{0,1\}^{N \times N}$ is a mask that hides all information except for a region around the VR keypoints of ${}^{D}I_{E}$ (see Figs. 2.3 and 2.4).

(iii) Fuse the activation by

$$ {}^{D}f_{S,E} = \frac{m_{\downarrow}}{2} \odot ({}^{D}f_{S} + {}^{D}f_{E}) + (1 - m_{\downarrow}) \odot {}^{D}f_{S}, \tag{2.8} $$

where $\odot$ denotes element-wise multiplication and $m_{\downarrow}$ is a down-scaled version of the binary mask m.

(iv) Decode the fused representation with the decoder $\mathcal{G}^{Dec}$ of the generator network

$$ I_{O} = \mathcal{G}^{Dec}({}^{D}f_{S,E}). \tag{2.9} $$

This additional branch to our inference pipeline is visualized in the blue-marked area of Figure 2.3.

When capturing the two videos (with and without HMD), it is important to have a high variety of mouth expressions in the set of source frames. We therefore propose to capture two videos of the operator, while reading a sentence that covers many phonemes. The sentence "That quick beige fox jumped in the air over each thin dog, look out he shouts for he's foiled you again, creating chaos", is known to match these requirements and gave us good results during testing.

2.2.6 *Training*

The training pipeline is illustrated in Figure 2.4.

During inference, we use information retrieval to select the current expression frame, based on the optimized keypoint transformations ${}^{S}T_{M}$. During training, however, we use the VoxCeleb dataset (Nagrani et al., 2017) (prepared using the video preprocessing code from Siarohin et al. (2019b)), which mainly consists of celebrities being interviewed. In this setup, it is not trivial to select an expression frame: If we would just set the expression frame to be the driving frame itself, the network would learn to simply copy the information.

To avoid this, we select the expression frame from a small interval around the driving frame. This makes the assumption that temporally close frames also exhibit similar expressions.

The generator network (especially the feature encoder sub-modules) can thus learn to mostly ignore the lower-face region of the motion-transmitted source image, since the expression frame is generally closer to the driving frame. This, however, leads to temporal instabilities whenever the expression frame is switched. We counteract this by choosing a reasonable interval around the driving frame and augment the chosen expression frame using color jittering and injection of several types of random noise as proposed by Carlson et al. (2018). We argue that in this setup, the generator

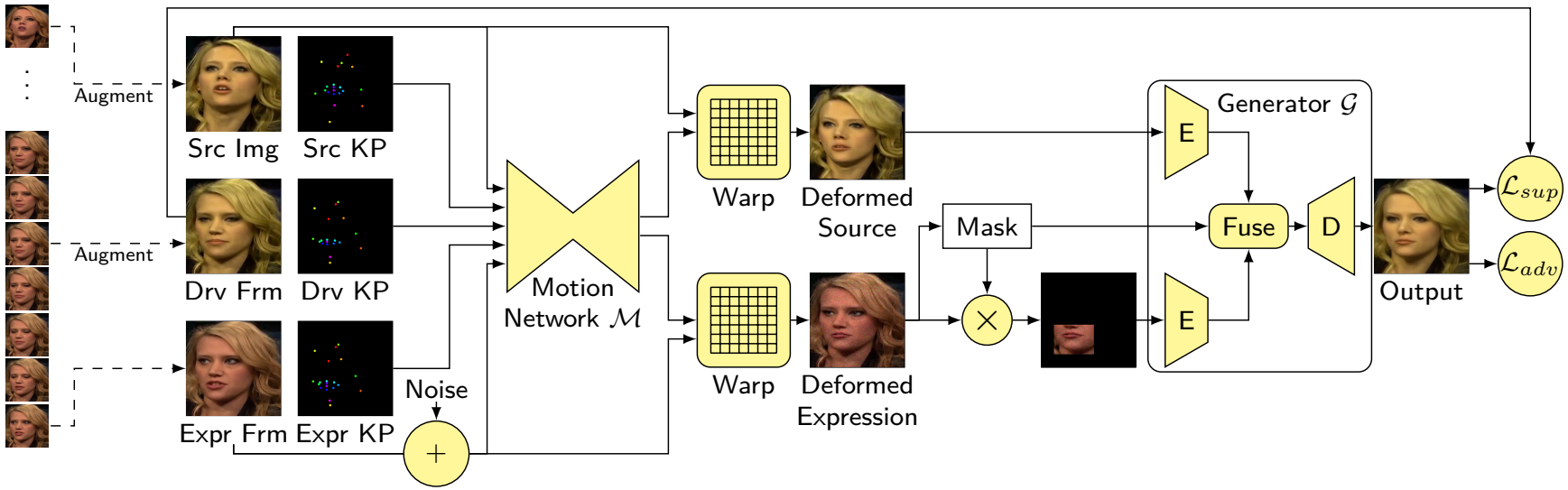

Figure 2.4: Training the VR facial animation network from videos. The training loss $\mathcal{L}_{sup}$ is minimized when the network reconstructs the driving frame (Drv Frm) from source image (Src Img) and keypoints (Src KP), as well as the driving keypoints (Drv KP). Following Siarohin et al., 2019b, we generate adversarial losses ($\mathcal{L}_{adv}$) using a keypoint-aware discriminator network. The expression frame (Expr Frm) is chosen from a close time interval around the driving frame and is available as auxiliary input which is already close to the target expression.

network is now explicitly guided to keep the deformed source image information ${}^{D}I_S$ to generate a proper facial animation.

We train the pipeline end-to-end using perceptual loss and employ a keypoint-aware discriminator network for generating adversarial losses, similar to Siarohin et al. (2019b).

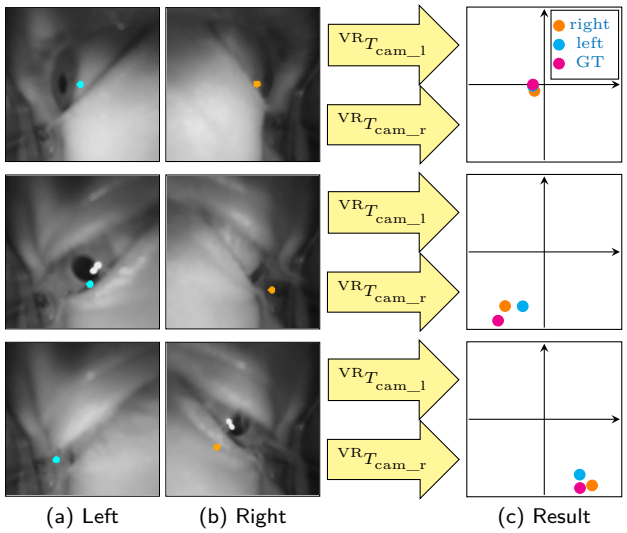

Figure 2.5: Eye tracking. We show the eye keypoints in the left and right image frame, which are learned without direct supervision (a, b). The resulting transformed gaze direction prediction is shown in (c) in cyan and orange, with the ground truth in magenta. The transformations ${}^{VR}T_{cam}$ are learned.

2.2.7 *Eye Tracking and Animation*

We introduce an image-driven eye tracking pipeline that needs to be calibrated once and can be trained in less than a minute for an operator. To obtain training data, we request the operator to follow with their gaze a red dot that moves in the VR display. This gives us calibration/training triplets of left eye images, right eye images and 3D gaze directions.

Network Architecture. We build a very lightweight hourglass network (Newell et al., 2016) with only two downsampling and two upsampling layers that takes an input eye image and outputs a heatmap which is used to generate a single keypoint (see Fig. 2.5). We map this keypoint coordinate into the VR space using a learned homogeneous transformation ${}^{VR}T_{cam}$, which is jointly optimized with the hourglass network. While the homogeneous transformation is trained with supervision, we train the hourglass end-to-end in a self-supervised manner.

Inference. During inference, we take the mean prediction p of both the left eye prediction $p_L \in [-1,1]^2$ and the right eye prediction $p_R \in [-1,1]^2$. We, furthermore, estimate a normalized confidence measure

$$c = 1 - \frac{1}{2\sqrt{2}}||p_L - p_R||_2 \in [0,1] \tag{2.10}$$

which is large when both predicted eye coordinates are close to each other.

At this stage, only the recognition of the eye openness remains. We found that the eye openness strongly correlates with the gaze direction, i.e. the more an operator looks down the less open the eyes are. This property was also learned by our networks. To simulate both eyes, the networks only use one keypoint at the upper eyelid directly above the pupil center of the left eye (see Fig. 2.6). During inference, we can thus control the gaze and eye openness by modifying a single keypoint of the source image.

The assumptions above apply as long as the eyes are not completely closed, i.e. not when blinking. It is beneficial to detect this case without requiring additional annotations. Our experiments showed that whenever the eyes are closed, there is a very low confidence value. This is due to the fact that the gaze direction predictions of both eyes are very different, since we did not equip the networks with the capabilities of handling such cases. Hence, we detect closed eyes whenever the confidence parameter is below a threshold λ_C. This also has the benefit of hiding implausible eye configurations from the viewer by showing the operator with closed eyes.

Eye Coordinate System. One remaining problem is still to define the region in the source image in which the eye keypoint can move, i.e. a coordinate system mapping. One possibility is to automatically find these boundaries, by capturing a second video of the operator (without the HMD) in which the eyes are moving.

To decrease required capture time, we built an interactive annotation tool, which renders the source image with eye keypoints at the current cursor position. This allows us to manually define the boundaries of the possible eye keypoints in the source image, which are illustrated in Figure 2.6. The annotated boundaries then define a normalized coordinate system which is centered in the frontal-facing gaze direction.

2.2.8 *Temporal Consistency Filters*

We apply several filters to the image retrieval process and facial keypoints to enhance the temporal consistency.

Expression Frame Filtering. We apply a straightforward filter that regulates the expression frame retrieval process: We only allow to change the expression frame I_E, when there is another frame $1+\lambda_{swap}$ times closer to the current projected VR keypoints. This hysteresis avoids fast switching.

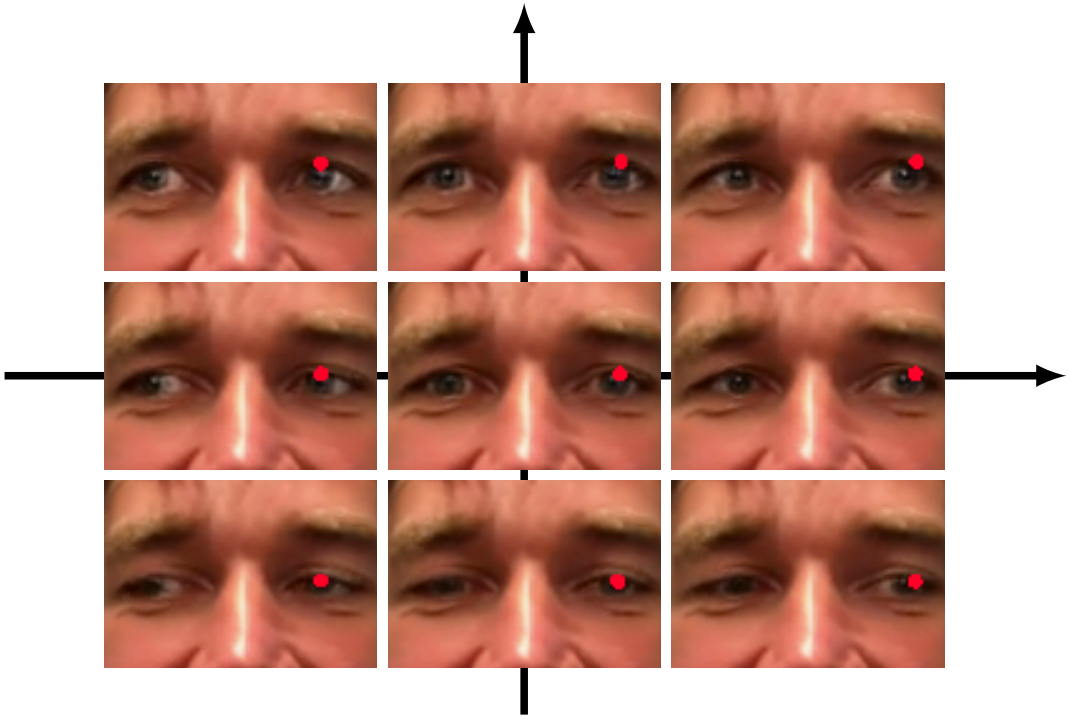

Figure 2.6: Illustration of the eyes reacting to keypoint manipulations. The center image defines the origin and the other images define the boundaries of the normalized eye coordinate system.

We also apply a recursive low-pass filter parameterized by $\lambda_E, \lambda_{\tilde{E}}, \lambda_O$. Instead of simply using the raw expression frame I_E, we propose to build the current expression frame $I_{\tilde{E}}^{(t)}$ at time step t according to

$$I_{\tilde{E}}^{(t)} = \lambda_E I_E^{(t)} + \lambda_{\tilde{E}} {}^{E}I_{\tilde{E}}^{(t-1)} + \lambda_O {}^{E}I_O^{(t-1)}, \tag{2.11}$$

where $I_E^{(t)}$ is the current (raw) expression frame, ${}^{E}I_{\tilde{E}}^{(t-1)}$ is the last expression frame $I_{\tilde{E}}^{(t-1)}$ (recursively) deformed to the current expression frame using the motion network, and ${}^{E}I_O^{(t-1)}$ is the last prediction $I_O^{(t-1)}$ deformed to the current expression frame, respectively. In practice, we choose $\lambda_E = 0.7$, $\lambda_{\tilde{E}} = 0.1$, and $\lambda_O = 0.2$.

Eye Keypoint Filtering. As explained in Sec. 2.2.7, we can obtain a confidence value of the current eye position by comparing the left and right eye. We then recursively low-pass filter the eye position and parametrize the filter with the confidence $c \in [0, 1]$. We propose to derive the filtered eye position $\tilde{p}$ from

$$\tilde{p}^{(t)} = \lambda_G \, p^{(t)} + (1 - \lambda_G) \, \tilde{p}^{(t-1)}, \tag{2.12}$$

where we determine the contribution $\lambda_G \in [0, 1]$ of the current eye position estimate $p^{(t)}$ according to:

$$\lambda_G = \begin{cases} 5c - 4, & \text{if } c \geq 0.8 \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

This low-pass filter is especially useful when the prediction p of the eye tracking network is noisy.

2.3 Experiments

We report several qualitative results and discuss our performance at the ANA Avatar XPRIZE semifinals.

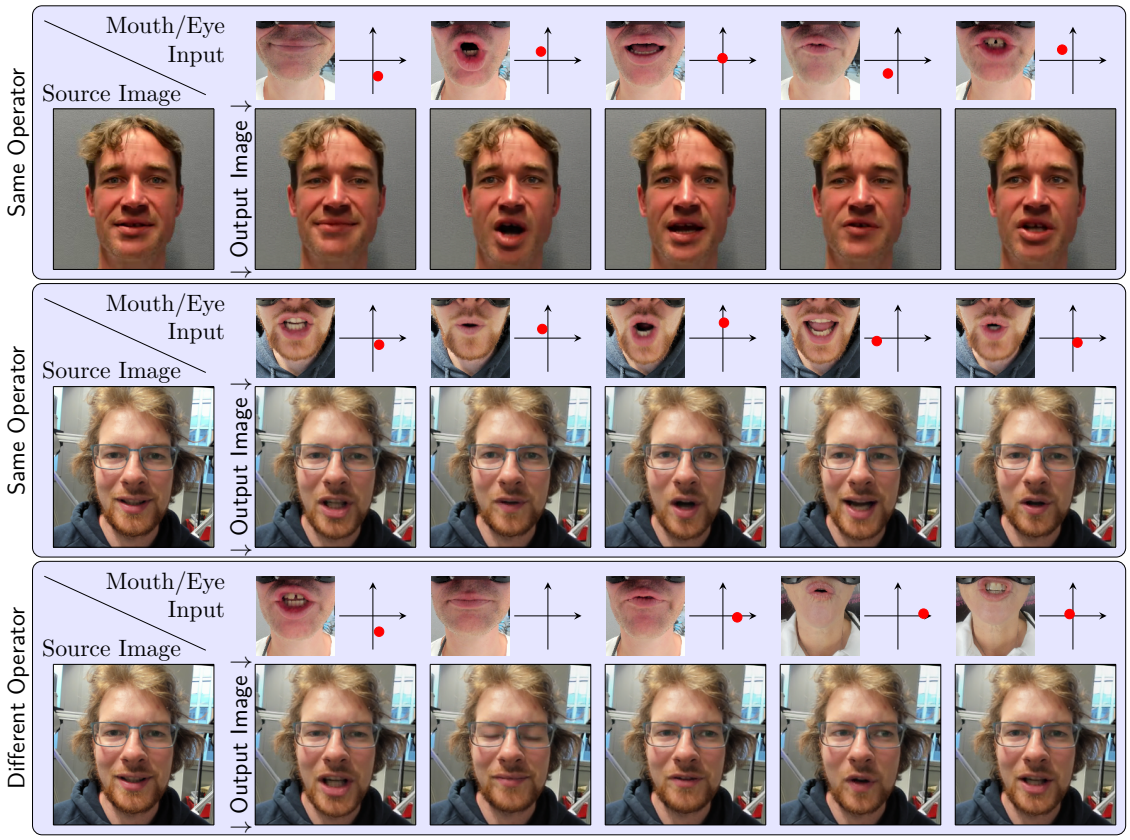

Figure 2.7: Generated faces. In each box we use the same source image (left) to generate a facial reconstruction given the mouth camera image and eye coordinates. Note how the system matches mouth and eye configurations closely (all rows) and even performs inter-person animation (bottom row).

2.3.1 *Qualitative Results*

VR Facial Animation. Figure 2.7 illustrates exemplary results of our full pipeline when (i) mapping from one operator to the appearance of the same operator and (ii) mapping to the appearance of another operator. Note that our complete forward pass runs with 33 frames per second (fps) on a single NVIDIA RTX 3090 graphics processing unit (GPU) and an image resolution of 256×256.

Image Retrieval. Figure 2.8 visualizes the accuracy of our proposed keypoint-based image retrieval. Experiments have shown that image retrieval from a set of source images allows us to generate much more accurate mouth expressions compared to just a single source. As shown in Figure 2.8, the image retrieval also works well between different operators.

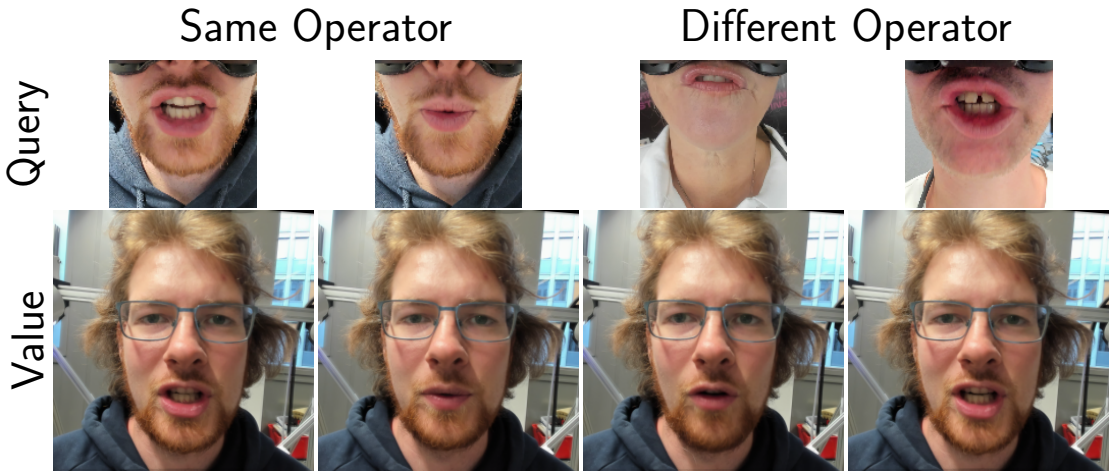

Figure 2.8: Image retrieval process experiments. Unlike the first two columns, the last two columns are examples for image retrievals where the operator keys and values are different from the operator in the query.

2.3.2 *The ANA Avatar XPRIZE Semifinals*

At the semifinals of the ANA Avatar XPRIZE competition, three scenarios had to be accomplished with our avatar system by a previously unknown operator. The scenarios were repeated in a second run, with the better score persisting. Before each run, a preparation time of one hour was allotted in which we could introduce the operator to the system. We also used this time to prepare the facial animation pipeline, including recording two source videos (with and without the HMD), optimizing the keypoint mapping, calibrating/training the eye tracking pipeline, and annotating eye coordinates using our interactive tool. On average, operator adaptation took us about 15 minutes, which could mostly be done in parallel to the general operator introduction. The scenarios all required the operator to interact though the avatar system with another person from the jury, i.e. the recipient. For the recipient, facial

animation is particularly important, as it helps to convey the focus of attention and the emotions of the operator and confirms the recipients that they are interacting with a real person.

XPRIZE defined rigorous scoring criteria. Specific criteria regarding facial animation were:

1. The Recipient was able to identify the remote Operator and felt the Operator was present in space.
2. The Recipient was able to understand the Operator's emotions through the Avatar.
3. The Recipient felt a sense of shared experience with the remote Operator.
4. The Recipient was able to understand the Operator's gestures through the Avatar.
5. The Operator was able to express their emotions.

Our team NimbRo performed very well at the these and all other criteria and was ranked first in the semifinals with an overall score of 99 out of 100 points. Figure 2.1 shows our avatar system and facial animations during the challenge.

2.4 Conclusion

We have demonstrated an efficient and real-time capable VR facial animation pipeline that generalizes well to operators unknown *a proiri* with a fast adaptation process. Our model can be trained on large-scale talking-head datasets, bypassing the alignment problem between images capturing the complete face and images from the HMD cameras. During inference, the target lower-face (VR) keypoints, which encode the lower facial expression are constructed from mouth camera keypoints that are mapped using transformations learned with iterative refinement. The learned keypoint mapping is also utilized to dynamically retrieve an expression frame, which is used to guide the animation in the mouth region. Furthermore, we demonstrated how to improve temporal consistency. Using the proposed method, our team reached an almost perfect score of 99/100 points and was ranked first in the ANA Avatar XPRIZE semifinals event in Miami, FL with a total of 28 teams. A useful extension of our method is to directly use visual features from the mouth image captured by the mouth camera. Direct visual mouth camera guidance is investigated in Chapter 3 of this thesis.

3

Attention-Based VR Facial Animation with Visual Mouth Camera Guidance

Preface

This chapter is adapted from the following publication:
Andre Rochow, Max Schwarz, and Sven Behnke (2023). "Attention-based VR facial animation with visual mouth camera guidance for immersive telepresence avatars." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1276–1283. DOI: 10.1109/IROS55552.2023.10342522, ©2023 IEEE, presented in Detroit, MI, USA.

Statement of Personal Contribution

The author of this thesis is the main author of the publication Rochow et al. (2023) and significantly contributed to all aspects of the publication and this chapter. These contributions particularly include proposing, conceptualizing, designing, and implementing the method, conducting a comprehensive literature review, training the models, evaluating the method, analyzing and interpreting the results, as well as active involvement in the ANA Avatar XPRIZE competition. Another contribution of the author was the collection and annotation of data required for finetuning the models. Furthermore, the author of this thesis was the primary writer of the paper and was the main contributor to the creation of the visualizations.

VR facial animation aims to control the facial motion of a specific person using input signals from sensors mounted on an HMD. In our case, the appearance of an individual is determined by source images captured without the use of an HMD. Particular challenges of VR facial animation are that the visual input information is often limited or occluded by the HMD and has a large perspective difference to the desired output animation. Furthermore, the weight of the HMD sometimes causes deformations even in the visible face parts, which can limit the mouth movements of the person wearing the HMD. Due to the alignment problem of HMD sensor data (such as mouth camera images) and facial images without the presence of an HMD, it is impossible to capture perfect ground truth data. Training VR facial animation models is therefore exceptionally challenging.

The VR facial animation system presented in this chapter was developed for the ANA Avatar XPRIZE competition[1] finals, where a previously unknown operator had to perform various tasks through our avatar robot system (see Fig. 3.1). Our robotic system (Lenz et al., 2025; Pätzold et al., 2023; Schwarz et al., 2023) consists of an operator station with a VR headset and arm exoskeletons, as well as an avatar robot. As in Chapter 2, we use a modified *Valve Index* HMD equipped with two eye cameras and a mouth camera (see Fig. 2.2). As visualized in Fig. 3.1, the operator's face is animated on a display that mirrors the operator head movement using a 6 DoF robotic arm. At the competition participants were judged not only on task performance, but also on immersion and the communication experience of a remote recipient. In particular, points were awarded when the operator was able to convey emotional cues to the recipient. Facial animation was thus a cornerstone of our strong performance at the competition finals in November 2022, where our team NimbRo won the first prize.

In Chapter 2, we formulated VR facial animation as a keypoint-driven face reenactment problem. This allowed us to train on large talking-head datasets and leverage knowledge obtained from many different appearances to animate unseen persons. In particular, we guided animation of the mouth region by dynamically retrieving a second source image (called expression frame) based on its keypoint similarity to the mouth camera image. Unfortunately, temporal inconsistencies occurred whenever changing the expression frame and the animation accuracy is strongly depended on the image retrieval quality. Furthermore, keypoint ambiguities limit the range of possible expressions.

In this chapter, we propose an extension to address these limitations while preserving the ability to generalize to unseen persons. We propose to utilize multiple source images with an attention mechanism driven by the mouth camera, enabling our method to dynamically weight relevant features. The method can be employed either with fixed source images, or with the last source image treated as a retrievable dynamic image, like the expression frame in Chapter 2. Even in the latter case,

1 https://www.xprize.org/prizes/avatar
2 https://www.youtube.com/watch?v=0D2UbZNw9sQ

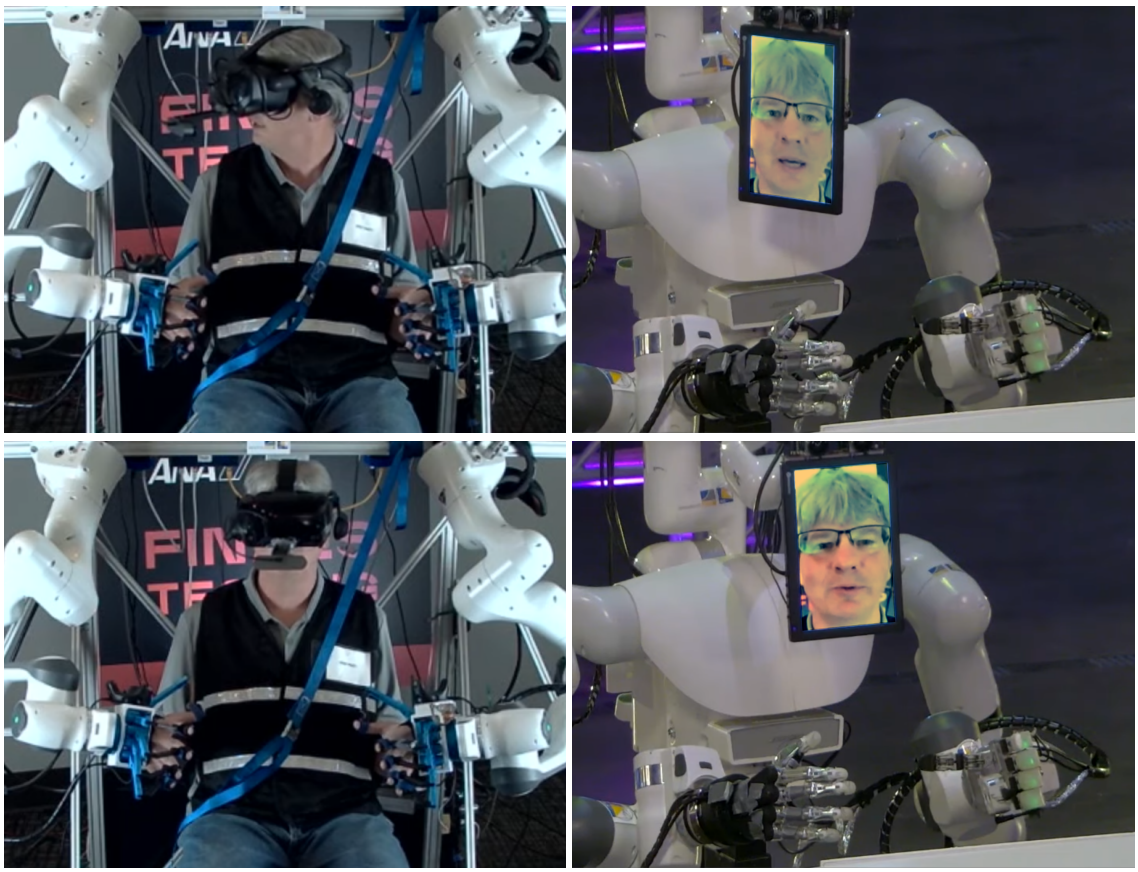

Figure 3.1: Facial animation of an operator interacting with a recipient through the NimbRo Avatar system at the ANA Avatar XPRIZE finals. Stills from our winning run[2]. Contrast was enhanced for easier viewing.

the use of multiple source images effectively mitigates the negative impact of the retrieved source image on temporal consistency.

We enhance the range of possible facial expressions and resolve keypoint ambiguities by introducing a mouth camera guidance that directly utilizes visual mouth camera features. As discussed, the alignment problem makes it very challenging to generate suitable training data. We adress this issue by proposing an efficient way to keep training on large talking-head datasets for generalizability during inference and additionally annotate a few image pairs with similar mouth expressions in the mouth and face camera that we merge into the training process. As we demonstrate in a detailed evaluation and the supplementary video[3], our VR facial animation pipeline generates more accurate and more temporally consistent results than our method presented in Chapter 2, with more movement in areas that are not associated with keypoints, such as the cheeks.

In addition to a real-time capable VR facial animation pipeline, our contributions include: (i) a source image attention mechanism that significantly improves temporal consistency and facial animation accuracy, (ii) an efficient way to leverage visual

3 https://www.ais.uni-bonn.de/videos/IROS_2023_Rochow

a) Ours b) Mesh c) No VR d) Stitched e) Cartoon

Figure 3.2: Types of facial animation at ANA Avatar XPRIZE finals. Examples from teams: a) NimbRo (first place), b) Pollen Robotics (second), c) Northeastern (Luo et al., 2022) (third), d) AVATRINA (Marques et al., 2022) (fourth), e) UNIST (sixth).

mouth camera information to resolve keypoint ambiguities and model a broader range of facial expressions, and (iii) emulation of mouth camera data, which allows training on available large-scale talking-head datasets.

3.1 Related Work

Face Reenactment. A task related to VR facial animation is face reenactment. Here, the head pose and facial expression encoded in a driving frame is to be visualized with the appearance given by a source image person. Often keypoints are used to represent the motion (Siarohin et al., 2019a,b; Zhao et al., 2021a). A motion network predicts a deformation grid to deform source images into a defined target motion. Siarohin et al. (2019b) propose to use image feature-based local affine transformations in the motion network that allow to model complex motions. Gafni et al. (2021) propose to condition a dynamic NeRF (Mildenhall et al., 2020) with motion information extracted from driving frames.

VR Facial Animation. In VR facial animation, the motion is often encoded in eye camera images and a mouth camera image (Lombardi et al., 2018; Wei et al., 2019, Chapters 2 and 3) or even in audio recordings (Richard et al., 2021a). Lombardi et al. (2018) render a virtual avatar by utilizing a VAE that can be conditioned with motion parameters obtained from the HMD. They train a second VAE on real and synthetic mouth camera images and map similar expressions of both domains to similar latent codes by manually controlling the latent variable that determines the domain. However, they do not handle facial deformations caused by the HMD explicitly. Wei et al. (2019) generate synthetic ground truth data with an expression-preserving style transfer network, which maps between the mouth camera domain and the avatar domain. Richard et al. (2021a) bypass the alignment problem by omitting mouth camera images completely and using audio instead. They generate impressive results; however, the reduced amount of information significantly limits the expressivity. Especially when the user is silent, the animation task is ill-posed. Unfortunately, all these methods need a significant amount of person-specific data

capture and person-specific training, which makes them unsuitable for use-cases that require instant application, such as the ANA Avatar XPRIZE competition.

We note that additional related work, which was published after our VR facial animation methods from this chapter and Chapter 2, is described in Sec. 4.5.1.

From ANA Avatar XPRIZE finals video footage we recognize five categories of face animation techniques used by participants (see Fig. 3.2). Out of the 12 teams selected for the two competition days, three teams had no VR headset. In this case, video streaming suffices for face display, but operator immersion is limited. Very similarly, one team displayed mouth camera footage directly, stitched together with previously captured footage of the operator's eyes. The rest of the teams used expression information from mouth trackers and/or audio to animate either 2D emoji drawings (three teams) or rendered 3D meshes, adapted to roughly match the operator's attributes like hair color and gender (four teams). Our team was the only one to produce a photorealistic animated face image.

3.2 Method

Our proposed method is an extension of the one presented in Chapter 2, which is referred to as the baseline. The basic modules and steps remain, with important functionalities added into the pipeline and generator network (see Fig. 3.3).

3.2.1 *Recap of the Baseline Inference Pipeline*

We provide a short recap on the basic steps of our baseline inference pipeline from Chapter 2, which is visualized in Fig. 2.3. It is composed of 1) capturing and preprocessing, 2) image retrieval, 3) construction of the driving keypoints, 4) deforming, and 5) fusing and refining.

1) Capturing and Preprocessing: We capture two videos of the operator, with and without the HMD, respectively. The mouth camera only captures the lower facial region and the second (source image) video captures the complete front-facing head of the operator. Both videos are recorded during speech, capturing different expressions. From the source video, we select an arbitrary source image I_S which subsequently defines the operator appearance. Keypoints are extracted from all frames of both videos. We differentiate between VR keypoints $\mathcal{K}_{\mathcal{VR}}$ (consisting of mouth keypoints and one chin keypoint that represent the lower facial expression), which are also visible in the mouth camera, and keypoints extracted from $\mathcal{K}_{\mathcal{F}}$ which primarily encode the head pose and information about the eyes. Specifically, all eye information (i.e., gaze direction and eye openness) is encoded in a single keypoint (see Fig. 2.6) called k_{eye}. Given the set of mouth camera video VR keypoints $\{\mathcal{K}_{\mathcal{VR}}(I_{M_0}), \mathcal{K}_{\mathcal{VR}}(I_{M_1}), \ldots, \mathcal{K}_{\mathcal{VR}}(I_{M_k})\}$ and source video VR keypoints $\{\mathcal{K}_{\mathcal{VR}}(I_{S_0}), \mathcal{K}_{\mathcal{VR}}(I_{S_1}), \ldots, \mathcal{K}_{\mathcal{VR}}(I_{S_m})\}$, we optimize a keypoint mapping $\Pi_{S_i}(\mathcal{K}_{\mathcal{VR}})$ that maps VR keypoints from the mouth camera into the space of source frame I_{S_i} from the captured source video (see Sec. 2.2.4). $\Pi(\cdot)$ corrects

effects caused by the perspective change and deformations caused by the HMD's weight. Other preprocessing steps include the manual definition of an eye coordinate system in the source image and the training of the eye tracking model, including the acquisition of calibration/training data (see Sec. 2.2.7).

2) Image Retrieval: The image retrieval process searches the source video for a so-called expression frame that has a similar mouth expression as the live mouth camera image. Given the mapped mouth camera keypoints $\Pi_{S_i}(\mathcal{K}_{\mathcal{VR}}(I_M))$, we therefore retrieve the source video frame I_{S_i} with the best matching keypoints (see Sec. 2.2.5). The retrieved expression frame is then utilized to guide the animation process in the mouth region.

3) Construction of the Driving Keypoints: The keypoints $\mathcal{K}(\hat{I}_D)$ of an imaginary driving frame that specify the facial target expression and pose are constructed here (see Sec. 2.2.3). The keypoints that determine the head pose are simply copied from the source image I_S since we move the face display on the robot and thus do not require head movement in the output animation. The gaze keypoint $\hat{k}_{eye}$ is estimated by transferring eye tracking results from the eye cameras into the normalized gaze coordinate system defined in the source image (see Sec. 2.2.7). The VR keypoints $\mathcal{K}_{\mathcal{VR}}(\hat{I}_D)$ are generated by mapping the current mouth camera keypoints into the source image space. The imaginary driving keypoints are therefore defined as

$$\mathcal{K}(\hat{I}_D) := \Pi_S(\mathcal{K}_{\mathcal{VR}}(I_M)) \oplus \rho(\mathcal{K}_{\mathcal{F}}(I_S), \hat{k}_{eye}), \tag{3.1}$$

where I_M is the mouth camera image, $\Pi_S(\cdot)$ maps each VR keypoint $k_M^{(j)} \in \mathcal{K}_{\mathcal{VR}}(I_M)$ into the space of source image I_S, $\oplus$ denotes the concatenation operation, and $\rho(\cdot)$ replaces the eye keypoint detected in I_S with the modified values $\hat{k}_{eye}$ in order to include the operator's current gaze direction and eye openness.

4) Deforming: The motion network $\mathcal{M}$ generates deformation grids $\mathcal{M}_{S\leftarrow D}$ and $\mathcal{M}_{E\leftarrow D}$ that are used to sample a deformation of the source image and expression frame into the imaginary driving keypoints (see Sec. 2.2.2). The motion network cannot generate new content, but it generates a good initialization for the generator network.

5) Fusing and Refining: The generator network $\mathcal{G}$ combines the deformed source image and the mouth region of the deformed expression frame (see Sec. 2.2.5). It generates a realistic output image with the appearance of the source image, as well as the facial expressions and head pose specified by the constructed imaginary driving keypoints.

3.2.2 *Source Image Attention Mechanism*

We address temporal inconsistencies, as occurring in our baseline method from Chapter 2, by using more than two source images and introducing an attention

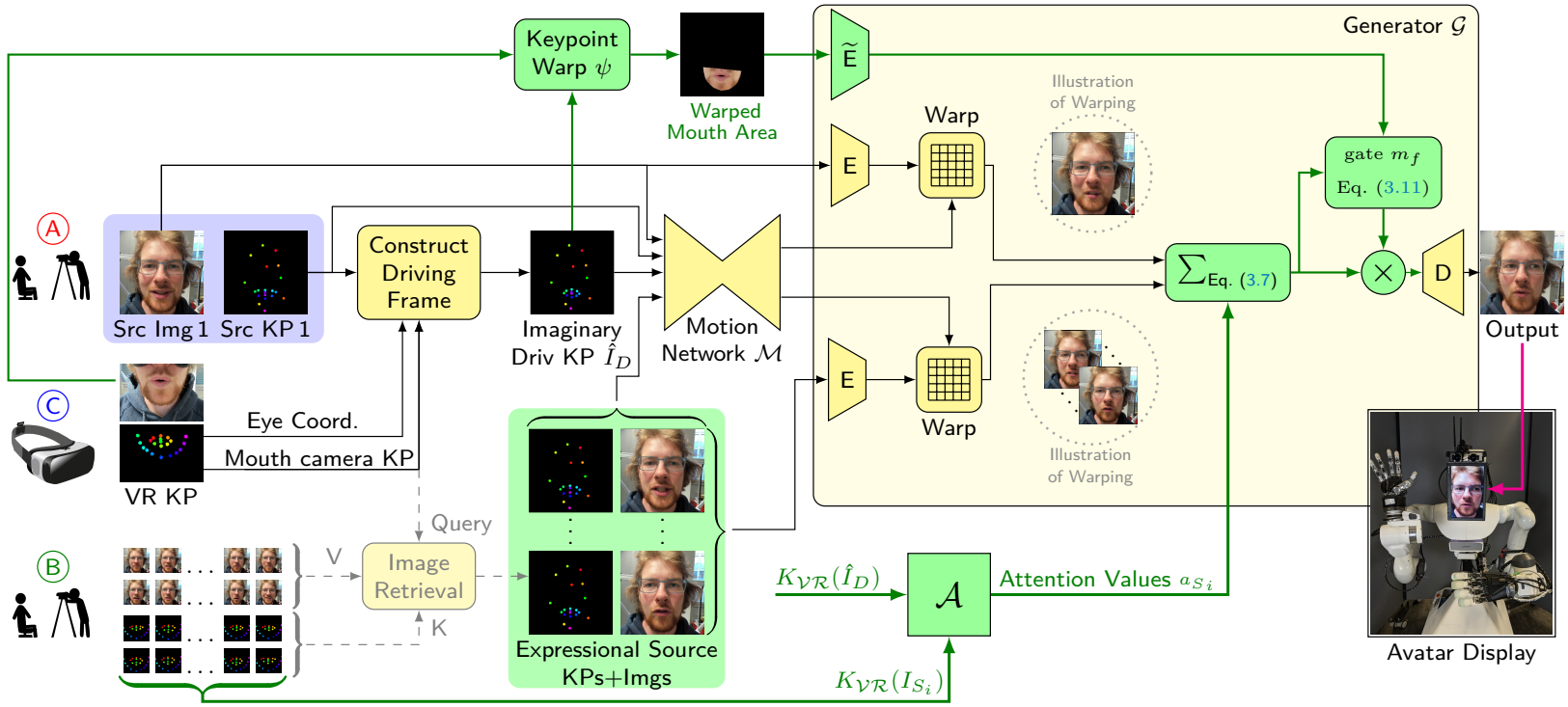

Figure 3.3: Inference pipeline for the extended VR facial animation method. New components compared to our previous method (Chapter 2) are highlighted in green. We select 4-5 still source images from a portrait video of the operator shot before the run as source images (A). The remaining frames are optionally used as a key-value storage of retrievable expression keypoints and corresponding images (B). The live keypoints measured inside and outside the VR headset (C) are then projected to the first source image frame, where they are optionally used to retrieve the closest expression frame with keypoints from the storage. The keypoints of all source images including the retrieved one and a constructed set of driving keypoints then enter the motion network $\mathcal{M}$, which estimates a deformation grid that is used to warp the source images features, extracted by the generator-encoder network, to match the driving keypoints. The illustration of warping in $\mathcal{G}$ shows the deformation grid applied to the image instead of encoded features. The deformed features are aggregated over the source images in the lower facial region using a trainable attention mechanism $\mathcal{A}$. The mouth camera image from the HMD is warped into the lower facial area of the constructed driving keypoints and then encoded by a separate encoder network $\tilde{E}$. An estimated mask m_f gates the aggregated deformed source features using the warped mouth camera features. The masked aggregated features are then decoded to produce the output.

mechanism that equips the network with the ability to decide on how much information it requires from each source image. The attention mechanism works in several stages. We distinguish between two types of input images, the appearance (or first) source image I_S (or I_{S_1}) and the expressional source images $I_{S_2}, I_{S_3}, \ldots, I_{S_n}$. The first source image conserves all the appearance information of the operator, whereas the expressional source images are used to generate more accurate animations, by presenting the network different variations of the lower facial region of an operator. Especially the mouth area has a lot of variations due to occlusions, disocclusions and a many degrees of freedom when speaking. Given the selected source images $I_{S_1}, I_{S_2}, \ldots I_{S_n}$, we extract the corresponding keypoints $\mathcal{K}_{\mathcal{F}}(I_{S_1}), \mathcal{K}_{\mathcal{F}}(I_{S_2}), \ldots, \mathcal{K}_{\mathcal{F}}(I_{S_n})$ and VR keypoints $\mathcal{K}_{\mathcal{VR}}(I_{S_1}), \mathcal{K}_{\mathcal{VR}}(I_{S_2}), \ldots, \mathcal{K}_{\mathcal{VR}}(I_{S_n})$.

For a sequence of VR keypoints $k_{S_i}^{(j)} \in \mathcal{K}_{\mathcal{VR}}(I_{S_i})$ of the source image I_{S_i}, we generate a distance tensor $\mathcal{D}_{S_i}$ with

$$\mathcal{D}_{S_i}^{q,r} = \frac{k_{S_i}^{(q)} - k_{S_i}^{(r)}}{\max \mathcal{D}_{S_i}} \in \mathbb{R}^2. \tag{3.2}$$

The distance tensor $\mathcal{D}_D$ is generated for the driving keypoints $\mathcal{K}_{\mathcal{VR}}(I_D)$ analogously. We then estimate similarity vectors of the source distance tensors

$$\vec{x}_{S_i} = \vec{\mathcal{D}_{S_i}} W^S \in \mathbb{R}^{256} \tag{3.3}$$

and the driving distance tensor

$$\vec{x}_D = \vec{\mathcal{D}_D} W^D \in \mathbb{R}^{256}, \tag{3.4}$$

where $W^S, W^D \in \mathbb{R}^{d,256}$ are learned weight matrices and $\vec{\mathcal{D}}$ represents a flattened vector representation of a distance tensor. The similarity values are finally given by the scaled dot products

$$x_{S_i} = \frac{\vec{x}_{S_i} \vec{x}_D^{\,T}}{\sqrt{256}} \in \mathbb{R}, \tag{3.5}$$

which are fed into a softmax function to generate attention values $a_{S_i} \in \mathbb{R}$. These steps are summarized with $\mathcal{A}$ in Figs. 3.3 and 3.4.

Before we calculate the weighted sum we extract features $E_{S_i} = E(I_{S_i})$ of all source images I_{S_i}, using the generator encoder network E (see Fig. 3.3), and align them in the driving keypoints. This is achieved by deforming the features into the driving keypoints using the deformation grid $\mathcal{M}_{S_i \leftarrow D}$ estimated by the motion network. The deformation generates a roughly aligned feature representation

$${}^{D}E_{S_i} = \mathcal{M}_{S_i \leftarrow D}[\, E(I_{S_i})\,]. \tag{3.6}$$

The aggregated deformed source image features

$$^{D}E_S = (1 - B_{LF})\,^{D}E_{S_1} + \sum_{i=1}^{n} a_{S_i} B_{LF}\,^{D}E_{S_i}, \tag{3.7}$$

are generated by a weighted sum in the lower facial region, where B_{LF} is a binary mask that crops out the lower facial region of the deformed source images features and a_{S_i} are the attention values.

3.2.3 *Visual Mouth Camera Guidance*

In order to capture the lower-face outline we extract 10 additional jaw outline keypoints. The VR keypoints of image I with all 11 jaw outline keypoints (including the chin keypoint that is already in $\mathcal{K}_{\mathcal{VR}}(I)$) are defined as $\mathcal{K}'_{\mathcal{VR}}(I)$ and referred to as the full VR keypoints. Note that the optimized keypoint mapping Π_S in this chapter is also capable of mapping all jaw outline keypoints from the HMD mouth camera image into the source image space.

We address keypoint ambiguities by leveraging visual information from the current mouth camera image to guide the animation process. It is challenging to directly process the mouth camera image due to perspective changes and deformations caused by the HMD.

Our key idea for addressing this issue is to use the obtained full VR keypoints from the mouth camera image $\mathcal{K}'_{\mathcal{VR}}(I_M)$ and its projection $\Pi_S(\,\mathcal{K}'_{\mathcal{VR}}(I_M)\,)$ into the target head pose (i.e., the head pose of the first source image I_S). We first estimate a Delaunay triangulation and then use barycentric coordinates to sample the mouth camera image in the target keypoints. We define the mouth area keypoint warping

$$\psi(\,I_1, \mathcal{K}'_{\mathcal{VR}}(I_1), \mathcal{K}'_{\mathcal{VR}}(I_2)\,) \tag{3.8}$$

to be a function that samples the image I_1 with keypoints $\mathcal{K}'_{\mathcal{VR}}(I_1)$ in the keypoints $\mathcal{K}'_{\mathcal{VR}}(I_2)$ of image I_2. If we set $I_1 = I_M$ and $I_2 = I_D$ this gives us an approximation that accounts for the perspective change and the deformations caused by the HMD.

3.2.3.1 *Mouth Camera Emulation during Training*

Unfortunately, the alignment problem of mouth camera images and complete faces without an HMD makes it impossible to obtain perfect ground truth pairs for training. In our previous approach (see Chapter 2), the information bottleneck posed by the VR keypoints enables training on large-scale talking-head datasets which helps generalization to unseen persons without finetuning.

To maintain this behavior and still provide visual mouth camera information, we propose a training-time data augmentation scheme. We add different types of camera noise (Carlson et al., 2018), but also simulate imperfect transformation by performing a keypoint warping on the driving frame to itself ($I_1 = I_2 = I_D$) with

noise added to the keypoints (see Fig. 3.4). The noise-augmented keypoint sequences are given by augmenting with (i) a normal distributed random scaling factor of the keypoint vector, (ii) a normal distributed random translation of the keypoint vector, and (iii) a normal distributed offset for each keypoint in the vector.

During training, the resulting keypoint warping function

$$\psi(\ \omega^I[\ I_D\ ], \omega^K[\ \mathcal{K}'_{\mathcal{VR}}(I_D)\ ], \omega^K[\ \mathcal{K}'_{\mathcal{VR}}(I_D)\ ]\ ) \tag{3.9}$$

therefore only utilizes I_D in combination with an image noise operator ω^I and a keypoint noise operator ω^K (see Fig. 3.4).

3.2.3.2 *Gating Network*

Additionally, we allow usage of the warped mouth area only through gated convolutions, which prevents direct information propagation. We feed the keypoint-warped representation of the mouth area into the mouth image encoder $\widetilde{E}$ (see generator in Figs. 3.3 and 3.4) that has a downsampling factor of four. This estimates the warped mouth area features.

$$\widetilde{E}_M = \widetilde{E}(\ \psi(\ I, \mathcal{K}'_{\mathcal{VR}}(I), \mathcal{K}'_{\mathcal{VR}}(I_D)\ )\ ), \tag{3.10}$$

where $\psi(\cdot)$ performs the keypoint warping from image I into the full driving VR keypoints $\mathcal{K}'_{\mathcal{VR}}(I_D)$.

For gating the aggregated deformed source features ${}^D E_S$, we concatenate the warped mouth area features $\widetilde{E}_M$ with ${}^D E_S$ and feed them through a small residual network with two layers to compute the gating weights (see Fig. 3.3). The resulting features in the main branch are therefore given by

$$f = \underbrace{\sigma(\ \phi[\widetilde{E}_M \oplus {}^D E_S]\ )}_{=:m_f} \odot {}^D E_S, \tag{3.11}$$

where $\odot$ is the elementwise multiplication, $\oplus$ is concatenation, $\sigma(\cdot)$ is the sigmoid function, and $\phi[\cdot]$ is the convolutional feature extraction of the small residual network.

Inducing visual mouth camera information implicitly through gating allows to mask out incorrect activations in the aggregated deformed source features ${}^D E_S$ (see Eq. (3.7)) while still being able to encode additional information without direct information propagation. This is especially beneficial when performing inter-operator animation (see Fig. 3.6) or generalizing from complete faces during training to mouth camera images during inference.

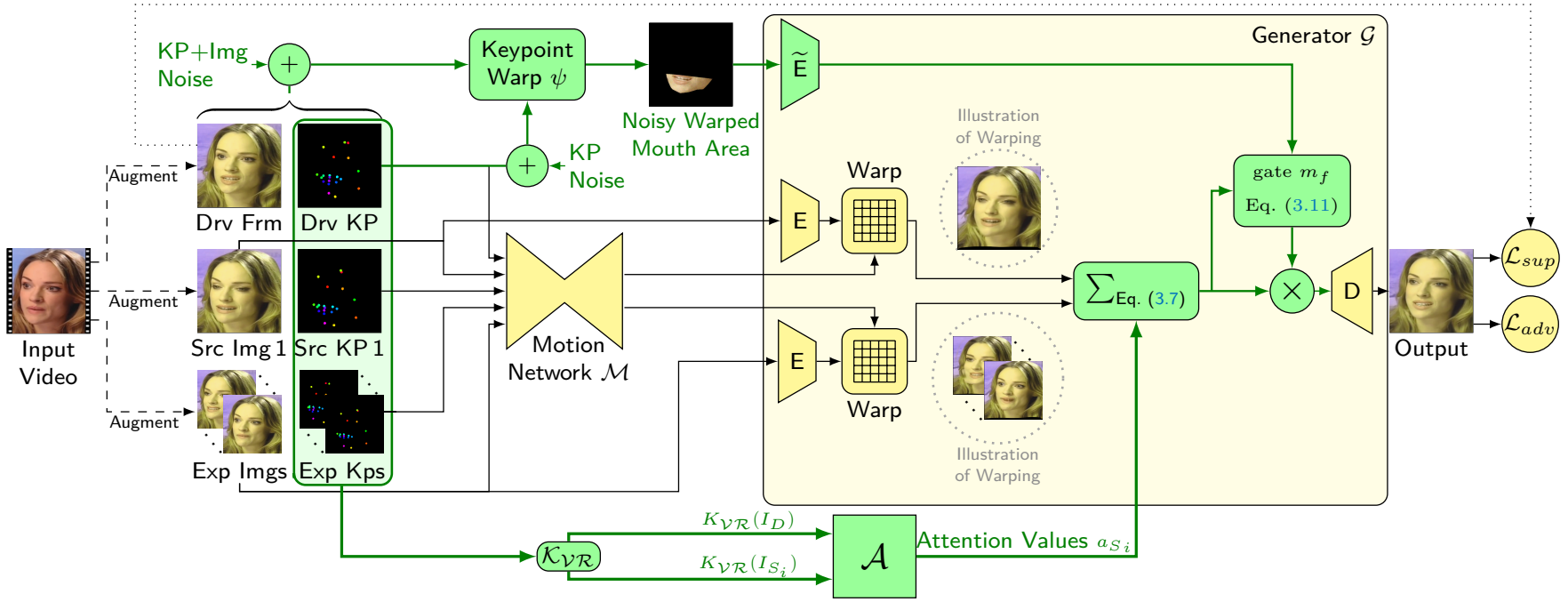

Figure 3.4: Training the extended VR facial animation network from videos. New components compared to our previous method from Chapter 2 are highlighted in green. The supervised training loss $\mathcal{L}_{sup}$ is minimized when the network reconstructs the driving frame given the source images, source keypoints, noisy warped mouth area, and the driving keypoints. Furthermore, a keypoint-aware discriminator network judges the quality of the generated image ($\mathcal{L}_{adv}$). The source images are chosen randomly from the input video. During training, we simulate mouth camera guidance by utilizing the lower facial region of the driving frame itself. We regularize the mouth camera guidance by injecting image noise and keypoint noise. This simulates different lighting and imperfect keypoint warping as present during inference when the real mouth camera image is utilized.

3.2.4 *Training*

The training pipeline is visualized in Fig. 3.4. All modules are trained end-to-end on the VoxCeleb dataset (Nagrani et al., 2017), prepared using the video preprocessing code from Siarohin et al. (2019b). We train with perceptual loss and utilize a keypoint-aware discriminator network to generate adversarial losses, similar to Siarohin et al. (2019b). Given a video, we randomly choose one driving frame and $n = 5$ different source images, from which the last four are expressional source images. We extract facial keypoints and estimate a deformation grid of all source images into the driving keypoints using the motion network $\mathcal{M}$. Simultaneously, we estimate the attention values a_{S_i}. The source image features are then deformed and aggregated in the lower facial region using the attention values (see Eq. (3.7)). The features are conditioned with the keypoint-warped mouth area (see Eq. (3.11)) and decoded to the output image.

We initialize the keypoint detector, motion network, generator-encoder, and generator-decoder with weights of our trained baseline (see Chapter 2). The new components (attention mechanism $\mathcal{A}$ and gating network) are trained from scratch. We found that the initialization with the baseline weights resulted in a very fast progress.

Finetuning with Imperfect VR Annotations: Unlike the baseline, our extended method allows to explicitly train with mouth camera images. We therefore extend the datasets with some VR facial animation samples. We annotate such samples by manually searching for lower-face expression correspondences in the mouth camera and the face camera. The manual alignment is a very challenging task and often there is no perfect solution. Due to time limitations and efficiency reasons, we only annotate 13 different operators of our system and chose a fraction of training samples from such imperfect annotations. To prevent overfitting, we randomly scale, rotate and crop the facial images. Random cropping followed by rescaling to a quadratic image also changes face aspect ratio.

During finetuning, we select 6% of the training samples from the annotated VR datasets and 94 % from the VoxCeleb dataset, which gives similar importance to our annotated videos and videos from VoxCeleb.

3.2.5 *Inference*

Preprocessing, which is briefly explained in Sec. 3.2.1, remains almost equivalent to our baseline (see Chapter 2) and takes approximately 15 minutes. However, instead of just one source image, we select n=4 or n=5 fixed source images with different facial expressions. Given the current mouth camera image we optionally retrieve the best matching image (see Sec. 2.2.5) which will be treated as an expressional source image. Following our baseline, we then construct the imaginary driving keypoints using the mapped mouth camera keypoints, the head pose keypoints from the first source image, and the eye tracking results (see Eq. (3.1)). The deformation, attention

and generator steps are equivalent to the training pipeline. During inference, the keypoint warping and gating step is always performed with the mouth camera image from the HMD (see Fig. 3.3). We therefore set $\widetilde{E}_M$ (see Eq. (3.10)) in Eq. (3.11) to

$$\widetilde{E}_M = \widetilde{E}(\ \psi(\ I_M, \mathcal{K}'_{\mathcal{VR}}(I_M), \Pi_S(\mathcal{K}'_{\mathcal{VR}}(I_M))\ )\ ), \tag{3.12}$$

where I_M is the mouth camera image and $\Pi_S(\mathcal{K}'_{\mathcal{VR}}(I_M))$ are the full mouth camera VR keypoints projected into the space of the first source image (i.e., the full VR keypoints of the imaginary driving frame).

3.2.6 *Temporal Consistency*

The baseline method presented in Chapter 2 often struggles generating temporally consistent facial animations. The abrupt change of the expression frame induces the greatest negative influence.

Our proposed attention mechanism can be used in two different configurations. In the first configuration, all n=5 source images are fixed, which minimizes the temporal inconsistencies as the continuous attention weight function changes smoothly with the mouth camera stream. The second configuration allows to retrieve the last expressional source image I_{S_5} during inference dynamically, which improves the output quality slightly (see Tab. 3.1). Here, the utilization of multiple source images, mitigates the negative impact of the dynamic source image on temporal consistency (see Tab. 3.2) compared to the baseline approach (Chapter 2), where temporal consistency is strongly influenced by retrieved expression frames. To further control the risk of temporal inconsistencies, we introduce a maximum attention value a_{max} for the retrieved images in the attention mechanism. This parameter allows us to control the tradeoff between quality and temporal consistency (see Tab. 3.2). During testing, we set a_{max} operator-specifically but with a default value of 25%. In case the image retrieval does not perform well, the a_{max} value can be reduced.

Furthermore, the proposed visual mouth camera guidance reduces the network's dependency on the retrieved source image which also contributes to the temporal consistency.

3.3 Experiments and Evaluation

We compare against our baseline method from Chapter 2, which we used at the ANA Avatar XPRIZE semifinals. A fair comparison to other methods (Lombardi et al., 2018; Richard et al., 2021a; Wei et al., 2019) is not feasible as they perform per operator optimization with a significant amount of training, preprocessing, and data capturing. All reported qualitative and quantitative results are obtained with unseen persons.

	MEAN			Male1			Male2			Male3			Fem1			Fem2		
Method	psnr	ssim	lpips	psnr	ssim	lpips	psnr	ssim	lpips	psnr	ssim	lpips	psnr	ssim	lpips	psnr	ssim	lpips
Ours-50%	**28.83**	**.8603**	**.0357**	**29.27**	**.8642**	**.0365**	**28.66**	**.8610**	.0368	**28.59**	**.8439**	.0368	**29.87**	**.9028**	**.0233**	**27.74**	**.8298**	**.0451**
Ours	28.75	.8586	.0361	29.08	.8603	.0373	28.63	.8602	.0370	28.45	.8410	.0375	29.87	.9023	.0235	27.72	.8294	.0452
Ours-5-Fix	28.50	.8504	.0376	28.87	.8494	.0401	28.30	.8521	.0383	28.06	.8274	.0399	29.66	.8974	.0238	27.59	.8257	.0461
Ours-Short	28.28	.8550	.0376	28.64	.8486	.0437	28.45	.8589	**.0318**	28.19	.8436	**.0364**	28.69	.8963	.0246	27.42	.8275	.0515
Ours-NF	27.20	.8369	.0465	27.77	.8350	.0432	27.36	.8363	.0467	27.13	.8267	.0437	27.50	.8842	.0357	26.23	.8024	.0630
Baseline	25.10	.7809	.0580	24.68	.7513	.0646	24.97	.7974	.0585	26.79	.7868	.0470	23.89	.8108	.0472	25.19	.7584	.0728
Ours-10-Skip	28.69	.8568	.0363	29.03	.8582	.0372	28.52	.8566	.0375	28.40	.8399	.0380	29.80	.9010	.0236	27.71	.8283	.0452
Baseline-10-Skip	24.96	.7758	.0596	24.61	.7469	.0653	24.91	.7902	.0589	26.69	.7858	.0481	23.78	.8074	.0500	24.83	.7486	.0756

Table 3.1: Ablation study. *NF*: No finetuning on mouth camera images, *Short*: finetuning for a short time which leads to only 4000 VR images in the training batches, *10-Skip*: only one out of ten source video images retrievable, *Ours*: maximum image retrieval attention parameter $a_{max} = 25\%$, *Ours-50%*: $a_{max} = 50\%$, *Ours-5-Fix*: only five fixed source images without image retrieval.

3.3.1 *Quantitative Results*

To generate quantitative results, we utilize the annotated VR dataset. The mouth camera image is the input and the corresponding facial image will be the driving frame. We evaluate our method on five different persons. As our method is intended to improve the facial animation in the lower facial region, we only measure the metrics Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) (Wang, 2004), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) in the lower half of the face without background.

3.3.1.1 *Accuracy*

Tab. 3.1 shows that all proposed model variants significantly outperform the baseline method from Chapter 2. All of our ablations, besides Ours-Short and Ours-NF, in Tab. 3.1 are trained for 50 epochs with annotated VR samples as explained in Sec. 3.2.4. The Ours-NF ablation, however, was never trained with VR samples. In this case, the missing regularizing influence results in overfitting after roughly five epochs, so we report the results at this training step. Interestingly, Ours-NF already outperforms the baseline in all metrics significantly. Ours-Short is only finetuned for 5 epochs which corresponds to just 4000 VR/Face image pairs that have been seen. This is already enough to generate similar results as obtained with 50 epochs VR finetuning.

Ours-10-Skip and Baseline-10-Skip represent ablations where only one out of ten images in the source video is retrievable. This results in a larger gap between the driving frame and retrievable source images. When reducing retrievable images and thus the number of presented facial expressions by factor ten, quality is only influenced slightly (see mean metrics of Ours vs. Ours-10-Skip in Tab. 3.1).

Our second method variation (Ours-5-Fix) further limits the number of different facial expressions presented to the network. It has only five fixed source images and therefore uses no image retrieval. The results indicate that image retrieval is not essential in our method for achieving good animations. Comparing all our method ablations shows that $a_{max} = 50\%$ generates the highest accuracy, but results in reduced temporal consistency, compared to $a_{max} = 25\%$ and Ours-5-Fix without image retrieval, as evaluated in Tab. 3.2.

3.3.1.2 *Temporal Consistency*

In our proposed method and in the baseline from Chapter 2, temporal inconsistencies mainly occur whenever a new expression frame is selected, which happens in roughly every second frame when speaking. Measuring temporal consistency in animated facial images is a non-trivial task, especially when disoclusions and complex facial deformations occur. To reduce these effects, we use the motion network to deform the previous prediction into the current one. This allows comparison using perceptual similarity (LPIPS (Zhang et al., 2018)), with the assumption that two consecutive

Method	Male2	Female1	Male3	Female2
Ours-5-Fix	**+0.0 %**	**+0.0 %**	**+0.0 %**	**+0.0 %**
Ours-25%	+6.2 %	+5.9 %	+11.5 %	+8.7 %
Ours-50%	+7.6 %	+8.3 %	+15.8 %	+16.8 %
Baseline	+50.8 %	+86.2 %	+106.5 %	+151.9 %
Ours-50%+TCF	(+1.3 %)	(+1.3 %)	(+4.5 %)	(+4.4 %)
Baseline+TCF	(+21.6 %)	(+33.5 %)	(+45.0 %)	(+88.1 %)

Table 3.2: Measure of temporal inconsistency in the generated animations. Values normalized to Ours-5-Fix. Lower is better. 25% and 50% indicate the a_{max} parameter, TCF means temporal consistency filtering, which is applied to the dynamic expression frame (see Sec. 2.2.8). Persons sorted by image retrieval quality (left: good).

frames exhibit only small expressional differences. Importantly, unintended discontinuous flicker effects lead to large errors in this metric. Note that the proposed measure does not necessarily correlate with accuracy.

In Tab. 3.2, we report temporal inconsistency for four different persons from Tab. 3.1, which are ordered with a descending image-retrieval quality from left to right. The best temporal consistency is obtained without image retrieval (Ours-5-Fix). When using image retrieval, the measured temporal consistency decreases with the maximum attention parameter a_{max} (see Sec. 3.2.6). Together with Tab. 3.1, this highlights the temporal consistency vs. accuracy tradeoff, which is controllable through a_{max}. However, compared to the baseline, all of our model variants perform much better, which is due to the baseline's strong dependence on the retrieved image (i.e., the expression frame). The discrepancy to our method gets larger with worsening image retrieval quality.

To increase temporal consistency, the baseline method (Baseline+TCF) explicitly minimizes this measuring scheme by recursively low-pass filtering the retrieved expression frame using the deformations of the last expression frame and the last prediction (see Sec. 2.2.8), which is exactly what we measure. However, even if the image retrieval works fine, this comes with the cost of a reduced image quality in the lower facial region. Even though our ablations already achieve significantly better results than Baseline+TCF, we equip an additional ablation using $a_{max} = 50\%$ with the same recursive filtering scheme (Ours-50%+TCF) to allow a fairer comparison.

3.3.2 *Qualitative Results*

Qualitative results are shown in Figs. 3.5 to 3.7. Fig. 3.5 compares our method with ground truth and the baseline presented in Chapter 2. It shows exemplary results of our quantitative evaluation in Tab. 3.1. As can be seen, our results are much more accurate and closer to the ground truth. Unlike our method, the baseline fails whenever a bad expression frame is retrieved.

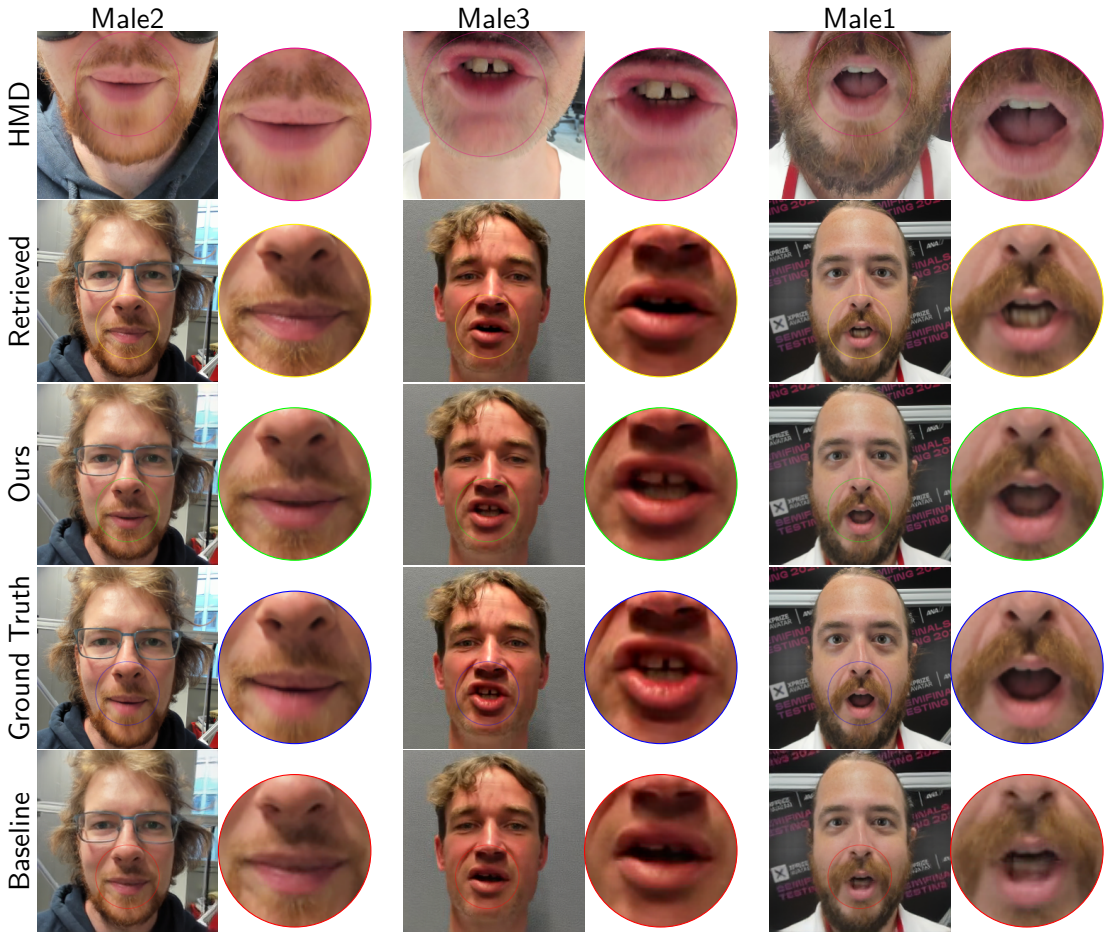

Figure 3.5: Visual results of our quantitative analysis in Tab. 3.1. For all examples the image retrieval (second row) was inaccurate, which led to poor results for the baseline (bottom) presented in Chapter 2. Our method still generates good results.

Fig. 3.6 demonstrates very challenging mouth expressions obtained when mapping from the mouth camera input to a different person. This experiment shows that, unlike our baseline, the proposed mouth camera guidance allows to resolve keypoint ambiguities and properly displays very challenging facial expressions, such as lips which partly stick together.

Fig. 3.7 contains inference results compared with the baseline. In particular, we want to highlight that Ours-5-Fix produces almost the same visual results as our method configuration with image retrieval (Ours).

The supplementary video[4] contains an animated comparison.

4 https://www.ais.uni-bonn.de/videos/IROS_2023_Rochow

Figure 3.6: VR facial animation from mouth camera input to the appearance of a different operator. Mouth camera guidance resolves keypoint ambiguities and models a broader range of mouth expressions (note the lips which partly stick together). All images are cropped for visualization.

3.3.3 *Throughput and Latency*

We use pipelining techniques to enhance the throughput from 29 fps to 34 fps on an NVIDIA A6000 GPU with very low latency (34 ms excluding and 51 ms including camera exposure time).

3.3.4 *The ANA Avatar XPRIZE Finals*

At the ANA Avatar XPRIZE competition finals in November 2022, our team and three different operators had to accomplish three test runs, of which the first one was a qualification run. The goal was to complete ten different tasks as fast as possible. For each completed task one point was awarded. Five additional points were awarded for usability and the ability to understand emotions and gestures. Especially for these, a facial animation was mandatory. Two tasks consisted of interacting with a human recipient. Our facial animation pipeline allowed seamless and immersive interaction between operator and recipient, which was rewarded with a full judge score on all three days. Overall, our Team NimbRo achieved a perfect score (15/15) with the fastest time in all three runs.

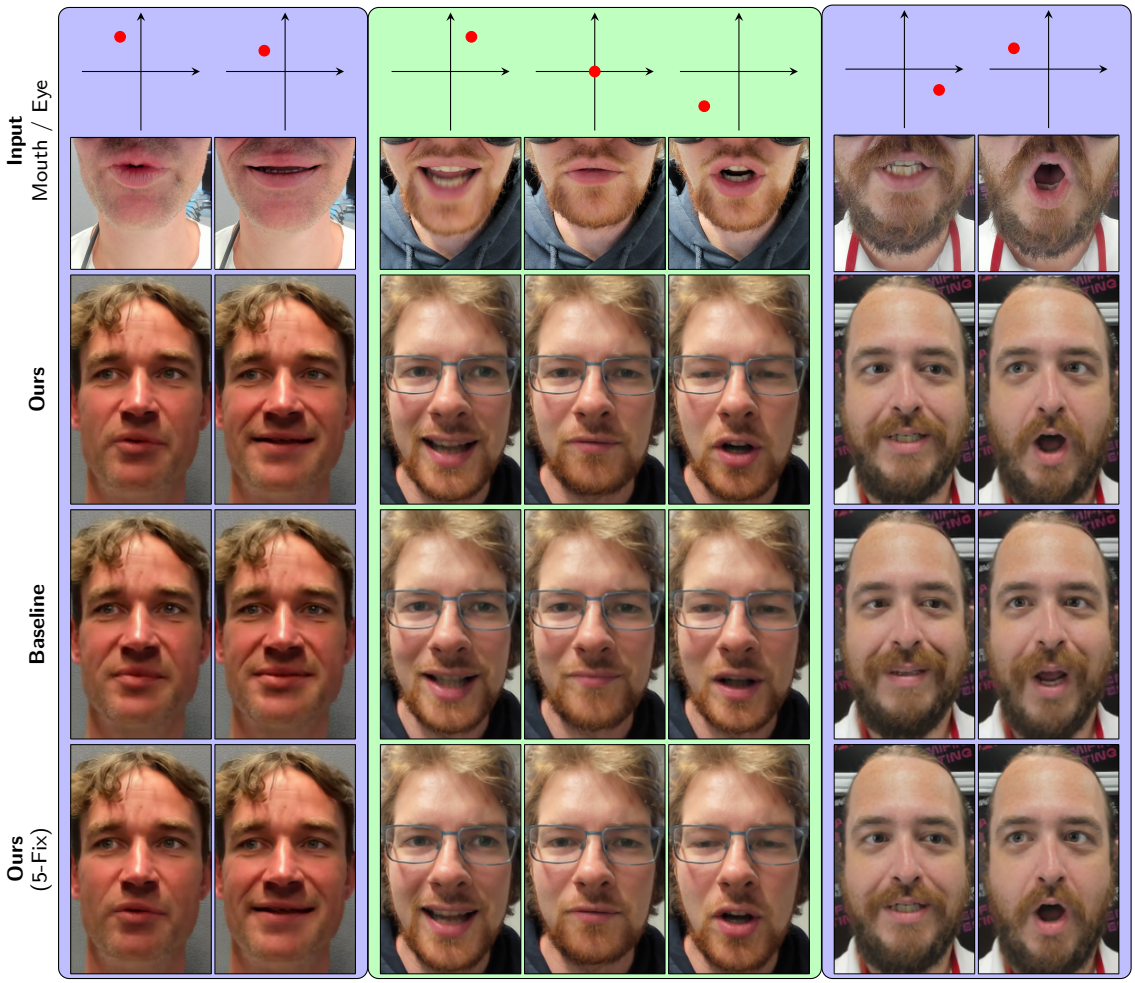

Figure 3.7: Generated faces during inference, given the mouth camera image and eye coordinates. All images are cropped for visualization. See also the supplementary video (see Footnote 4) for an animated comparison.

3.4 Conclusion

We proposed a real-time capable VR facial animation approach that generalizes well to unseen operators and allows for modeling a broader range of facial expressions, compared to solely keypoint-driven approaches. We extended the baseline from Chapter 2 with a source image attention mechanism and developed a way to inject visual mouth camera image information into the animation pipeline without overfitting. These two extensions yield better accuracy and significantly improve temporal consistency which is important for smooth interaction. Our method still struggles in generating unusual expressions such as sticking out the tongue. Furthermore, movement in the upper face is still limited.

4

FSRT: Facial Scene Representation Transformer for Face Reenactment

Preface

The main part of this chapter is adapted from the following publication (in conjunction with the corresponding Supplementary Material):
Andre Rochow, Max Schwarz, and Sven Behnke (2024). "FSRT: Facial Scene Representation Transformer for Face Reenactment from Factorized Appearance, Head-pose, and Facial Expression Features." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7716–7726. DOI: 10.1109/CVPR52733.2024.00737, , presented in Seattle, WA, USA.

Statement of Personal Contribution

The author of this thesis is the main author of the publication Rochow et al. (2024) and significantly contributed to all aspects of the publication and this chapter. These contributions particularly include proposing, conceptualizing, designing, and implementing the method, conducting a comprehensive literature review, evaluating the method, as well as analyzing and interpreting the results. Specifically, the extensive evaluation of the method included, among other elements, the design, execution, and analysis of a user study. Additionally, the author of this thesis was responsible for training the model ablations and made the primary contribution to the writing of the paper, including the generation of visualizations.

Unpublished Content

In addition to the content of the publication Rochow et al. (2024), this chapter contains unpublished material in Sec. 4.5, which describes an experiment on leveraging the FSRT method for VR facial animation. VR facial animation is examined in Chapters 2 and 3 of this thesis. The unpublished material is contributed by the author of this thesis.

Face reenactment (Siarohin et al., 2019a,b) is a special case of the motion transfer task. Its objective is to synthesize a realistic facial animation combining the appearance given by one or more images of a source person and the facial expression and head motion of a driving video, which may be of a different person. The driving frame is used to transform the source image to the desired expression and head pose. When the driving video is of the same person (self-reenactment), applications include, e.g., low-bandwidth video conferencing (Wang et al., 2021b). The more interesting and challenging case is when the driving video is of a different person (cross-reenactment) since, if successful, only one or few images of the source person are required to create a realistic facial animation.

Most face reenactment methods are CNN-based (Ha et al., 2020; Hong et al., 2022; Hsu et al., 2022; Siarohin et al., 2019a,b; Wang et al., 2019, 2021b; Wiles et al., 2018; Zakharov et al., 2019; Zhao and Zhang, 2022; Zhao et al., 2021a); and many of these utilize optical flow between the source and driving frames for morphing the source image, followed by a refinement stage (Hong et al., 2022; Siarohin et al., 2019a,b; Wang et al., 2021b; Zhao and Zhang, 2022).

Inspired by recent successes in scene reconstruction (Sajjadi et al., 2022a; Sajjadi et al., 2022b), we apply a transformer-based architecture to face reenactment that encodes the face of the source person as a set of latent vectors (see Fig. 4.1). This representation is learned in a self-supervised way. We then sample each target pixel location with a transformer-based decoder conditioned on keypoints and an expression vector that are extracted from the driving frame. The learned set-latent representation of the source person factorizes their head pose, appearance, and facial expression, which enables accurate head motion and facial expression transfer, also for cross-reenactment.

While our method yields state-of-the-art results in absolute motion transfer (i.e., when the driving keypoints are used unmodified), it especially increases robustness in the case of relative motion transfer (Siarohin et al., 2019b), a mode which reduces unwanted leaking of face shape from the driving frame. Relative motion transfer is initialized by finding a driving frame with a well-matched head pose and expression to the source image, and then operates by applying motions to the source image in a relative fashion.

Many methods encode facial expression by keypoints, which are augmented with local spatial transformations (Hong et al., 2022; Siarohin et al., 2019b; Zhao and Zhang, 2022). Here, relative motion is encoded as relative transforms from the initial driving frame and is applied in the source image. Of course, this is highly sensitive to the initialization. By decoupling head pose from expression and describing expression in an absolute manner, our approach reduces the influence of initialization on expression transfer.

Finally, we note that our approach makes few assumptions since there is no explicit model of motion or correspondence. Instead, the set-latent representation of the source person is learned in a self-supervised way using conditioning with keypoints

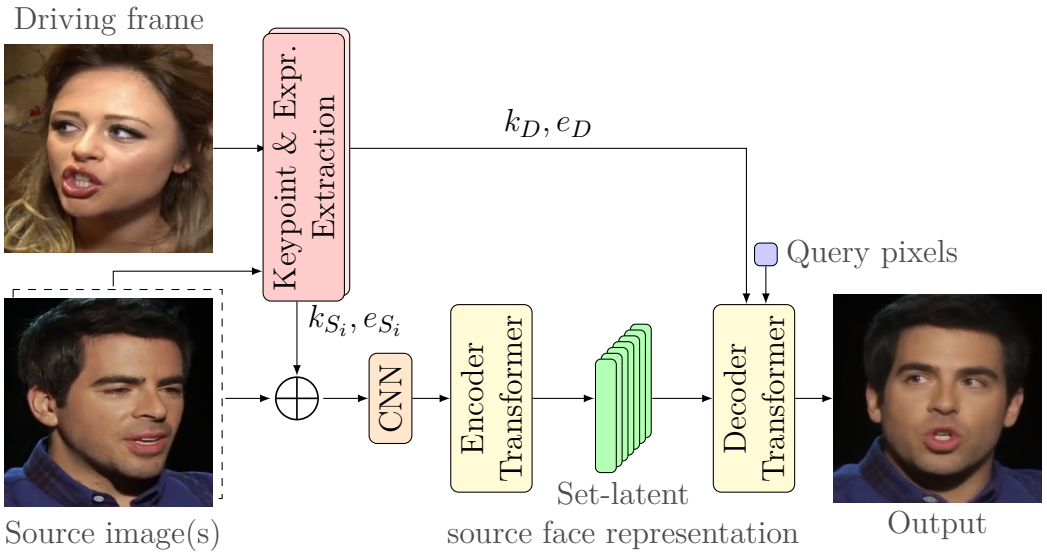

Figure 4.1: Overview of our method (relative motion transfer). The source image(s) are encoded along with keypoints k_S, capturing head pose, and facial expression vectors e_S to a set-latent representation of the source person. The decoder attends this representation for a query pixel, conditioned on keypoints k_D and a facial expression vector e_D extracted from the driving frame. $\oplus$ denotes pixel-wise concatenation. Images from the VoxCeleb test set (Nagrani et al., 2017).

and a facial expression vector from each source image on the encoder and from the driving frame on the decoder, respectively.
In summary, our contributions include:

1. A novel transformer-based architecture for face reenactment which learns a global set-latent representation of source images and allows rendering conditioned on keypoints and expression vectors,

2. a latent expression description invariant to appearance, head shape, and head pose, greatly improving cross-reenactment,

3. augmentation and regularization methods for training that support the separation of facial expression information (expression vector), appearance (set-latent representation), and head pose (keypoints) without overfitting,

4. application of adversarial and perceptual losses to scene representation transformers, which greatly improves realism and sharpness,

5. a detailed evaluation, where we outperform previous state-of-the-art methods in motion transfer quality and temporal consistency, including a user study which also shows superiority in subjective preference trials.

The source code is publicly available[1] and video results can be found on the project page[2].

1 https://github.com/andrerochow/fsrt
2 https://andrerochow.github.io/fsrt

4.1 Related Work

Face Image Synthesis. Face image synthesis deals with the generation of new non-existing faces (Hou et al., 2019) even from text input (Huang et al., 2023; Lin et al., 2023), completion of missing facial regions of known faces (Li et al., 2017; Zhou et al., 2020), or manipulation of expression and appearance of known faces in manual (Huang et al., 2023; Jo and Park, 2019; Lee et al., 2020) or automatic manner (Choi et al., 2018; Deng et al., 2020; Pumarola et al., 2018; Shen et al., 2020).

In contrast to these methods, our approach combines the continuously changing head pose and expression from the driving video with the appearance of the source, so that a natural and temporally consistent video stream is produced.

Talking Head Synthesis and Face Reenactment. Talking Head Synthesis aims to make head poses, emotions, and especially speech controllable. Here, lip movement is mainly reconstructed from audio (Ma et al., 2023; Richard et al., 2021b; Wang et al., 2023a; Wang et al., 2023b; Wang et al., 2023c; Wang et al., 2021a; Wiles et al., 2018; Xu et al., 2023; Zhou et al., 2019, 2021). Closely related, face reenactment (Thies et al., 2016), which is a motion transfer task, aims to apply the motion given by a driving frame to the appearance defined by a source image. An even more challenging problem is VR facial animation, where the driving face is additionally occluded by an HMD (Lombardi et al., 2018; Richard et al., 2021a; Thies et al., 2018; Wei et al., 2019, Chapters 2 and 3, Sec. 4.5). Here, especially the alignment problem between facial images and mouth camera images makes it difficult to generate training data.

Generally, there are different types of driving information being used. Where some methods only utilize facial keypoints (Hsu et al., 2022; Siarohin et al., 2019a; Wang et al., 2019; Zakharov et al., 2019; Zhao et al., 2021a), other methods additionally use image features from a driving frame (Ha et al., 2020; Hong et al., 2022; Siarohin et al., 2019b, 2021; Wang et al., 2021b; Zhao and Zhang, 2022). Also, audio can be used if available (Agarwal et al., 2023).

Some methods (Hsu et al., 2022; Zhao et al., 2021a) utilize external 3D keypoints extraction (Bulat and Tzimiropoulos, 2017) for face reenactment. Hsu et al. (2022) use a separate generator to predict more accurate driving keypoints, given initial keypoints and a source image. Siarohin et al. (2019a) learn keypoints self-supervised and use them to predict a deformation grid of the source image into the driving keypoints. To resolve keypoint ambiguities, Siarohin et al. (2019b) estimate local affine transformations into a canonical space for each keypoint area. This allows far more facial expressions to be represented. Based on (Siarohin et al., 2019b), Hong et al. (2022) learn depth maps, which they use to predict more accurate keypoints and dense depth-aware attention maps, which can attend to important semantic facial areas. Zhao and Zhang (2022) use a motion estimation based on thin-plate spline transformations to produce a more flexible optical flow. They use

multi-resolution depth maps and occlusion masks to inpaint missing regions more realistically.

However, driving motion does not necessarily has to be described by keypoints (Li et al., 2023; Pang et al., 2023; Siarohin et al., 2021; Thies et al., 2016; Wiles et al., 2018). Wiles et al. (2018) directly predict the sampling coordinates to a canonical embedding of a face. A separate driving network then predicts the mapping from the embedded face to the driving frame. Siarohin et al. (2021) bypass the keypoint estimation step from predicted heatmaps and consider them as regions. They estimate the principal components of the region to predict an in-plane rotation and scaling, which is used to estimate a more accurate pixel-wise optical flow. Pang et al. (2023) learn a disentanglement of pose and expression, so that different driving sources can be used. In another approach, Li et al. (2023) learn to embed a source image into a canonical volume and predict the deformation of individual sampled rays to estimate the optical flow.

Wang et al. (2022) and Gong et al. (2023) replace the motion network proposed by Siarohin et al. (2019a) with custom modules (Linear Motion Decomposition and a transformer module enabling domain switching, respectively), but remain fundamentally based on CNNs (in encoder and decoder) and a warp-and-refine architecture.

Unlike related methods, we use a transformer-style architecture to predict a latent scene representation of the source images and learn expression vectors which are decoupled from appearance and head pose information. Instead of modeling optical flow and motion explicitly, we learn to attend the latent scene representation, with keypoints and latent expression vectors extracted from a driving frame. This allows us to generate results of higher accuracy, while significantly improving the temporal consistency.

Scene Representation Transformers. While transformers (Vaswani et al., 2017) were originally developed for natural language processing, vision transformers have also been highly successful (Dosovitskiy et al., 2021; Liu et al., 2021). In terms of novel view synthesis, Sajjadi et al. (2022b) proposed Scene Representation Transformers (SRT) to learn an internal scene representation encoded in a set of latents vectors. Given these latent vectors and a camera pose, SRT allow novel-view rendering without explicitly modeling the scene geometry. Based on this, Sajjadi et al. (2022a) propose a slot attention module to instead predict an object-centric slot scene representation, in which different objects are separated without any supervision.

In this work, we reformulate SRT (Sajjadi et al., 2022b) for the face reenactment task and demonstrate how to efficiently model and query dynamics in the learned face representation. Unlike Sajjadi et al. (2022b), we aim to reconstruct photorealistic faces. To this end, we propose training with perceptual (Johnson et al., 2016) and adversarial losses (Goodfellow et al., 2014), which significantly improves the output quality.

4.2 Method

The SRT architecture (Sajjadi et al., 2022b) encodes one or more posed images to an internal representation and reconstructs views from arbitrary viewpoints. We adapt the architecture and the input representation in such a way that we learn an internal representation from one or more facial images. Reconstruction then allows free choice of head pose and facial expression. Given a set-latent representation of an encoded face, head pose and facial expression can be controlled by ten keypoints and a latent expression vector. Abstractly, the internal representation can also be understood as an embedding that separates the appearance of a person from the head pose and expression.

Given an input representation $\{R_{S_i}\}$ (see Sec. 4.2.1), the transformer encoder $\mathcal{E}$ (Fig. 4.2 and Sec. 4.2.1) produces a set-latent scene representation

$$\{z_z \in \mathbb{R}^d\} = \mathcal{E}(\text{CNN}(\{R_{S_i}\})), \tag{4.1}$$

where CNN (Sec. 4.2.1) is a convolutional feature extractor backbone (shared in case of multiple input images), as proposed by Sajjadi et al. (2022b). Given this set-latent representation and the query $Q_{I_D}(q)$ (see Sec. 4.2.1), the transformer decoder $\mathcal{D}$ predicts the pixel color

$$C(q) = \mathcal{D}(Q_{I_D}(q) \mid \{z_z\}). \tag{4.2}$$

Our full architecture is visualized in Fig. 4.2.

4.2.1 *Input and Query Representation*

Given are one more facial source images I_{S_i}. We encode each image through ten keypoints k_{S_i}, computed by a keypoint detector, and a latent expression vector e_{S_i} which we learn in self-supervised manner. The keypoints are normalized to $(-1, 1)$ and positionally encoded (Mildenhall et al., 2020)

$$f(p, s_O, O) = \bigoplus_{m=s_O}^{s_O+O-1} \sin(2^m \pi p) \oplus \cos(2^m \pi p) \tag{4.3}$$

to obtain

$$\gamma_{\text{key}}(k_{S_i}) = \bigoplus_{j=0}^{n_\mathcal{K}} f\left(k_{S_i}^{(j)}, s_{O_{\text{key}}}, O_{\text{key}}\right) \tag{4.4}$$

where $n_\mathcal{K}$ is the number of keypoints, O_{key} is the number of octaves per keypoint, $s_{O_{\text{key}}}$ is the keypoint start octave, and $\bigoplus$ is the vector concatenation.

During face reenactment, keypoints might move out of the image boundaries $(-1, 1)$. Due to this, we set $s_{O_{\text{key}}} = -1$ to add an additional negative octave, which

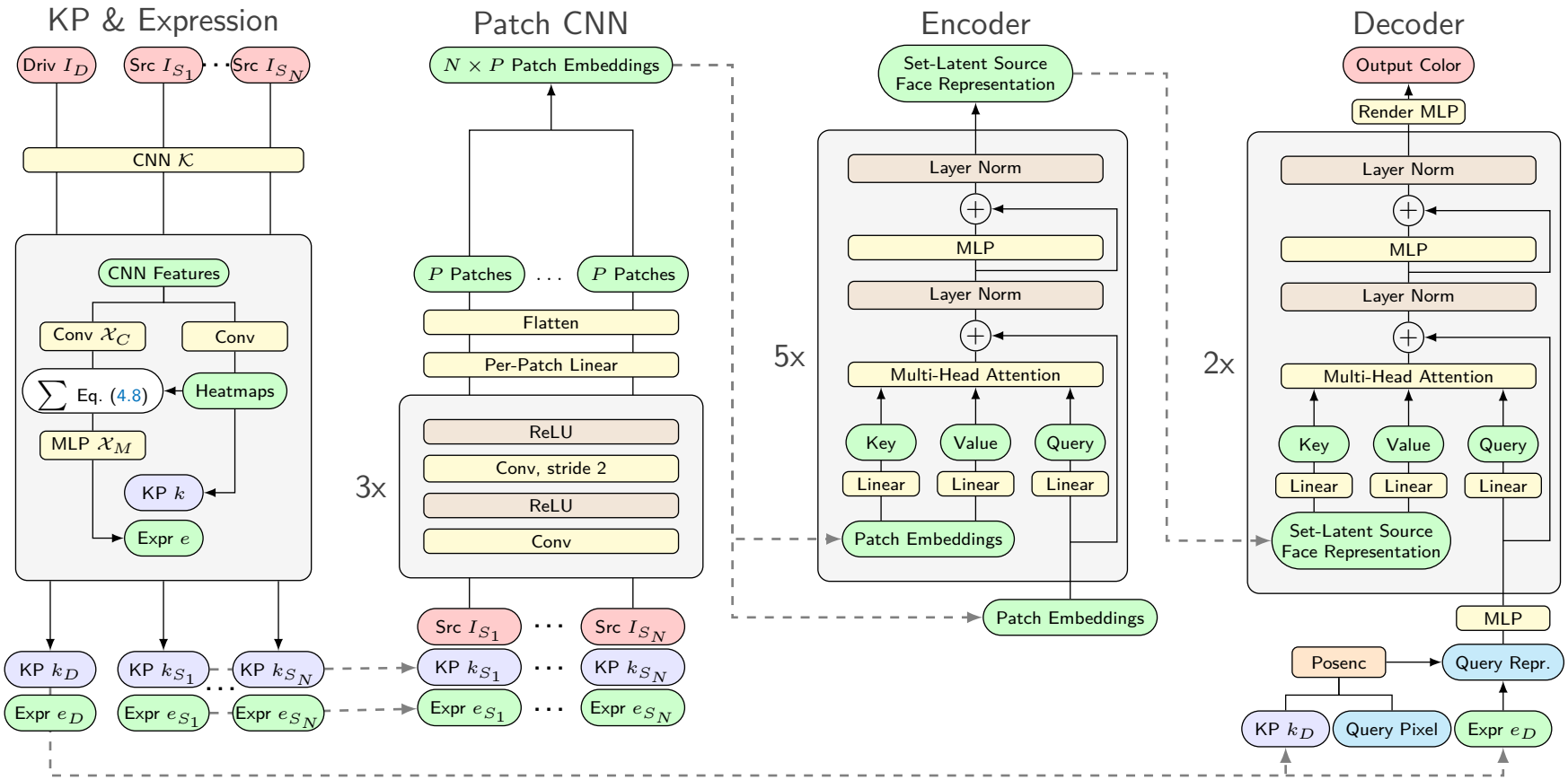

Figure 4.2: Architecture details. Given the driving frame and source images, we extract facial keypoints and latent expression vectors. Extracted source information are used to generate the input representation of the Patch CNN. The encoder infers the set-latent source face representation from the patch embeddings as in SRT (Sajjadi et al., 2022b). The decoder is applied for each query pixel individually and is conditioned with the driving keypoints and the latent driving expression vector. For further implementation details we refer to Sec. 4.4.1 in the Supplementary Material.

extends the interval of uniquely encodable values to $(-2, 2)$. We generate training samples with keypoints outside the image by estimating them before cropping, which is explained in more detail in Sec. 4.4.1.2.

Similar to the keypoints, each image pixel with normalized coordinate $p = (x, y)$ receives a 2D positional encoding (Mildenhall et al., 2020)

$$\gamma_{\text{pix}}(p, O_{\text{pix}}, s_{O_{\text{pix}}}) = f\left(p, s_{O_{\text{pix}}}, O_{\text{pix}}\right). \tag{4.5}$$

The input representation of the source images I_{S_i} at pixel $p = (x, y)$ is then given by

$$R_{S_i}(p) = [c_p, \gamma_{\text{pix}}(p), \gamma_{\text{key}}(k_{S_i}), e_{S_i}], \tag{4.6}$$

where c_p is the RGB color at pixel p and e_{S_i} is the latent expression vector extracted from I_{S_i} (see Sec. 4.2.1). Note that each pixel is conditioned with the full keypoint encoding and the full latent expression vector, which quickly leads to a large input representation.

The decoder is queried for every output pixel $q = (x', y')$. Instead of the camera pose, as in (Sajjadi et al., 2022b), each positionally-encoded query pixel is additionally conditioned with the desired target keypoints k_D (i.e., the driving keypoints) and the latent expression vector e_D of the desired expression (i.e., the latent expression vector of the driving frame).

Hence, the pixel-wise query representation is

$$Q_{I_D}(q) = [\gamma_{\text{pix}}(q), \gamma_{\text{key}}(k_D), e_D]. \tag{4.7}$$

Intuitively, the decoder attends to the most important features of the representation $\{z_z\}$ to render the source image appearance with the desired motion given by k_D and e_D.

Note that our method does not require camera extrinsics or intrinsics, since we operate directly in pixel space.

Keypoint Detector. The keypoint detector $\mathcal{K}$ is a fully convolutional hourglass network (Newell et al., 2016) as proposed by Siarohin et al. (2019a,b). Like $\mathcal{K}_{\mathcal{F}}$ in Chapters 2 and 3 (see Sec. 2.2.2), the keypoint detector is used as-is from a pretrained First Order Motion Model (Siarohin et al., 2019b) and not trained further. Note that it has also has the same weights as $\mathcal{K}_{\mathcal{F}}$. However, instead of removing the single keypoint that captures the position of the lower lip (as we do in Chapters 2 and 3), we keep all 10 predicted keypoints. For each input image I, it predicts heatmaps $H_I^{(j)} \in [0, 1]^{H \times W}$, $j = 1, \ldots, n_{\mathcal{K}}$, which define the pixelwise presence confidence of keypoint $k_I^{(j)}$. For all experiments, we fix the number of keypoints to $n_{\mathcal{K}} = 10$.

Expression Network. We assume that the last feature maps $F_{I,\mathcal{K}}$ predicted by the keypoint detector capture local image properties. Given this assumption we build an expression network $\mathcal{X}$ that recycles $F_{I,\mathcal{K}}$ and the predicted keypoint heatmaps

H_I for an image I (see Fig. 4.2). We filter $F_{I,\mathcal{K}}$ by a learned 7×7 convolution $\mathcal{X}_C$, producing $F_{I,\mathcal{X}}$ with shape $[n_f, n_\mathcal{K}, h', w']$. To focus on facial features, we utilize H_I to aggregate the features spatially for each keypoint $k_I^{(j)}$:

$$\vec{f}_{I,\mathcal{X}}^{(j)} = \bigoplus_{c=1}^{64} \left[\sum_{y=0}^{h'} \sum_{x=0}^{w'} H_I^{(j)}(x,y) F_{I,\mathcal{X}}^{(c)}(j,x,y) \right] \in \mathbb{R}^{n_f} \tag{4.8}$$

to obtain

$$f_{I,\mathcal{X}} = \bigoplus_{j=1}^{n_\mathcal{K}} \left[\vec{f}_{I,\mathcal{X}}^{(j)} \right] \in \mathbb{R}^{n_\mathcal{K} \cdot n_f}, \tag{4.9}$$

where c is the channel index, j is the keypoint index, and $\bigoplus$ is the concatenation operation. Using a 4-layer multi-layer perceptron (MLP) $\mathcal{X}_M$, we compute the latent expression vector

$$e_I = \mathcal{X}_M \left(f_{I,\mathcal{X}} \right). \tag{4.10}$$

The expressional information of all important keypoint areas are thus spread throughout e_I. Additionally, keypoint regions that do not contain important expression information can be filtered out by combining the local information.

Patch CNN. As in (Sajjadi et al., 2022b), the shared CNN is designed to reduce the spatial dimension of the input data and fuse patch information. In each block, the height and width are reduced by factor two and the number of feature maps is doubled. For all experiments, we use three CNN blocks followed by a final convolution with kernel size one, which generates the number of feature maps $n_\mathcal{E}^{fm}$ needed for the transformer encoder. The features with shape $[bs, n_\mathcal{E}^{fm}, H/8, W/8]$ are then reshaped to $[bs, \frac{H+W}{8}, n_\mathcal{E}^{fm}]$ which is the patch embedding input to the encoder.

Encoder. Following Sajjadi et al. (2022b), the standard transformer alternates global self-attention (between all tokens) and small MLP networks (see Fig. 4.2). Following Sajjadi et al. (2022a), we drop source ID embeddings and reduce the number of attention blocks to five. Through self-attention across all patch embeddings, the encoder learns a set-latent scene representation $\{z_z\}$ in which each vector z_z captures global scene and dynamics information. Note that the cardinality of the set-latent representation scales linearly with the number of source images.

Decoder. The transformer decoder does not use self-attention, but instead attends the set-latent scene representation with the query Q_{I_D}. This is repeated for two times, followed by a render MLP that predicts the final output color at a certain pixel location. For better performance, the query is first fed through a small 2-layer MLP that spreads the information in all dimensions, as proposed by Sajjadi et al. (2022a). Furthermore, we also use a final 5-layer render MLP that predicts the output color given the output of the attention module. Intuitively, the decoder learns to

attend to the most important features of $\{z_z\}$ to infer the pixel information of the encoded facial image(s) with the requested head pose and facial expression. Note that unlike SRT (Sajjadi et al., 2022b), we do not only request a novel view of a static scene but also certain dynamics within the scene, such as mouth movement. It is therefore necessary for the encoder to output a highly flexible scene representation (see experiments with small decoder in Sec. 4.3.2).

Due to the transformer design, the decoder can handle $\{z_z\}$ of any cardinality. Thus, a trained encoder and decoder can operate on a flexible number of source images.

For a given source face, we only need to predict the set-latent scene representation once and each query pixel is estimated independently of the others. This is an advantage over CNN-based approaches (Hong et al., 2022; Siarohin et al., 2019a,b; Wang et al., 2021b; Zhao and Zhang, 2022), because it allows the model throughput to be scaled linearly with the number of available GPUs. Only one copy of the decoder needs to reside on each GPU.

4.2.2 *Augmentation and Regularization*

Ideally, the network should learn to decouple appearance, pose, and expression information into set-latents z_z, keypoints k_I, and expression vector e_I, respectively. This separation of concerns enables cross-reenactment. In practice, the method is prone to overfitting, since we can only train in the self-reenactment regime, where ground truth is available. This results in latents that jointly encode appearance, pose and expression, which is visible when cross-reenacting to a different person. Artifacts appear in the background area around the face and the model also deforms the source person to be closer to the face shape of the driving frame (see Fig. 4.3). Hence, we do not reach the intended separation level. To combat this, we implement several data augmentation and regularization measures.

Color Augmentation. To prevent colors leaking from the driving frame to the output image, we apply color jitter augmentation on the source images. Specifically, we create two color-jittered versions I_S^{A1}, I_S^{A2} of the input image I_S. The expression network $\mathcal{X}$ is run to extract expression vectors $e_{I_S^{A1}}, e_{I_S^{A2}}$. An additional regularization term is added to enforce invariance to color jitter:

$$\mathcal{L}_{\text{aug}} = \frac{1}{|e|} \left\| e_{I_S^{A1}} - e_{I_S^{A2}} \right\|_2^2 . \tag{4.11}$$

While the encoder $\mathcal{E}$ is always trained with RGB colors from I_S^{A1}, it receives the expression vector $e_{I_S^{A2}}$. This further improves color invariance.

Cropping. To reduce background information in the output of $\mathcal{X}$, we further randomly crop the driving frame (just for $\mathcal{X}$). Here, we define $\Omega(\cdot)$, which selects a random crop with awareness of facial keypoints as proposed by Bulat and Tz-

imiropoulos (2017). This crop is then scaled back to the original size, which can change the aspect ratio. We add a loss term

$$\mathcal{L}_{\text{aug,D}} = \frac{1}{|e|} \left\| e_{I_D^A} - e_{\Omega(I_D^{A3})} \right\|_2^2 \tag{4.12}$$

on the expression vectors of color-jittered versions I_D^A, I_D^{A3}, in which A is either A_1 or A_2. Adding $\Omega(\cdot)$ to the loss term encourages that $\mathcal{X}$ extracts scale-invariant expression information only from the face region. Primarily, the expression vector of $\Omega\left(I_D^{A3}\right)$ is also utilized for decoding. However, in 25% of cases, $e_{I_D^A}$ is selected, which employs the same color-jittering (A_1 or A_2) applied to the source images.

Statistical Regularization. Data augmentation alone is not enough to completely prevent head pose, expression, and appearance information from being jointly encoded (see Fig. 4.3). We take inspiration from VICReg (Bardes et al., 2022), a method for regularization of unsupervised feature learning based on invariance, variance, and covariance, but adapt it to encourage the focus on expression information. Invariance against augmentations is already covered by Eqs. (4.11) and (4.12).

The covariance part aims to decorrelate along the feature dimension. Intuitively, decorrelation encourages separation of expression from head pose, shape, and appearance and enables the network to drop non-expressional information (which is already encoded in keypoints and scene representation $\{z_z\}$). Given a batch of source images and driving frames concatenated in the batch dimension

$$\begin{aligned} E = [&e_{S_1}^{(1)}, \ldots, e_{S_{(n_{src})}}^{(1)}, e_D^{(1)}, \\ &\vdots \\ &e_{S_1}^{(bs)}, \ldots, e_{S_{(n_{src})}}^{(bs)}, e_D^{(bs)}] \end{aligned} \tag{4.13}$$

with shape $[(n_{src}+1)bs, c]$, we estimate the covariance of the individual dimensions $\text{Cov}(E)$. The covariance loss is

$$\mathcal{L}_{\text{Cov}}^E = \frac{1}{c}(\underbrace{\sum_{i \neq j} [\text{Cov}(E)]_{i,j}^2}_{\text{off diagonal}} + \underbrace{\sum_k [\text{Cov}(E)]_{k,k}^2}_{\text{diagonal (variance)}}). \tag{4.14}$$

In contrast to VICReg, we minimize the diagonal variance terms as well, which represents an additional information bottleneck. In experiments, this regularization was helpful.

Since we train with supervision, there is no risk of ending up in a mode collapse, so the batch-variance criterion of VICReg is not required. Conversely, we found that

encouraging variance *along the feature dimension* of each vector with a hinge loss (penalizing vanishing features)

$$\mathcal{L}_{\text{Var}}^{E} = \frac{1}{|E|} \sum_{e \in E} \max\left(0,\ 1 - \sqrt{\text{Var}(e) + \epsilon}\right), \tag{4.15}$$

leads to better results and a more stable training. Finally, we define $\mathcal{L}_{\text{Cov}} = \mathcal{L}_{\text{Cov}}^{E_1} + \mathcal{L}_{\text{Cov}}^{E_2}$ and $\mathcal{L}_{\text{Var}} = \mathcal{L}_{\text{Var}}^{E_1} + \mathcal{L}_{\text{Var}}^{E_2}$, where E_1 and E_2 are the differently augmented variants of E.

4.2.3 *Training*

We use the VoxCeleb dataset (Nagrani et al., 2017) and prepare it using the video preprocessing code from Siarohin et al. (2019b). It consists of ∼3000 videos from 419 different identities divided into a total of ∼17.000 utterances with a resolution of 256×256. During training, we sample n_{src} source frames and one driving frame from the same video. Keypoints are extracted using the detector network of (Siarohin et al., 2019b), which is not trained further.

We train the rest of our method in three distinct phases. In all phases, we apply the regularization loss

$$\mathcal{L}_{\text{reg}} = \frac{\lambda_{\text{aug}}}{2}\left(\overline{\mathcal{L}_{\text{aug}}} + \overline{\mathcal{L}_{\text{aug,D}}}\right) + \lambda_{\text{Cov}}\mathcal{L}_{\text{Cov}} + \lambda_{\text{Var}}\mathcal{L}_{\text{Var}}, \tag{4.16}$$

where $\overline{\mathcal{L}_{\text{aug}}}$ and $\overline{\mathcal{L}_{\text{aug,D}}}$ are the mean values of $\mathcal{L}_{\text{aug}}$ and $\mathcal{L}_{\text{aug,D}}$ calculated over the entire batch. We start in Phase I with warm-up training, optimizing the MSE loss (Sajjadi et al., 2022b)

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{q \sim I_D} \|\mathcal{D}(q) - I_D(q)\|_2^2, \tag{4.17}$$

where we approximate $\mathbb{E}_{q \sim I_D}$ with 4096 sampled pixels.

Using only pixel-wise losses leads to blurry images (see Fig. 4.3). To address this issue, we propose to add the perceptual loss $\mathcal{L}_P$ (Johnson et al., 2016) in Phase II to generate more details. During our experiments, we found that the batch size must be large enough to avoid local minima and poor performance. Since training on the full frames already exceeds 80GB with a batch size of four, we compute gradients only sub-sampled to 128^2 pixels and compute the remaining pixels without gradient information. We apply a random pixel offset to ensure that all positions are covered during training. This trick allows us to estimate image-based losses without requiring costly gradient estimation for the entire image.

Finally, in Phase III, we then add adversarial losses $\mathcal{L}_A$, which guide the model to predict realistic images. Following Siarohin et al. (2019b), we utilize a CNN-based keypoint-aware discriminator $\mathcal{A}$ with 4 blocks and also add a feature matching loss $\mathcal{L}_A^F$ between the discriminator maps predicted from the generated image and the ground truth image.

The final loss in phase three is thus:

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}} + \lambda_P\mathcal{L}_P + \lambda_A\mathcal{L}_A + \lambda_A^F\mathcal{L}_A^F. \tag{4.18}$$

In our experiments, we train with a batch size of 24 on three NVIDIA A100 GPUs (80GB), for 200k iterations in Phase I, 300k iterations in Phase II, and approximately 500k iterations in Phase III, depending on the validation performance (see Sec. 4.4.1.2). We set $\lambda_{\text{MSE}} = 1$, $\lambda_P = 0.01$, $\lambda_A = 0.001$, $\lambda_A^F = 0.01$, $\lambda_{\text{aug}} = \lambda_{\text{Cov}} = 1$, and $\lambda_{\text{Var}} = 0.2$. Especially Phase I is very important to avoid overfitting. When skipped, we experience extremely slow training progress and easily end up in a bad local minimum.

4.2.4 *Inference*

For self-reenactment, the inference follows the training pipeline. In contrast, for cross-reenactment, the driving frame comes from a different person. This means that keypoints k_D can be taken as-is (absolute motion transfer) or adapted (relative motion transfer). This adaption is calibrated from a selected driving frame that best matches the head pose and expression (measured through the normalized facial keypoints of Bulat and Tzimiropoulos (2017)). Following Siarohin et al. (2019b), the scale is estimated by comparing the volume of the convex hull of facial keypoints. Driving keypoint movement is then scaled correctly and added to the keypoints of the source image.

In both cases, the facial expression vector does not depend on pose, shape, or appearance and is applied as-is, which is a particular advantage of our method.

4.3 Experiments

In this section, we carry out various experiments on the official VoxCeleb test dataset (Nagrani et al., 2017) with image size 256^2. Additional results are reported in the Supplementary Material (see Sec. 4.4). We compare against the state-of-the-art methods FOMM (Siarohin et al., 2019b), TSMM (Zhao and Zhang, 2022), DaGAN (Hong et al., 2022), OSFS (Wang et al., 2021b) (third-party implementation), and DPE (Pang et al., 2023).

4.3.1 *Self-reenactment*

In self-reenactment, the source image is selected as the first frame in the driving video. In the case of two source images, we also select the last frame. We then reconstruct every tenth frame within the video, ensuring that each driving frame is at least ten frames apart from the closest source image. We compare the animations to ground truth using the PSNR, SSIM (Wang, 2004), mean L1 error, and Average

Method	#KP	SSIM↑	PSNR↑	L1↓	AKD↓
DPE	0	.7180	22.94	.0484	3.07
FOMM	10	.7310	22.90	.0470	2.26
DaGAN	15	.7563	23.51	.0450	2.10
DaGAN/dv2[1]	15	.7346	22.81	.0493	2.50
OSFS[2]	15	.7327	22.97	.0471	2.33
TSMM	50	<u>.7660</u>	<u>23.76</u>	.0433	**2.00**
Ours/2-Src	10	**.7891**	**25.00**	**.0360**	<u>2.04</u>
Ours	10	.7576	23.67	<u>.0421</u>	2.13
$\|e\| = 128$	10	.7558	23.56	.0428	2.16
$\|e\| = 64$	10	.7535	23.61	.0430	2.18
small $\mathcal{D}$	10	.7548	23.60	.0430	2.17
$n_{\mathcal{K}} = 0$	0	.7445	23.56	.0436	2.64

Table 4.1: Self-reenactment results (including ablations) on the official VoxCeleb test set (Nagrani et al., 2017). Underlined values are the second best.

[1] Uses depth network trained on VoxCeleb2 (Chung et al., 2018) for inference
[2] Third-party implementation

Keypoint Distance (AKD). To compute the AKD, we utilize external facial keypoints provided by Bulat and Tzimiropoulos (2017).

In Tab. 4.1, we compare with related methods. Our multi-source ablation outperforms related methods in terms of accuracy. For single source images, we achieve state-of-the-art performance. While TSMM (Zhao and Zhang, 2022) slightly outperforms our method in SSIM, AKD, and PSNR, we note that they inpaint only disoccluded regions of the detected background. This produces nearly perfect reconstructions in static background areas that are also visible in the source image. Furthermore, our method generalizes much better for cross-reenactment and produces more temporally consistent animations, as highlighted with our user study (see Tab. 4.2).

While a low AKD and high SSIM value are good for self-reenactment, they often indicate that face shapes predicted by a model are highly dependent on the driving face structure. This, however, is detrimental for cross-reenactment, where the source appearance may be distorted by poorly matching driving keypoints (see shape deformations of related methods in Fig. 4.4). Also, for relative motion, the alignment assumption (explained in Sec. 4.2.4) is often not perfectly satisfied, leading to poorly matching keypoints. Our method is more robust to this (see Fig. 4.5), because we do not use the structure of the driving frame to estimate the optical flow and we encode appearance information invariant to the driving keypoints in the set-latent scene representation.

4.3.2 *Ablation Study*

We run an ablation study to compare quantitative results (see Tab. 4.1). A qualitative comparison and implementation details are reported in the Supplementary Material (see Sec. 4.4). Particularly, qualitative ablation study results comparing our models ablations (excluding Ours/$n_{\mathcal{K}} = 0$) are visualized in Fig. 4.9 (cross-reenactment) and Fig. 4.10 (self-reenactment).

Do we need keypoints? We report an ablation without keypoint encoding (Ours/$n_{\mathcal{K}} = 0$), i.e. all pose information is carried implicitly in the latent vector e, removing factorization of pose and expression, which makes relative motion transfer impossible. As can be seen in Tab. 4.1, this change results in worse self-reenactment performance. See Fig. 4.15 in the Supplementary Material for qualitative cross-reenactment comparisons.

What size should latent vectors have? Our reference model is trained with one source image and a latent expression vector of size $|e|$=256. As mentioned in Sec. 4.2.2 and visualized in Fig. 4.3, training without our proposed regularization leads to overfitting, resulting in shape deformation, color distortion, and background artifacts. With $|e|$=128 and $|e|$=64 latent expression dimensions, the self-reenactment performance decreases only slightly, showing that we can significantly reduce the amount of driving information being transmitted (e.g. for low-bandwidth videoconferencing) without losing much accuracy. In cross-reenactment, we also noticed a slight degradation in the transmission of facial expressions. However, the results are still good.

How efficient is the set-latent representation for decoding? We train a model (Ours/small $\mathcal{D}$) with a significantly smaller decoder (see Sec. 4.4.1.1 for architecture details). Interestingly, we achieve a self-reenactment performance close to the reference model. This indicates that the set-latent representation is very efficient to decode and already models facial dynamics. However, the sharpness of fine details, such as hair, is slightly degraded. Reducing the decoder capacity increases throughput from 11 fps to 23 fps, enabling real-time application on a single NVIDIA RTX 4090 GPU. As mentioned in Sec. 4.2.1, we can scale throughput linearly with the number of GPUs, while decreasing latency accordingly.

Can we improve results with multiple source images? Unlike state-of-the-art methods (Hong et al., 2022; Pang et al., 2023; Siarohin et al., 2019b; Wang et al., 2021b; Zhao and Zhang, 2022), our architecture allows the use of an arbitrary and flexible number of source images when available (e.g., when extracted from a video). Multiple source images can help the model understand person-specific face dynamics. In this experiment, we train a model ablation (Ours/2-Src) with two source images. As Tab. 4.1 shows, the results are significantly improved. Interestingly, the model generalizes to more than two source images even without explicit training, as examined in Sec. 4.4.2.1.

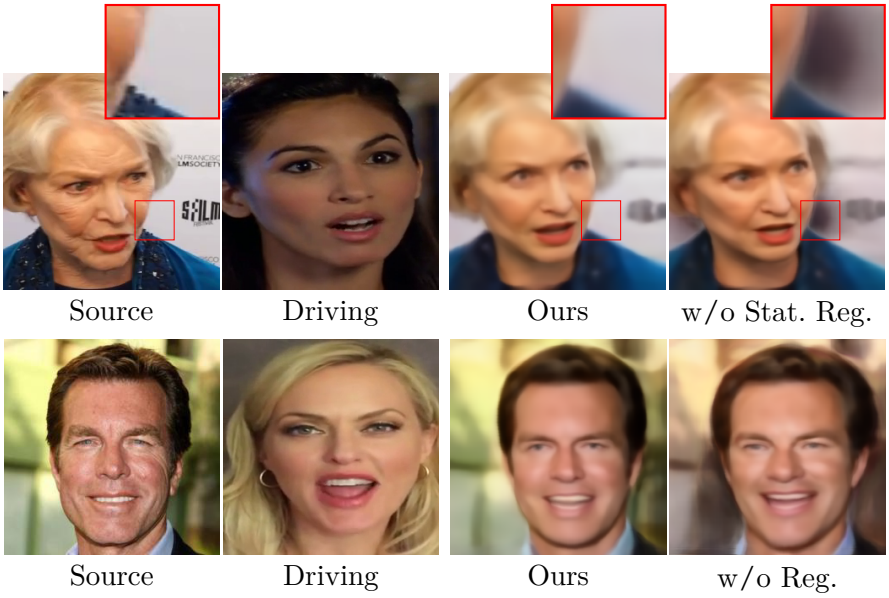

Figure 4.3: Regularization benefit in Phase I training (relative motion transfer). If trained without statistical regularization (w/o Stat. Reg.), artifacts originating from the driving frame are visible in the background around the face boundary. When dropping regularization entirely (w/o Reg.), color distortions, background artifacts, and shape deformations are clearly visible. The lower sequence uses a source image from the CelebA-HQ dataset (Karras et al., 2018). Remaining images are from the VoxCeleb test set (Nagrani et al., 2017).

4.3.3 *Cross-reenactment*

Our main motivation is to perform cross-reenactment. We sample 20 source images and driving videos from the official VoxCeleb (Nagrani et al., 2017) test set and compare our videos to state-of-the-art animations in a pairwise user study in Tab. 4.2. To be fair, we only use a single source image. We also present qualitative results in Figs. 4.4 and 4.5 and report additional user study information in Sec. 4.4.1.3. In general, our method is better at cross-ID motion transfer, while producing more consistent and natural results. For additional qualitative inference results on CelebV (Wu et al., 2018), CelebA-HQ (Karras et al., 2018), and VoxCeleb2 (Chung et al., 2018) see our Supplementary Material (Sec. 4.4).

	FOMM	DaGAN	DaGAN/dv2	TSMM	OSFS	DPE
Relative	97% (20)	98% (20)	95% (19)	97% (20)	87% (19)	
Absolute	94% (20)	99% (20)	96% (20)	92% (19)	94% (19)	95% (20)

Table 4.2: Cross-reenactment user study. Pairwise preferences between state-of-the-art and our method. Higher values show higher preference for our videos. DPE (Pang et al., 2023) has no relative mode. (·) shows the amount out of 20 scenes for which we got the majority of votes.

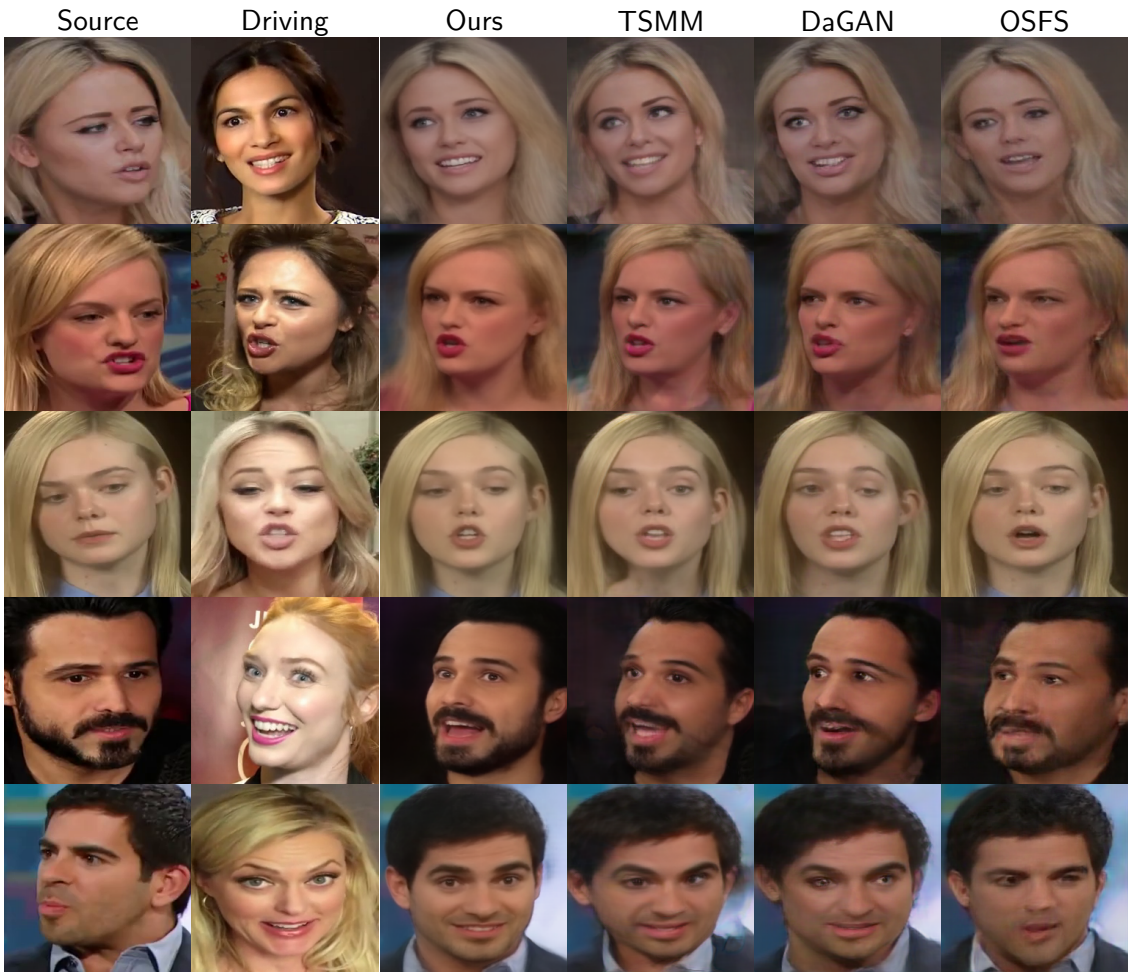

Figure 4.4: Cross-reenactment comparison with absolute motion transfer on the VoxCeleb test set (Nagrani et al., 2017). We generate more accurate expressions with less shape deformations (higher identity (ID) preservation).

Absolute Motion. When the driving keypoints are simply copied, users mainly prefer the animations generated by our method (see Tab. 4.2). Since our method is more robust to poorly matching keypoints, we produce fewer shape deformations than other keypoint-based methods (see Fig. 4.4). Furthermore, we consistently animate larger pose offsets.

Relative Motion. More interesting and challenging is animating with relative motion. Here, best performance can be achieved when the facial expression representation is decoupled from head pose and shape. As Tab. 4.2 and Fig. 4.5 illustrates, we significantly outperform previous state-of-the-art methods. When analyzing the results, we noticed that related methods show poor performance when there is no good match for the source expression and head pose in the driving video.

Figure 4.5: Comparison with state-of-the-art in cross-reenactment with relative motion transfer. Our method is more robust to the alignment assumption for relative motion transfer, generates more accurate expressions, and handles larger pose offsets. All images are from the VoxCeleb test set (Nagrani et al., 2017), except the lower block, which shows generalization to source images from the CelebA-HQ dataset (Karras et al., 2018).

4.3.4 *Limitations*

Our method struggles to generate out-of-distribution expressions such as sticking out the tongue or looking back. While we produce sharper mouth and eye regions, details in the background and hair are sometimes slightly reduced, compared to CNN-based methods. We believe that the model allocates most of its capacity to the face. Compared to CNN approaches that simply learn to forward background pixels from the input, our model must encode the background in the set-latents and reconstruct it by attending the correct features. Increasing model capacity or optimizing the query representation might lead to improvements.

4.4 Supplementary Material

4.4.1 *Implementation Details*

We present important training and architecture details, including the parameter values that were used.

4.4.1.1 *Architecture Details*

Keypoint Detector. The keypoint detector consists of a 5-block Hourglass network (Newell et al., 2016) with a block expansion of 32 and a maximum feature map size of 1024. For keypoint extraction, the images are resized to 64×64. After decoding, the heatmaps are predicted by a final 7×7 convolution. Keypoint locations are given by the centroids of the corresponding heatmap.

Latent Expression Extractor. The latent expression extractor $\mathcal{X}$ has a single 7×7 convolutional layer that predicts $n_f = 32$ individual feature maps for each keypoint. For each keypoint, the individual feature maps computed by the keypoint detector are aggregated in x and y direction with the weights of the corresponding heatmap. After aggregating the features of each keypoint individually, the information is concatenated and fused to predict a global expression vector. The fusion is performed by a 4-layer MLP with $(640 - 1280 - 640)$ hidden units and $|e|$ output neurons.

Input and Query Representation. For both, the positional encoding in the input and query representation, we set the number of octaves to $\mathcal{O}_{pix} = 16$ and $\mathcal{O}_{key} = 4$ with start octaves $s_{\mathcal{O}_{pix}} = -1$ and $s_{\mathcal{O}_{key}} = -1$. Together with a latent expression dimension of $|e| = 256$, this results in a query representation of size $|Q_{I_D}| = 416$ and an input representation R_{S_i} with 419 input channels, since we also encode the RGB pixel color of the source image.

Patch CNN. In all experiments, we set the output feature dimension of the Patch CNN to $n_{\mathcal{E}}^{fm} = 768$. Since we are processing a very large number of input channels (419 when $|e| = 256$), we use a bottleneck of 96 feature maps in the first convolutional layer.

Encoder. The transformer encoder also has a feature dimension of 768. Each multi-head attention layer uses 12 heads with an attention dimension of 64. The encoder processes the patch embedding of each source image individually, so that the cardinality of the set-latent scene representation scales linearly with the number of source images. This allows a flexible number of source images to be used. In total, the encoder and Patch CNN (with $|e| = 256$) have 29,774,112 parameters.

Decoder. The decoder has a feature dimension equal to the size of the query representation $|Q_{I_D}|$. The input MLP (see decoder in Fig. 4.2) has two layers with 720 hidden units and $|Q_{I_D}|$ output neurons. In the attention blocks, we also use 12

heads with an attention dimension of 64. The MLP inside the attention block, which fuses the information from the individual heads, has two layers and $2|Q_{I_D}|$ hidden units. The final 5-layer render MLP has $(1536 - 1536 - 1536 - 768)$ hidden units and three output neurons for the RGB color.

For our small decoder ablation Ours/small$\mathcal{D}$, we reduce the number of heads from 12 to 6 and also halve the number of hidden units of the MLP inside the attention block. Finally, we replace the render MLP with a smaller 3-layer version with $(1536 - 768)$ hidden units. Compared to our standard decoder, the number of parameters is reduced from 15,310,131 to 6,012,723.

Discriminator. For the keypoint-aware discriminator $\mathcal{A}$, we use the implementation of Siarohin et al. (2019b) which is based on (Isola et al., 2017). The input is an RGB image concatenated with ten heatmaps representing the driving keypoints. In total, we use four blocks, resulting in 512 output features with a downsampling factor of 16. For further implementation and loss details, we refer to Siarohin et al. (2019b).

4.4.1.2 *Training Details*

We train on three NVIDIA A100 (80GB) GPUs for about 23 days. We found that warming up (i.e. Phase I training, explained in Sec. 4.2.3) is essential to avoid ending up in local minima. Also, the batch size should be large enough. In our experiments we found out that 24 is sufficient. With a batch size of eight, training progressed slowly and appeared to be very unstable. Furthermore, we ended up in a local minimum with poor inference performance. When adding adversarial losses in training Phase III, we allow the discriminator to warm up for 500 iterations without computing gradients for the model. This is essential since otherwise the untrained discriminator will influence the current training progress with gradients of large magnitude.

Stopping Criterion. We extract a validation dataset, which we use to validate the self- and cross-reenactment performance. The self-reenactment performance is measured as in Tab. 4.1. For cross-reenactment, we randomly sample source images and driving videos. Model performance is judged visually by us. We found that it is not necessary to choose between good self- and cross-reenactment performance, as both are typically correlated. We thus use self-reenactment scores as a way to find promising models and then verify cross-reenactment performance.

Visualizing Out-of-frame Motion. As explained in Sec. 4.2.1, we use a negative octave in the positional encoding of pixels and keypoints to uniquely encode values in $(-2, 2)$. However, the VoxCeleb dataset (Nagrani et al., 2017) (prepared using the video preprocessing code from Siarohin et al. (2019b)) itself has no out-of-frame motion. Instead, we create out-of-frame motion by cropping the image with respect to the keypoints of the source images. We use external pre-estimated face keypoints (Bulat and Tzimiropoulos, 2017) and select a random crop of all selected

images (source and driving) such that the keypoints of all source images are inside. Finally, the images are resized back to 256×256, which may change the aspect ratio and induces additional regularization. In some cases, the driving face will now be partially outside the image—generating corresponding training samples.

Since cropping will reduce the image resolution to less than 256^2, we download the dataset at the highest resolution possible so that the crop (before resizing) is ideally larger than 256^2 and no image detail is lost.

The keypoint detector can only predict keypoints within the image. Therefore, we detect keypoints of the uncropped images and use the cropping information to transform them into the cropped images.

Unlike the source keypoints, the latent expression vectors are extracted directly from the cropped source images. When extracting expression vectors from the driving frame, the differently augmented driving frame version (as explained in Sec. 4.2.2), ensures that the driving face is inside the image. In Fig. 4.6, we show that not addressing out-of-frame motion leads to poor results when keypoints are outside the image or close to the image boundaries.

4.4.1.3 *User Study Details*

We selected 30 different people to participate in the user study (see Tab. 4.2). Since we compared the methods in pairs, each participant was only allowed to judge one related method. Furthermore, each participant judged both relative motion transfer and absolute motion transfer. The face reenactment task was initially explained, and participants were instructed to base their decision on the following two criteria:

1. Does the motion transfer work well (including ID preservation)?
2. Does the animation look like a natural and consistent video?

Each participant was simultaneously shown the source image, the driving video, our result and the animation of the comparison method. In each of the 20 sequences, we randomized whether our method was shown on the left or on the right. Participants could only decide once the video had run through. However, the video automatically restarted, so that there was no overall time limit. A decision was made by clicking on the preferred video.

4.4.2 *Additional Experiments and Results*

We report auxiliary experiments and more qualitative results here. We compare to the state-of-the-art methods FOMM (Siarohin et al., 2019b), TSMM (Zhao and Zhang, 2022), DaGAN (Hong et al., 2022), OSFS (Wang et al., 2021b) (third-party implementation), and DPE (Pang et al., 2023).

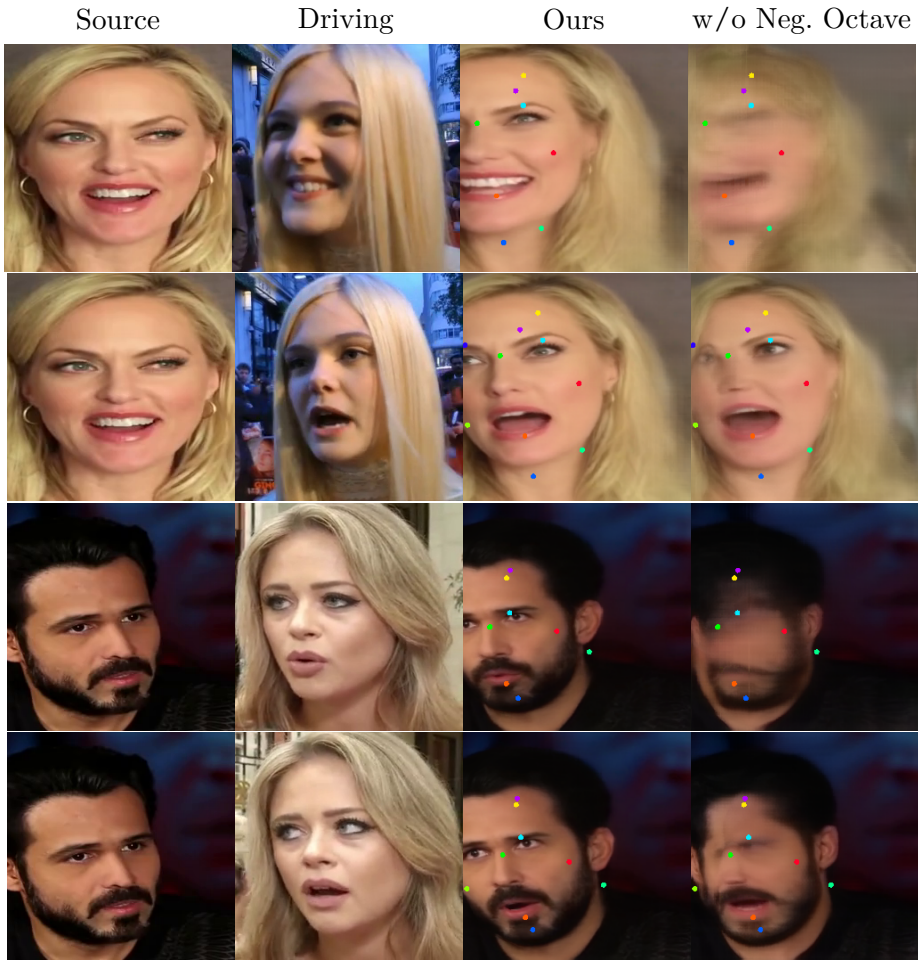

Figure 4.6: Out-of-frame motion with (Ours) and without explicit addressing keypoints outside the image (w/o Neg. Octave). Out-of-frame motion only occurs when relative motion transfer is used (see Sec. 4.2.4). The predicted images are visualized with the driving keypoints that were used in the decoder. Images from the VoxCeleb test set (Nagrani et al., 2017).

4.4.2.1 *Flexibility in the Number of Source Images*

We investigate the generalization behavior with respect to changing the number of source images during inference. Here, our reference model was trained with a single source image and with two source images. As reported in Tab. 4.3, the model trained with two source images generalizes in both directions, with fewer and with more source images used for inference. Interestingly, when reducing the number of source images to one (line Ours/2 $\rightarrow$ 1-Src in Tab. 4.3) it even produces slightly better self-reenactment results than our model explicitly trained with only one source image (line Ours in Tab. 4.3). With three source images available for inference (line Ours/2 $\rightarrow$ 3-Src in Tab. 4.3), the performance increases further, indicating that additional source images can be added at inference as available.

Method	SSIM↑	PSNR↑	L1↓	AKD↓
Ours	.7576	23.67	.0421	2.13
Ours/ 1 → 2-Src	.7181	23.06	.0453	2.42
Ours/ 2-Src	.7891	25.00	.0360	2.04
Ours/ 2 → 3-Src	.8092	25.80	.0325	2.00
Ours/ 2 → 1-Src	.7610	23.85	.0418	2.13

Ours/$t \rightarrow i$-Src means that the model trained with t source images is evaluated with i source images during inference.

Table 4.3: Self-reenactment results on the official VoxCeleb test set (Nagrani et al., 2017) when generalizing to a different number of source images without explicit training. Training with two source images increases self-reenactment performance, even when only one source image is used for inference.

The model trained with only one source image shows a significant drop in performance when the number of source images is increased during evaluation (line Ours/$1 \rightarrow$ 2-Src in Tab. 4.3). Therefore, if a flexible number of source images is desired, we recommend training with at least two source images. Alternatively, the number of source images can be chosen flexibly during training. To ensure that the data can still be batched, we recommend always selecting the maximum number of source images, but masking the set-latents of unnecessary source images in the attention module of the decoder.

4.4.2.2 *Ablation Study*

In Figs. 4.9 and 4.10 we present qualitative results of our ablations (see Sec. 4.3.2) in the cross- and self-reenactment situation, respectively. In terms of motion transfer accuracy, our reference model with $|e|$=256 produces slightly better results than models using $|e|$=64 or $|e|$=128.

By using two source images, information from both source images can be extracted and fused to produce more accurate animations. Especially if the second source image reveals occluded background or different head regions, less information has to be guessed by the model. As shown in Figs. 4.9 and 4.10, using multiple source images (Ours/2-Src) can help to produce animations with more detail in face, hair, and background.

Our ablation with a small decoder (Ours/small$\mathcal{D}$) has a motion transfer capability similar to our reference model (Ours), but with a slightly reduced sharpness in the animations.

4.4.2.3 *Comparison with State-of-the-Art Methods*

In Fig. 4.11 and Fig. 4.12 we present additional cross-reenactment results on the VoxCeleb test set (Nagrani et al., 2017) with relative and absolute motion transfer

Figure 4.7: Out-of-distribution results with relative motion transfer generated by our method. The source images are extracted from popular paintings and the driving frames are from the VoxCeleb2 test set (Chung et al., 2018).

compared to all state-of-the-art methods from our user study (see Tab. 4.2). While TSMM (Zhao and Zhang, 2022), DaGAN (Hong et al., 2022), OSFS (Wang et al., 2021b), and FOMM (Siarohin et al., 2019b) are also keypoint based, DPE (Pang et al., 2023) uses a latent head pose description. This, however, eliminates the ability to perform relative motion transfer. As the visualizations show, our method produces significantly more natural results with higher ID preservation and more accurate and plausible motion transfers. Especially when there is a large pose offset, related methods often fail to produce satisfactory results. For animated results, see our project page.[3]

4.4.2.4 *Out-of-Distribution Animation*

As shown in Fig. 4.7, our model trained on VoxCeleb (Nagrani et al., 2017) generalizes to out-of-distribution source images extracted from popular paintings.

4.4.2.5 *Generalizing to other Datasets*

We report generalization examples of our models trained on VoxCeleb (Nagrani et al., 2017) to other datasets (VoxCeleb2 (Chung et al., 2018), CelebA-HQ (Karras et al., 2018), and CelebV (Wu et al., 2018)) at inference time. Specifically, we show the following source → driving combinations:

- CelebA-HQ → VoxCeleb2 in Fig. 4.13,
- VoxCeleb2 → VoxCeleb2 in Fig. 4.14, and
- CelebV → CelebV in Fig. 4.15.

We note that VoxCeleb2 covers a significantly larger number of identities in the test set compared to VoxCeleb. As the results show, our model generalizes to all of these combinations, while still producing more accurate animations compared to related methods.

4.4.2.6 *Omitting Keypoints*

We present qualitative results of our model ablation Ours/$n_{\mathcal{K}} = 0$ without keypoints in Fig. 4.15. Compared to our reference model (Ours), we found that the accuracy of the motion transfer is slightly reduced. In particular, the animated gaze direction seems to be less accurate (see third row in Fig. 4.15). Omitting the keypoints makes it impossible to perform relative motion transfer, since all pose information is implicitly encoded in the expression vector e.

In this variant, images input to the expression network are not augmented through cropping, since this makes recovery of the head pose impossible without keypoints. However, we discovered that performing a random center crop with variable aspect ratio on the driving frame (while requiring the network to reconstruct the full driving frame) reduces shape deformations, since the network becomes invariant against

3 https://andrerochow.github.io/fsrt

Method	#KP	SSIM↑	PSNR↑	L1↓	AKD↓
Ours	10	.7576	23.67	.0421	2.13
$n_{\mathcal{K}} = 0$	0	.7445	23.56	.0436	2.64
+Crop Aug.	0	.7240	22.98	.0469	2.99

Table 4.4: Self-reenactment results on the official VoxCeleb test set (Nagrani et al., 2017). We compare our model ablation without keypoints (Ours/$n_{\mathcal{K}} = 0$) with an ablation that is additionally trained with random center cropping (Ours/$n_{\mathcal{K}} = 0$ + Crop Aug.). The scores of our reference model (Ours) are shown in the first row.

aspect ratio changes and scale (see Ours/$n_{\mathcal{K}} = 0$ + Crop Aug. in Fig. 4.8). While this might be useful in cross-reenactment applications where relative motion transfer is not required, it reduces self-reenactment scores (see Tab. 4.4)—where this invariance is not helpful but actually harmful. A particular reason for this might be that this variant cannot transfer zooming or dolly shots due to scale invariance.

4.4.2.7 *Statistical Regularization*

In Fig. 4.16, we visualize the effect of training without our proposed statistical regularization (see Sec. 4.2.2). As the results show, training without statistical regularization leads to significant artifacts around the animated face region, indicating that ID information leaks from the driving frame through the expression vector e_D. Our proposed factorization is therefore not achieved.

Figure 4.8: Ablations without keypoints. This comparison is using absolute motion transfer. When combining a keypoint-less model with random center cropping during training (right column), shape deformations and scale changes are prevented. The images are from the VoxCeleb test set (Nagrani et al., 2017), the VoxCeleb2 test set (Chung et al., 2018), and the CelebA-HQ dataset (Karras et al., 2018) (as indicated by the source → driving notation).

Figure 4.9: Ablation study in cross-reenactment on the VoxCeleb test set (Nagrani et al., 2017) with absolute motion transfer (upper block) and relative motion transfer (lower block). Our ablation Ours/2-Src consistently fuses the information of both source images. It produces more detail in the face, hair, and background, especially when the second source image reveals information missing in the first source image.

Figure 4.10: Ablation study in self-reenactment on the VoxCeleb test set (Nagrani et al., 2017). The accuracy of motion transfer (especially mouth and eye motion) decreases slightly when reducing the size of the latent expression vector e. In the first and fourth animation, Ours$|e|$=64 produces inaccurate mouth expressions. Ours/2-Src generates more detail by integrating the information from both source images.

Figure 4.11: Comparison with state-of-the-art on the VoxCeleb test set (Nagrani et al., 2017) in cross-reenactment (relative motion transfer). Our model generates more accurate expressions, is less sensitive to the alignment assumption (Sec. 4.2.4), and learns to realistically fill missing face parts (third row). Others often produce mismatched expressions and fail for large pose offsets. The last row shows a source image from CelebA-HQ (Karras et al., 2018).

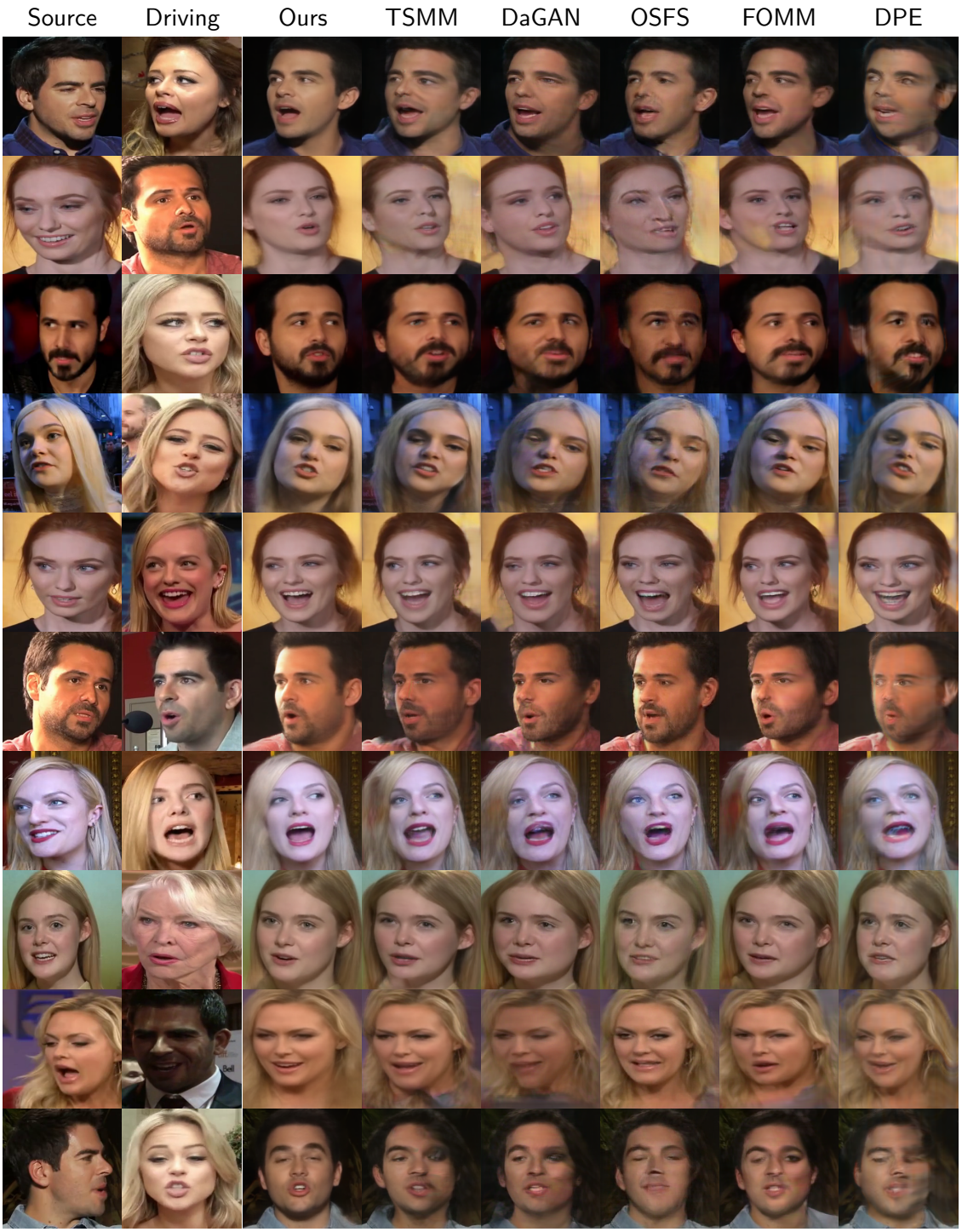

Figure 4.12: Comparison with state-of-the-art on the VoxCeleb test set (Nagrani et al., 2017) in cross-reenactment with absolute motion transfer. We generate more accurate facial expressions with better ID preservation. Related methods often produce strong shape deformations, artifacts and blurry results (especially in the mouth region). The sixth animation shows that our method even animates the sunlight on the side of the face.

Figure 4.13: Cross-reenactment generalization to driving videos from the VoxCeleb2 test set (Chung et al., 2018) and source images from the CelebA-HQ dataset (Karras et al., 2018) with relative motion transfer.

Figure 4.14: Cross-reenactment generalization to driving videos and source images both from the VoxCeleb2 test set (Chung et al., 2018) with relative motion transfer.

Figure 4.15: Comparison of our model with and without keypoints and state-of-the-art methods in cross-reenactment with absolute motion transfer. The top block shows generalization to source and driving frames extracted from the CelebV dataset (Wu et al., 2018). The bottom block shows generalization to driving frames extracted from the VoxCeleb2 test set (Chung et al., 2018) and source images from the CelebA-HQ dataset (Karras et al., 2018).

Figure 4.16: Benefit of statistical regularization (relative motion transfer). Training without statistical regularization leads to visible artifacts around the animated face (see red arrows), indicating that the identity of the driving person is leaking into the expression vector e_D. Images are from the VoxCeleb test set (Nagrani et al., 2017) (indicated with *) and the VoxCeleb2 test set (Chung et al., 2018) (remaining).

4.5 VR-FSRT: Leveraging FSRT for VR Facial Animation

This experiment is not part of the original paper (Rochow et al., 2024) on which this chapter is based. We demonstrate that FSRT can be adapted for the VR facial animation task, while only being trained on talking-head datasets. In contrast to Chapters 2 and 3, this variant bypasses the necessity of acquiring a keypoint mapping between the HMD mouth camera image space and the source image space (see Sec. 2.2.4). Optimizing this keypoint mapping requires the time-consuming capture of two distinct videos of the operator (i.e., one HMD mouth camera video and one video without the HMD).

4.5.1 *Recently Published Related Work*

After our methods from Chapters 2 and 3 were developed and published, other methods for conducting VR facial animation were released (Bai et al., 2024; Patel et al., 2024; Tran et al., 2024; Zhang et al., 2024).

Zhang et al. (2024) propose generating a subject-specific blendshape model (Blanz and Vetter, 1999) and learn to predict the blendshape coefficients, head pose, and texture from HMD camera input images. The blendshape coefficients are then used to reconstruct the mesh. The model is trained using the texture and mesh rendered from HMD camera views, in a self-supervised manner.

Tran et al. (2024) present a 3D face reenactment method that uses an expression vector to modify intermediate features of the pretrained real-time 3D lifting method proposed by Trevithick et al. (2023). This enables generating a tri-plane representation of the source image with the facial expression extracted from the driving frame. Novel views with arbitrary camera poses are then generated using volume rendering. Based on this method, VR facial animation is performed step by step. First, they use data captured from the built-in sensors of the Meta Quest Pro VR headset, along with the Meta Movement SDK (Meta, 2024b) and the Meta Headset Tracking SDK (Meta, 2024a), to drive a generic parametric blendshape model of a head in the Unity game engine (Unity, 2024). This rendered mesh is then used as the driving frame in the proposed face reenactment method, where two stereoscopic views are rendered so that they can be displayed in another HMD. However, their VR facial animation performance is constrained by two factors. Firstly, it is limited to the motion transfer accuracy of the generic mesh animation. Secondly, it is constrained by the performance of the face reenactment method when using synthetic generic meshes as driving frames. We note that their face reenactment method can be replaced by other existing methods (such as FSRT), potentially followed by 3D lifting (Trevithick et al., 2023) to allow generating stereoscopic views.

Patel et al. (2024) present an iterative method for registering an avatar face model to HMD camera images of unseen individuals. They utilize the universal avatar face model from Cao et al. (2022) to create a personalized avatar from identity information obtained from either a light stage or a phone scan, and render it from

a certain viewpoint and with a certain expression. In each iteration, the avatar is rendered with the current expression and viewpoint estimate. The renderings of the current iteration, along with the raw HMD camera images and the style-transferred HMD camera images, are employed to update the estimates. Corresponding ground truth expression and viewpoint data of each individual is obtained using costly subject-specific training. Here, they utilize the method of Schwartz et al. (2020) but with a unified latent expression space (Cao et al., 2022) for all individuals. Their approach allows registering HMD camera images of unseen individuals to their avatar model in roughly 0.4 seconds per frame. They also present a direct regression model, trained on the same ground truth labels, that enables rendering unseen individuals in real-time by directly predicting the latent expression codes. Their iterative approach significantly outperforms the direct regression method.

Finally, Bai et al. (2024) generate subject-specific input-groundtruth correspondences of HMD camera inputs and ground truth latent codes for expression and gaze. The expression and gaze codes can be used to animate a universal avatar face model based on (Cao et al., 2022) (a single model shared across all identities), which only requires the encoded identity for adapting to a certain person. They then learn to predict expression and gaze codes from the HMD camera images with an universal facial encoder network that only needs to be calibrated for a new operator with multiple anchor expressions captured by the HMD cameras. This allows to encode facial motion from HMD camera inputs of arbitrary identities and thus to animate arbitrary avatars during inference. The preprocessing time for a new operator is short, since only images of the operator's appearance (to encode the identity) and anchor expressions must be captured.

Training the methods proposed by Bai et al. (2024) and Patel et al. (2024) requires significant amounts of data captured with camera domes and HMD cameras of many different identities. Such data is available in the recently published Ava-256 dataset (Martinez et al., 2024), which contains more than 200 million images of 256 different identities. In contrast, the VR facial animation methods presented in this thesis can primarily be trained on talking-head datasets. For a discussion on leveraging Ava-256 data for training our methods, we refer to Chapter 5.

4.5.2 *Required Modifications to FSRT*

In contrast to the standard FSRT approach of extracting the expression vector from a complete face image, we propose an alternative method that involves adapting the network learning to extract facial expression features exclusively in the lower facial region. Additionally, we remove one keypoint, that determines the position of the lower lip, from the output of the FSRT keypoint detector. This modification is crucial in ensuring that the mouth expression is entirely defined by implicit latent representations (i.e., the driving expression vector). Consequently, it becomes unnecessary to translate keypoints from the mouth camera image to the source image space.

4.5.2.1 *Extended Architecture*

The only component of the model architecture that was modified is the expression encoder. The standard FSRT expression encoder recycles feature maps of the keypoint detector and requires keypoint information of the entire face to extract the expression vector. Instead, a ResNet-50 (He et al., 2016) is employed, where the final fully connected layer is replaced by a MLP with $(1280 - 1280 - 640)$ hidden units and $|e|$=64 (expression vector size) output neurons.

In order to achieve real-time capability on a single NVIDIA RTX 4090 GPU, we employ the small FSRT decoder network (see Sec. 4.4.1.1).

4.5.2.2 *Augmentation and Training*

We precompute facial keypoints of image I using the method proposed by Bulat and Tzimiropoulos (2017) and utilize jaw outline keypoints k_I^C and mouth keypoints k_I^M (the same that were used for the "full VR keypoints" in Chapter 3) to determine a keypoint-aware random crop of the lower facial region. This crop is required for extracting the latent expression vector.

As with FSRT, we define three distinct color-jitter augmentations A_1, A_2 and A_3 (see Sec. 4.2.2). While augmentation A_3 is exclusively applied to the driving frame, A_1 and A_2 are also applied to the source image.

For all image augmentations A_{1-3} we define the cropping value at the top to be a random variable

$$c_{top} \sim U\left\{\min_y(k_I^M) - 25, \min_y(k_I^M)\right\}, \tag{4.19}$$

where $U\{a, b\}$ is the discrete uniform distribution on the interval $\{x \in \mathbb{Z} \mid a \leq x \leq b\}$. The remaining cropping values on the left, right, and bottom (c_{left}, c_{right}, and c_{bottom}, respectively) are determined differently for images with augmentation $A_{1,2}$ and A_3. For images augmented with A_1 and A_2, the remaining cropping values (left, right, and bottom, respectively) are defined as the random variables

$$c_{left} \sim U\left\{0, \min_x(k_I^C)\right\}, \tag{4.20}$$

$$c_{right} \sim U\left\{\max_x(k_I^C), 255\right\}, \tag{4.21}$$

$$c_{bottom} \sim U\left\{\max_y(k_I^C), 255\right\}. \tag{4.22}$$

For images augmented with A_3 we crop in a close interval ($\{-10, 10\}$) around $\min_x(k_I^C)$, $\max_x(k_I^C)$, and $\max_y(k_I^C)$ enforcing that scale invariant features are extracted only in the lower facial region, especially when combined with the invariance loss in Eqs. (4.11) and (4.12). Prior to expression extraction, the cropped lower-face images are resized to a fixed size of 128 pixels in width and 96 pixels in height.

We also implement the VR-FSRT training procedure in the three phases (Phases I, II, and III described in Sec. 4.2.3). However, we make adjustments to the timing when specific regularization losses are activated.

We initialize with a Resnet-50 model that has been pretrained on the facial expression recognition dataset Affectnet-8 (Mollahosseini et al., 2017). The pretrained weights and implementation of the ResNet-50 are provided by Zhao et al. (2021b). We finetune the pretrained ResNet-50 model, because it was not trained on lower-face images, and it is constrained to only eight distinct facial expressions, leading to insufficient expression transfer outcomes without additional finetuning. Since the gradients in the first iterations are not very meaningful, we initially freeze the weights of the pretrained Resnet-50 for 100k iterations to prevent loss of knowledge from pretraining.

The invariance losses (see Eq. (4.11) and Eq. (4.12)) must be deactivated for 100k iterations. In our experiments, activating them directly led to a bad local minima, resulting in a lack of motion in the lower facial region. The other regularization losses $\mathcal{L}_{Cov}$ (see Eq. (4.14)) and $\mathcal{L}_{Var}$ (see Eq. (4.15)) are activated after 200k iteration, which yields significantly better inference outcomes compared to activating them from the first iteration or training without them.

For training and inference, we utilize only a single input source image. VR-FSRT models are trained on the VoxCeleb dataset (Nagrani et al., 2017), prepared using the video preprocessing code from Siarohin et al. (2019b), with an image size of 256×256. Moreover, we train for roughly 1.9 million iterations (200k in Phase I, 300k in Phase II, and the remaining in Phase III) with a batchsize of 24 using 3 NVIDIA A100 GPUs, which takes approximately 23 days.

4.5.2.3 *Reference Model*

During inference, when the driving keypoints are fixed but the mouth is moving, the upper face region of the animated person is not always perfectly static. While we were able to improve the head staticity by using more than four keypoint octaves to encode the keypoint locations, this yielded a significant decrease in overall performance. Interestingly, we found that we can significantly improve the head staticity of our model with four keypoint octaves (default value in FSRT) by training without encoding of the source expression vector (VR-FSRT$^{\text{w/o } e_S}_{\mathcal{O}_{key}=4}$), while achieving similar motion transfer performance in the lower facial region. Accordingly, we define this model as our reference model, designated VR-FSRT.

Further research is needed to explore other potential solutions to the staticity problem.

A qualitative comparison of VR-FSRT and our method from Chapter 3 is visualized in Figs. 4.17 and 4.18, where Fig. 4.18 shows failure cases where our method from Chapter 3 performs better in reconstructing the gaze direction or the lower facial region.

4.5.3 *Preprocessing and Inference*

In the preprocessing stage, the operator's source image is captured initially. Subsequently, the interactive annotation tool, as outlined in Sec. 2.2.7, is employed to annotate the eye coordinate system. Concurrently, the eye tracking of the HMD is calibrated. The operator-specific training of our self-developed eye tracking (described in Sec. 2.2.7), including the acquisition of calibration/training data, is a relatively brief process, typically taking a bit more than a minute. The crop region of the mouth camera image is determined automatically using the mouth and jaw outline keypoints predicted by $\mathcal{K}'_{\mathcal{VR}}$ (see Sec. 3.2.3) of our VR facial animation method from Chapter 3. This is done by selecting the smallest region containing all keypoints, followed by the incorporation of additional margins. Note that the crop region is fixed after the initial determination.

During inference, the initial step consists of encoding the source image and predicting the set-latent representation, denoted as $\{z_z\}$. The subsequent step involves the construction of the driving keypoints. This is achieved by first copying the source image keypoints and then replacing the eye keypoint with the eye tracking result transformed into the normalized gaze coordinate system defined in the source image (see Sec. 2.2.7). We then extract the expression vector from the current cropped mouth camera image and predict the output pixels using the transformer decoder, conditioned on the expression vector and constructed driving keypoints.

VR-FSRT achieves a decoder throughput of approximately 26 fps or 23 fps including the driving expression vector extraction on a single NVIDIA RTX 4090 GPU.

4.5.4 *Results and Discussion*

A quantitative comparison with our previous VR facial animation methods (as reported in Tab. 3.1) is not applicable in this case, as they are evaluated using the ground truth driving keypoints of the lower facial region, where the losses are evaluated. However, we provide a visual comparison in Fig. 4.17 to demonstrate that VR-FSRT reaches good visual performance. Overall VR-FSRT produces less accurate expressions and more blurry results (see Figs. 4.17 and 4.18). Furthermore, our method from Chapter 3 produces a more accurate gaze direction (see Fig. 4.18).

Benefit of VR-FSRT. While we do not claim to outperform our VR facial animation method from Chapter 3, we reduce the preprocessing time significantly, making it more valuable and applicable when less preparation time is available. The methods utilized at the ANA Avatar XPRIZE competition (see Chapters 2 and 3) required a preprocessing time of roughly 15 minutes, of which a significant amount was allocated to capturing the two source videos necessary to optimize the keypoint mapping (see Sec. 2.2.4). The time required to capture the videos constituted a substantial proportion of the total preparation time (60 min), which could otherwise have been allocated for training the operator to control our avatar system. As

operator training time directly correlates with task performance (Lenz et al., 2025), a reduction in facial animation preprocessing time is highly advantageous. The total preprocessing time of VR-FSRT required for adapting to a new operator (including steps where the operator is not actively involved) is minimal, typically taking around three minutes.

Gaze Direction. The gaze direction and eye openness are only defined by a single keypoint on the left eye (see Fig. 2.6). Since the eye keypoint also moves relative to the head pose, the model must learn to separate the eye keypoint motion corresponding to a change in head pose from the eye keypoint motion corresponding to a change in gaze direction. The latter is typically minimal since the eye is relatively small. That this is a challenging task is also reflected in the training procedure, where the model requires a significant amount of training time before beginning to extract meaningful gaze directions from this keypoint. A more explicit representation of gaze direction, such as the pupil center coordinate of each eye relative to a normalized eye coordinate system, could improve the gaze accuracy and also reduce the network capacity required for reconstructing the gaze direction.

Future Work. This experiment demonstrated the effectiveness of our FSRT method adapted to the VR facial animation task. VR-FSRT was trained exclusively on talking-head datasets, which generates a considerable domain gap during inference. In future research, we could explore methods to address this domain gap, with the objective of achieving substantial improvement of motion transfer accuracy within the lower facial region. Additionally, exploring methods to improve the reconstruction accuracy of the gaze direction is necessary to achieve a performance similar to our VR facial animation method presented in Chapter 3. Whereas VR-FSRT was exclusively trained on talking-head datasets, our method presented in Chapter 3 was finetuned with manually annotated pairs of mouth camera images and complete face images with roughly matching lower-face expressions. It is expected that the performance of VR-FSRT would also improve by integrating these pairs into the training procedure.

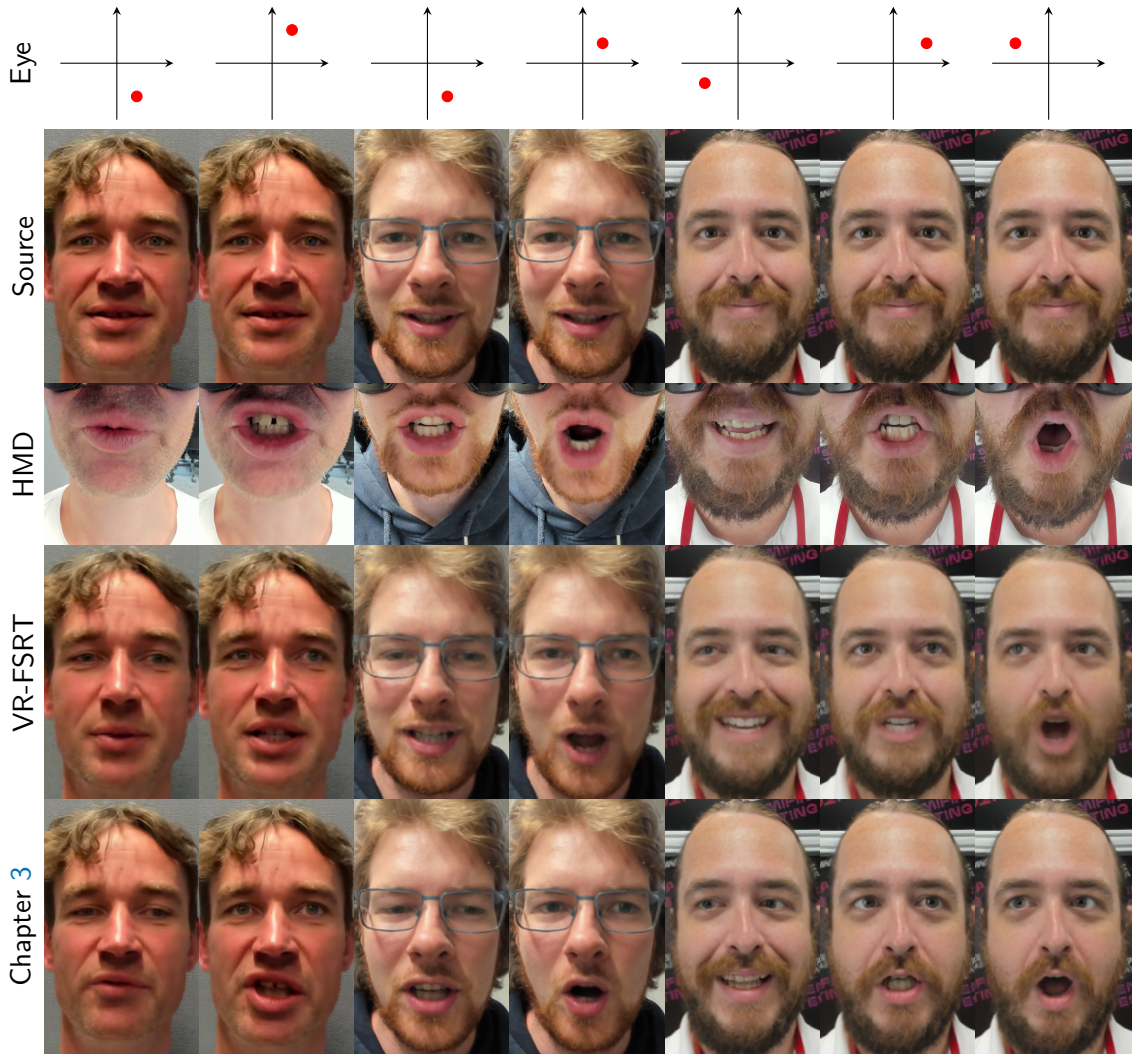

Figure 4.17: Qualitative VR-FSRT results compared to our VR facial animation method from Chapter 3. All images are cropped for visualization. The crop of the HMD mouth camera image used for the expression extraction is different to the crop visualized here.

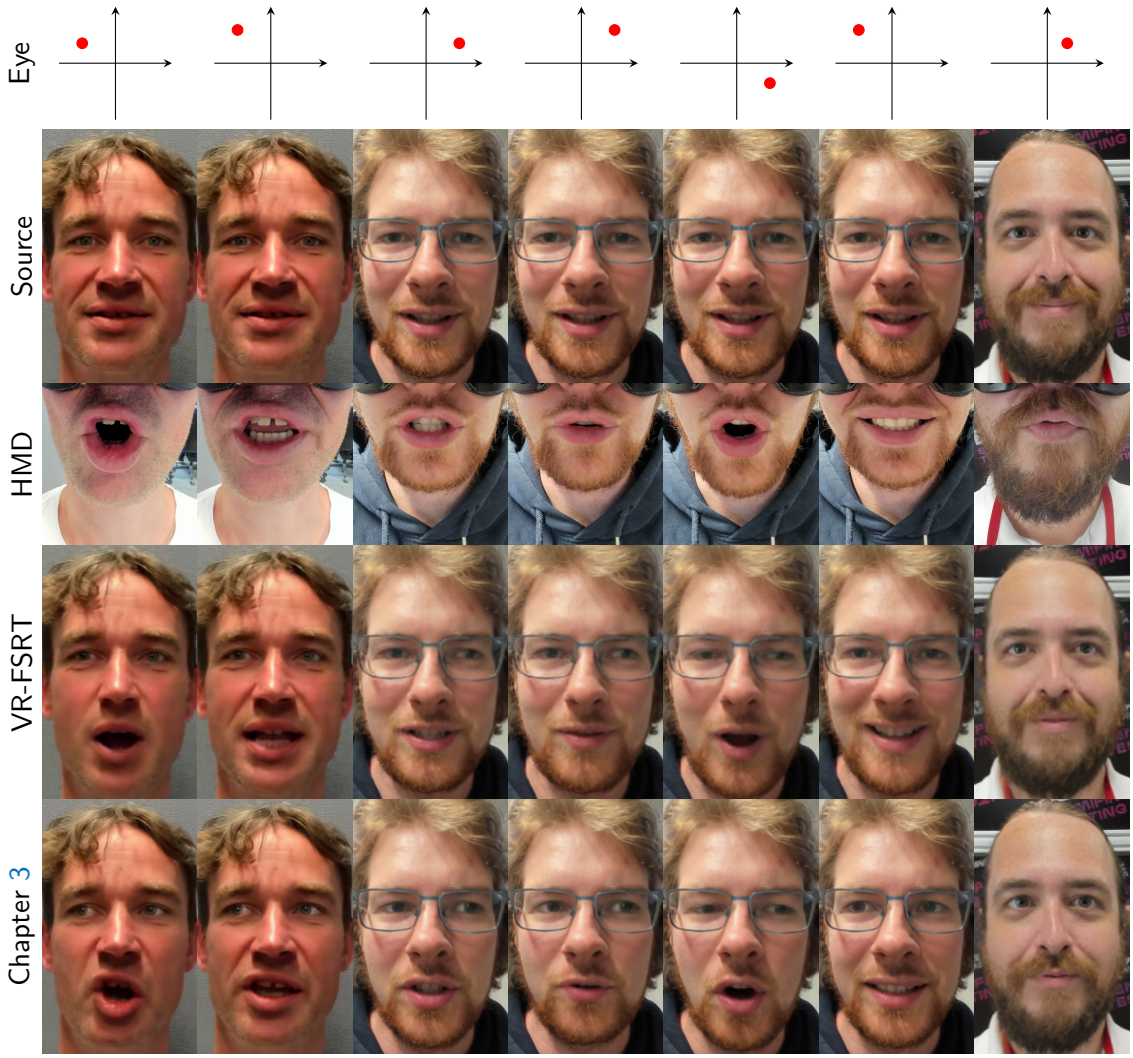

Figure 4.18: Limitations of VR-FSRT. Inaccurate gaze reconstructions generated by VR-FSRT are shown in columns 1-6. Examples where our method from Chapter 3 generates better animations in the lower facial region are mainly presented in columns 1, 2, 4, and 7. All images are cropped for visualization. The crop of the HMD mouth camera image used for the expression extraction is different to the crop visualized here.

4.6 Conclusion

We have proposed a state-of-the-art method for face reenactment. To our knowledge, this is the first transformer-based architecture for this purpose. We learn latent expression features that are free of appearance, shape or pose information, making them perfectly suited for cross-reenactment. Our method achieves fast inference speed, which allows real-time application. We proposed a regularization and training scheme which are necessary to guide the network to represent the scene as desired. Future work could investigate further improving the animation quality of fine details (e.g. in the hair) and utilizing volume rendering techniques to reconstruct geometry. As demonstrated in a separate experiment, the method can be adapted to the VR facial animation task, which was primarily examined in Chapters 2 and 3 of this thesis.

5

Conclusion and Outlook

In this thesis, we presented novel methods for face reenactment (Chapter 4) and VR facial animation (Chapters 2 and 3). Our VR facial animation methods contributed to the outstanding success of team NimbRo (Lenz et al., 2025; Schwarz et al., 2023) at the ANA Avatar XPRIZE competition.

The method from Chapter 2, which was employed in the semifinals, permits the animation of novel operators without the necessity for subject-specific training. As a result of limiting the driving information to keypoints and data retrieved from these keypoints, we are able to construct imaginary driving keypoints derived from source image keypoints, mapped mouth camera keypoints, and eye tracking results projected into the eye coordinate system defined in the source image. This enabled us to train our method in terms of face reenactment (video reconstruction) on the publicly available VoxCeleb dataset (Nagrani et al., 2017), which contains talking-head videos of celebrities extracted from YouTube videos. We thus circumvented the alignment problem of mouth camera images from the HMD and complete face images without an HMD. The training of our method on such large datasets with considerable variance in appearance enables generalization to new, previously unseen individuals.

In order to achieve more precise facial animations in the lower facial region, we developed an image retrieval mechanism that identifies an expression frame with a higher probability of aligning with the mouth camera keypoints than the fixed source image. The expression frame enables the network to better adapt to person-specific mouth dynamics and create more accurate facial animations. In order to reduce temporal inconsistencies caused by expression frame changes, a recursive low-pass filtering approach was proposed. We developed an eye tracking method, which is trained in less than a minute with calibration/training data captured from two eye cameras mounted inside the HMD.

Our team achieved a score of 99 out of 100 points at the ANA Avatar XPRIZE competition semifinals, where facial animation was a relevant factor in five different scoring criteria.

For the ANA Avatar XPRIZE finals, the VR facial animation pipeline was extended in Chapter 3 to address the primary limitations: Temporal inconsistency and limited facial expressions that can be displayed as a result of keypoint ambiguities. Two significant features were incorporated into the method: (i) visual mouth camera guidance and (ii) multiple source images that are combined in feature space with a learned attention mechanism.

The visual mouth camera guidance facilitates the direct injection of visual information from the mouth camera image into the animation pipeline. This allows us

to animate a broader range of facial expressions and significantly improve motion transfer accuracy. In comparison to the keypoint-based expression frame retrieval, which constitutes the sole visual guidance in our baseline approach from Chapter 2, the mouth camera information is not constrained by the ambiguities inherent to keypoints. Furthermore, incorporation of the visual mouth camera guidance provides a smoother input signal compared to the discontinuities induced by the expression frame selection process, improving temporal consistency.

The employment of multiple source images provides the network with a variety of expressions in the lower facial region. Consequently, the network is able to adapt to the person-specific facial dynamics of each individual. The learned attention mechanism allows the network to weight the source images according to their relevance to the current mouth camera keypoints. The method may be employed with only fixed source images or with the last source image treated as a retrievable dynamic source image, as proposed in Chapter 2. Experimental results indicated that the use of only fixed source images produces comparable visual outcomes but slightly inferior quantitative results. However, this approach demonstrates enhanced temporal consistency compared to the second variant, where the last source image is dynamic. Nevertheless, even the second variant demonstrates a notable improvement in temporal consistency compared to the baseline approach (Chapter 2). The use of multiple source images and visual mouth camera guidance effectively mitigates the impact of the dynamic source image.

To retain the ability to train on large-scale talking-head datasets that guarantee generalization to unseen persons, we emulate mouth camera inputs from such samples. We addressed the remaining domain gap between emulated and real mouth camera images by additionally presenting our networks with samples from a very small dataset of manually annotated HMD mouth camera images and corresponding complete face images featuring roughly matching expressions.

Our team NimbRo emerged victorious in the ANA Avatar XPRIZE competition, receiving a prize of five million USD and achieving a perfect score of 15/15 points.

Following the development of methods for VR facial animation, the focus of Chapter 4 was directed towards the face reenactment task, which is closely related. One specific objective was to develop a method that learns to predict latent expression features that are decoupled from appearance and head pose. Appearance invariant expression features are of particular importance in the context of cross-reenactment, wherein the driving frame can be of a different identity. Head-pose invariance additionally enables the unmodified utilization of expression vectors derived from a head pose other than the target head pose. This is crucial when animating with relative motion transfer, where the driving and target head poses may differ.

Consequently, we have developed FSRT, a transformer-based method that initially encodes the appearance of a flexible amount of source images into a set of latent representations. Subsequently, the transformer decoder is capable of predicting pixels of the output image with the desired head pose and facial expression by conditioning the set-latent representation with keypoints encoding the head pose

and a latent vector encoding the expression. The method can be trained on large-scale public available talking-head datasets. As the method is trained in the video reconstruction (self-reenactment) regime, it is prone to overfitting, which impairs the generalization ability during inference. We therefore proposed several augmentation and regularization methods that support the factorization of appearance, head pose, and expression, thereby ensuring that the representation is perfectly suited for cross-reenactment. Our method outperforms previous state-of-the-art methods in terms of consistency and motion transfer quality at cross-reenactment, as indicated by the results of a user study.

We then constructed a modified version of FSRT (Chapter 4), namely VR-FSRT, that performs VR facial animation instead of face reenactment. Since only a single source image is required instead of capturing two videos of the operator (see methods from Chapters 2 and 3), the animation of new operators requires a remarkably reduced preprocessing time of approximately three minutes. In comparison, our VR facial animation method from Chapter 3 requires approximately 15 minutes for adaption to a new operator—but achieves better animation performance.

Outlook. We identify several aspects of our methods where further research might yield improvements. In our VR facial animation methods, the definition of the eye coordinate system (see Sec. 2.2.7) in the source image is a relatively quick process, though it necessitates manual annotation. Ideally, this coordinate system is automatically defined within the source image, to save the time that is required for manual annotation.

All our presented VR facial animation methods could be improved in animating facial regions that are covered by the HMD. The only data extracted from the interior of the HMD is the gaze direction and a Boolean value for closed eyes. Other upper-face expressions, such as eyebrow motion, are neglected. Here, the primary challenge lies in the generation of appropriate training data. Wei et al. (2019) address this problem by learning a 3D avatar model that can be animated with latent expression codes learned in a self-supervised manner, using an expression-preserving image style transfer from real HMD camera images to the synthetic avatar domain. However, this approach does not generalize to unseen individuals during inference, as it requires training on large amounts of subject-specific data. Recently, Martinez et al. (2024) published the Ava-256 dataset which contains more than 200 million images captured from camera dome sessions and HMD sessions of 256 subjects. When developing the VR facial animation methods from Chapters 2 and 3, such data was not available. We thus leveraged the variety of appearances contained in large-scale talking-head datasets to train methods that are capable of generalizing to unseen individuals during inference. However, the availability of the Ava-256 dataset (Martinez et al., 2024), containing highly task-specific data (compared to talking-head datasets), enables the development of much more powerful VR facial animation methods that generalize to unseen persons as well (Bai et al., 2024). When training our methods primarily on talking-head datasets, we could extract additional eyebrow keypoints and incorporate them into the training pipeline. However, detecting eyebrow

keypoints below the HMD and translating them to the source image space is not straightforward. Alternatively, subject-specific input-groundtruth correspondences of HMD camera images and animated 3D avatars could be generated using the Ava-256 dataset, similar to Bai et al. (2024). These can then be used to train a modified version of our method from Chapter 3 and VR-FSRT (Chapter 4), which would enable extraction of visual expression features from the eye cameras, as well as improving performance in the lower facial region. Furthermore, the generated input-groundtruth correspondences could be integrated into the standard training procedure on talking-head datasets, which contain an even larger number of identities. The model could then fuse knowledge obtained from both data sources, potentially enhancing its generalization capabilities.

The proposed FSRT model is limited to rendering output pixel colors. Nevertheless, Sajjadi et al. (2022b) demonstrated that Scene Representation Transformers are also capable of learning a neural radiance field (Mildenhall et al., 2020) to reconstruct geometry. Instead of sampling individual pixels we could predict the densities and colors of points sampled along a ray and aggregate them using volume rendering. However, suitable datasets with changing camera views are required that include information about camera intrinsics and extrinsics. As an alternative approach, 3D lifting, as proposed by Trevithick et al. (2023), could be utilized for generating a 3D volumetric representation of the driving frame, which then serves as ground truth for training FSRT or VR-FSRT. This would enable 3D training on datasets with primarily static cameras, such as VoxCeleb (Nagrani et al., 2017). Another approach that does not require any modification of our models is to simply append a 3D lifting stage to the 2D output of the methods proposed in this thesis.

A further limitation of our methods is that they are unable to generate out-of-distribution animations, such as those strongly involving the tongue or grimacing. One potential solution would be to extend the dataset with more specific data of the desired facial expressions.

The VoxCeleb2 dataset (Chung et al., 2018) is considerably larger than the VoxCeleb dataset (Nagrani et al., 2017), comprising more than six times as many videos and nearly five times as many different identities. Furthermore, the CelebV-HQ dataset (Zhu et al., 2022) contains more than twice as many identities as in VoxCeleb2, even though there are fewer videos in total. It would be insightful to investigate whether training on the VoxCeleb2 or CelebV-HQ datasets (or a combination of different datasets) could further improve the generalization ability of our models.

In this thesis we focused on performing facial animation driven by visual information. However, we did not consider the potential of incorporating audio input. One advantage of audio is that it is not really affected by an HMD that is worn. The translation from audio inputs of talking-head datasets (during training) to audio captured by a microphone on the HMD during inference is therefore a considerably more straightforward process in comparison to that of visual information. While some methods employ audio-only techniques for controlling mouth move-

ment (Ma et al., 2023; Wang et al., 2023a; Wang et al., 2021a), Agarwal et al. (2023) utilize a combination of visual and audio features. The VR facial animation method presented in Chapter 3 could be extended by incorporating audio through the encoding of a low-dimensional latent audio feature vector. This vector could then be injected into the latent space of the generator network. In our FSRT and VR-FSRT models (Chapter 4) encoded audio features could be fused with visual information in the MLP of the expression network. The latent expression vector would therefore be derived from visual and audio features.

Bibliography

Agarwal, Madhav, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar (2023). "Audio-Visual Face Reenactment." In: *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5178–5187.

Bai, Shaojie, Te-Li Wang, Chenghui Li, Akshay Venkatesh, Tomas Simon, Chen Cao, Gabriel Schwartz, Jason Saragih, Yaser Sheikh, and Shih-En Wei (2024). "Universal Facial Encoding of Codec Avatars from VR Headsets." In: *ACM Transactions on Graphics (TOG)* 43.4.

Bardes, Adrien, Jean Ponce, and Yann Lecun (2022). "VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning." In: *International Conference on Learning Representations (ICLR)*.

Blanz, Volker and Thomas Vetter (1999). "A morphable model for the synthesis of 3D faces." In: *Conference on Computer Graphics and Interactive Techniques*, pp. 187–194.

Bulat, Adrian and Georgios Tzimiropoulos (2017). "How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Cao, Chen, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, et al. (2022). "Authentic volumetric avatars from a phone scan." In: *ACM Transactions on Graphics (TOG)* 41.4, pp. 1–19.

Carlson, Alexandra, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson (2018). "Modeling camera effects to improve visual learning from synthetic data." In: *European Conference on Computer Vision (ECCV)*.

Choi, Yunjey, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo (2018). "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8789–8797.

Chung, J. S., A. Nagrani, and A. Zisserman (2018). "VoxCeleb2: Deep Speaker Recognition." In: *Conference of the International Speech Communication Association (INTERSPEECH)*.

Deng, Yu, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong (2020). "Disentangled and controllable face image generation via 3D imitative-contrastive learning." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5154–5163.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). "An Image is Worth

16x16 Words: Transformers for Image Recognition at Scale." In: *International Conference on Learning Representations (ICLR)*.

Gafni, Guy, Justus Thies, Michael Zollhofer, and Matthias Nießner (2021). "Dynamic neural radiance fields for monocular 4D facial avatar reconstruction." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8649–8658.

Gong, Yuan, Yong Zhang, Xiaodong Cun, Fei Yin, Yanbo Fan, Xuan Wang, Baoyuan Wu, and Yujiu Yang (2023). "ToonTalker: Cross-domain face reenactment." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7690–7700.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets." In: *Neural Information Processing Systems (NeurIPS)* 27.

Ha, Sungjoo, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim (2020). "MarioNETte: Few-shot face reenactment preserving identity of unseen targets." In: *AAAI Conference on Artificial Intelligence*. Vol. 34. 07, pp. 10893–10900.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Hong, Fa-Ting, Longhao Zhang, Li Shen, and Dan Xu (2022). "Depth-aware generative adversarial network for talking head video generation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3397–3406.

Hou, Xianxu, Ke Sun, Linlin Shen, and Guoping Qiu (2019). "Improving variational autoencoder with deep feature consistent and generative adversarial training." In: *Neurocomputing* 341, pp. 183–194.

Hsu, Gee-Sern, Chun-Hung Tsai, and Hung-Yi Wu (2022). "Dual-generator face reenactment." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 642–650.

Huang, Xun and Serge Belongie (2017). "Arbitrary style transfer in real-time with adaptive instance normalization." In: *International Conference on Computer Vision (ICCV)*, pp. 1501–1510.

Huang, Ziqi, Kelvin CK Chan, Yuming Jiang, and Ziwei Liu (2023). "Collaborative diffusion for multi-modal face generation and editing." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6080–6090.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros (2017). "Image-to-image translation with conditional adversarial networks." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134.

Jo, Youngjoo and Jongyoul Park (2019). "SC-FEGAN: Face editing generative adversarial network with user's sketch and color." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1745–1753.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). "Perceptual losses for real-time style transfer and super-resolution." In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 694–711.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen (2018). "Progressive Growing of GANs for Improved Quality, Stability, and Variation." In: *International Conference on Learning Representations (ICLR)*.

Lee, Cheng-Han, Ziwei Liu, Lingyun Wu, and Ping Luo (2020). "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lenz, Christian, Max Schwarz, Andre Rochow, Bastian Pätzold, Raphael Memmesheimer, Michael Schreiber, and Sven Behnke (2025). "NimbRo wins ANA Avatar XPRIZE immersive telepresence competition: human-centric evaluation and lessons learned." In: *International Journal of Social Robotics* 17.3, pp. 337–361. DOI: 10.1007/s12369-023-01050-9.

Li, Weichuang, Longhao Zhang, Dong Wang, Bin Zhao, Zhigang Wang, Mulin Chen, Bang Zhang, Zhongjian Wang, Liefeng Bo, and Xuelong Li (2023). "One-Shot High-Fidelity Talking-Head Synthesis with Deformable Neural Radiance Field." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17969–17978.

Li, Yijun, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang (2017). "Generative face completion." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3911–3919.

Lin, Xinmiao, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Yu Kong (2023). "Catch Missing Details: Image Reconstruction with Frequency Augmented Variational Autoencoder." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1736–1745.

Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (2021). "Swin transformer: Hierarchical vision transformer using shifted windows." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022.

Lombardi, Stephen, Jason Saragih, Tomas Simon, and Yaser Sheikh (2018). "Deep appearance models for face rendering." In: *ACM Transactions on Graphics (ToG)* 37.4, pp. 1–13.

Luo, Rui, Chunpeng Wang, Eric Schwarm, Colin Keil, Evelyn Mendoza, Pushyami Kaveti, Stephen Alt, Hanumant Singh, TaŞkin Padir, and John Peter Whitney (2022). "Towards Robot Avatars: Systems and Methods for Teleinteraction at Avatar XPRIZE Semi-Finals." In: *Int. Conf. on Intelligent Robots and Systems (IROS)*.

Ma, Yifeng, Suzhen Wang, Zhipeng Hu, Changjie Fan, Tangjie Lv, Yu Ding, Zhidong Deng, and Xin Yu (2023). "StyleTalk: One-shot talking head generation with controllable speaking styles." In: *AAAI Conference on Artificial Intelligence*. Vol. 37. 2, pp. 1896–1904.

Marques, Joao MC, N Patrick, Yifan Zhu, Neil Malhotra, and Kris Hauser (2022). "Commodity telepresence with the AvaTRINA Nursebot in the ANA Avatar XPRIZE semifinals." In: *RSS Workshop Towards Robot Avatars: Perspectives on the ANA Avatar XPRIZE Competition*.

Martinez, Julieta et al. (2024). "Codec Avatar Studio: Paired Human Captures for Complete, Driveable, and Generalizable Avatars." In: *NeurIPS Track on Datasets and Benchmarks*.

Meta (2024a). *Meta Quest Headset Tracking*. https://www.meta.com/help/quest/articles/headsets-and-accessories/using-your-headset/turn-off-tracking/. Accessed: 2025-02-17.

– (2024b). *Movement SDK for Unity*. https://developer.oculus.com/documentation/unity/move-overview/. Accessed: 2025-02-17.

Mildenhall, Ben, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng (2020). "NeRF: Representing scenes as neural radiance fields for view synthesis." In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 405–421.

Mollahosseini, Ali, Behzad Hasani, and Mohammad H Mahoor (2017). "Affectnet: A database for facial expression, valence, and arousal computing in the wild." In: *IEEE Transactions on Affective Computing* 10.1, pp. 18–31.

Nagrani, Arsha, Joon Son Chung, and Andrew Zisserman (2017). "VoxCeleb: A Large-Scale Speaker Identification Dataset." In: *Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2616–2620.

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). "Stacked hourglass networks for human pose estimation." In: *European Conference on Computer Vision (ECCV)*, pp. 483–499.

Pang, Youxin, Yong Zhang, Weize Quan, Yanbo Fan, Xiaodong Cun, Ying Shan, and Dong-ming Yan (2023). "DPE: Disentanglement of pose and expression for general video portrait editing." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 427–436.

Patel, Chaitanya, Shaojie Bai, Te-Li Wang, Jason Saragih, and Shih-En Wei (2024). "Fast Registration of Photorealistic Avatars for VR Facial Animation." In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 407–423.

Pätzold, Bastian, Andre Rochow, Michael Schreiber, Raphael Memmesheimer, Christian Lenz, Max Schwarz, and Sven Behnke (2023). "Audio-based roughness sensing and tactile feedback for haptic perception in telepresence." In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1387–1392. DOI: 10.1109/SMC53992.2023.10394062.

Pumarola, Albert, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer (2018). "GANimation: Anatomically-aware facial animation from a single image." In: *European Conference on Computer Vision (ECCV)*.

Richard, Alexander, Colin Lea, Shugao Ma, Jurgen Gall, Fernando De la Torre, and Yaser Sheikh (2021a). "Audio-and gaze-driven facial animation of codec avatars." In: *Winter Conference on Applications of Computer Vision (WACV)*, pp. 41–50.

Richard, Alexander, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh (2021b). "MeshTalk: 3D face animation from speech using cross-modality disentanglement." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1173–1182.

Rochow, Andre, Max Schwarz, and Sven Behnke (2023). "Attention-based VR facial animation with visual mouth camera guidance for immersive telepresence avatars." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1276–1283. DOI: 10.1109/IROS55552.2023.10342522.

– (2024). "FSRT: Facial Scene Representation Transformer for Face Reenactment from Factorized Appearance, Head-pose, and Facial Expression Features." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7716–7726. DOI: 10.1109/CVPR52733.2024.00737.

Rochow, Andre, Max Schwarz, Michael Schreiber, and Sven Behnke (2022). "VR Facial Animation for Immersive Telepresence Avatars." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2167–2174. DOI: 10.1109/IROS47612.2022.9981892.

Sajjadi, Mehdi S. M., Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetić, Mario Lučić, Leonidas J. Guibas, Klaus Greff, and Thomas Kipf (2022a). "Object Scene Representation Transformer." In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Sajjadi, Mehdi SM, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. (2022b). "Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6229–6238.

Schwartz, Gabriel, Shih-En Wei, Te-Li Wang, Stephen Lombardi, Tomas Simon, Jason Saragih, and Yaser Sheikh (2020). "The eyes have it: An integrated eye and face model for photorealistic facial animation." In: *ACM Transactions on Graphics (TOG)* 39.4.

Schwarz, Max, Christian Lenz, Raphael Memmesheimer, Bastian Pätzold, Andre Rochow, Michael Schreiber, and Sven Behnke (2023). "Robust immersive telepresence and mobile telemanipulation: NimbRo Avatar wins ANA Avatar XPRIZE Finals." In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. DOI: 10.1109/Humanoids57100.2023.10375179.

Schwarz, Max, Christian Lenz, Andre Rochow, Michael Schreiber, and Sven Behnke (2021). "NimbRo Avatar: Interactive immersive telepresence with force-feedback telemanipulation." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5312–5319.

Shen, Yujun, Jinjin Gu, Xiaoou Tang, and Bolei Zhou (2020). "Interpreting the latent space of GANs for semantic face editing." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9243–9252.

Siarohin, Aliaksandr, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe (2019a). "Animating arbitrary objects via deep motion transfer." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2377–2386.

– (2019b). "First Order Motion Model for Image Animation." In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Siarohin, Aliaksandr, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov (2021). "Motion representations for articulated animation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13653–13662.

Thies, Justus, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner (2016). "Face2Face: Real-time face capture and reenactment of RGB videos." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2387–2395.

Thies, Justus, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner (2018). "FaceVR: Real-Time Gaze-Aware Facial Reenactment in Virtual Reality." In: *ACM Transactions on Graphics (ToG)* 37, pp. 1–15.

Tran, Phong, Egor Zakharov, Long-Nhat Ho, Adilbek Karmanov, Ariana Bermudez Venegas, McLean Goldwhite, Aviral Agarwal, Liwen Hu, Anh Tran, and Hao Li (2024). "VOODOO XP: Expressive One-Shot Head Reenactment for VR Telepresence." In: *ACM Transactions on Graphics (TOG)* 43.6, pp. 1–26.

Trevithick, Alex, Matthew Chan, Michael Stengel, Eric Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano (2023). "Real-time radiance fields for single-image portrait view synthesis." In: *ACM Transactions on Graphics (TOG)* 42.4, pp. 1–15.

Unity (2024). *Unity Technologies*. https://unity.com/. Accessed: 2025-02-17.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 30.

Wang, Duomin, Yu Deng, Zixin Yin, Heung-Yeung Shum, and Baoyuan Wang (2023a). "Progressive Disentangled Representation Learning for Fine-Grained Controllable Talking Head Synthesis." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17979–17989.

Wang, Jiadong, Xinyuan Qian, Malu Zhang, Robby T Tan, and Haizhou Li (2023b). "Seeing What You Said: Talking Face Generation Guided by a Lip Reading Expert." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14653–14662.

Wang, Jiayu, Kang Zhao, Shiwei Zhang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou (2023c). "LipFormer: High-Fidelity and Generalizable Talking Face Generation With a Pre-Learned Facial Codebook." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13844–13853.

Wang, Suzhen, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu (2021a). "Audio2Head: Audio-driven One-shot Talking-head Generation with Natural Head Motion." In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Wang, Ting-Chun, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro (2019). "Few-shot Video-to-Video Synthesis." In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Wang, Ting-Chun, Arun Mallya, and Ming-Yu Liu (2021b). "One-shot free-view neural talking-head synthesis for video conferencing." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10039–10049.

Wang, Yaohui, Di Yang, Francois Bremond, and Antitza Dantcheva (2022). "Latent Image Animator: Learning to Animate Images via Latent Space Navigation." In: *International Conference on Learning Representations (ICLR)*.

Wang, Zhou (2004). "Image quality assessment: from error visibility to structural similarity." In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612.

Wei, Shih-En, Jason Saragih, Tomas Simon, Adam W Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh (2019). "VR facial animation via multiview image translation." In: *ACM Transactions on Graphics (ToG)* 38.4, pp. 1–16.

Wiles, Olivia, A Koepke, and Andrew Zisserman (2018). "X2Face: A network for controlling face generation using images, audio, and pose codes." In: *European Conference on Computer Vision (ECCV)*, pp. 670–686.

Wu, Wayne, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy (2018). "ReenactGAN: Learning to reenact faces via boundary transfer." In: *European Conference on Computer Vision (ECCV)*, pp. 603–619.

Xu, Chao, Junwei Zhu, Jiangning Zhang, Yue Han, Wenqing Chu, Ying Tai, Chengjie Wang, Zhifeng Xie, and Yong Liu (2023). "High-fidelity Generalized Emotional Talking Face Generation with Multi-modal Emotion Space Learning." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6609–6619.

Zakharov, Egor, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky (2019). "Few-shot adversarial learning of realistic neural talking head models." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9459–9468.

Zhang, Qiang et al. (2024). "REFA: Real-time Egocentric Facial Animations for Virtual Reality." In: pp. 4793–4802.

Zhang, Richard, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang (2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595.

Zhao, Jian and Hui Zhang (2022). "Thin-plate spline motion model for image animation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3657–3666.

Zhao, Ruiqi, Tianyi Wu, and Guodong Guo (2021a). "Sparse to dense motion transfer for face image animation." In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1991–2000.

Zhao, Zengqun, Qingshan Liu, and Feng Zhou (2021b). "Robust Lightweight Facial Expression Recognition Network with Label Distribution Training." In: *AAAI Conference on Artificial Intelligence*. Vol. 35. 4, pp. 3510–3519.

Zhou, Hang, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang (2019). "Talking face generation by adversarially disentangled audio-visual representation." In: *AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 9299–9306.

Zhou, Hang, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu (2021). "Pose-controllable talking face generation by implicitly modularized audio-visual representation." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4176–4186.

Zhou, Tong, Changxing Ding, Shaowen Lin, Xinchao Wang, and Dacheng Tao (2020). "Learning oracle attention for high-fidelity face completion." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7680–7689.

Zhu, Hao, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy (2022). "CelebV-HQ: A large-scale video facial attributes dataset." In: *European Conference on Computer Vision (ECCV)*, pp. 650–667.

A

Incorporated Publications

A.1 VR Facial Animation for Immersive Telepresence Avatars

This publications is the basis for Chapter 2.

The publication has been removed from the online version of the thesis and can be accessed via the DOI link above.

A.2 Attention-based VR facial animation with visual mouth camera guidance for immersive telepresence avatars

This publications is the basis for Chapter 3.

The publication has been removed from the online version of the thesis and can be accessed via the DOI link above.

A.3 FSRT: Facial Scene Representation Transformer for Face Reenactment from Factorized Appearance, Head-pose, and Facial Expression Features

This publication, in conjunction with the corresponding the Supplementary Material, is the basis for the main part of Chapter 4.

In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7716–7726. DOI: 10.1109/CVPR52733.2024.00737.

The publication has been removed from the online version of the thesis and can be accessed via the DOI link above.