Rheinland-Pfälzische
Technische Universität
Kaiserslautern
Landau

Optimization and Generative Models for Face Analysis

Dissertation

Thesis approved by
the Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Jilliam María Díaz Barros

Date of Defense:	December 19th, 2024
Dean:	Prof. Dr. Christoph Garth
Reviewer:	Prof. Dr. Didier Stricker
Reviewer:	Prof. Dr. Luís Gonzaga Mendes Magalhães

DE-386

Abstract

Human analysis, with an emphasis on the head and face, has been an important subject of study in a wide range of scientific fields. The facial expressions, eye gaze, gestures and head pose provide non-verbal cues about the physical and mental state of individuals, their emotions, consciousness and attentiveness. In computer vision, these cues have been leveraged to study face images, where fields such as emotion recognition, face identification and gaze estimation have emerged as cornerstones in human modeling and understanding.

Research on face analysis has enabled the development of assistance tools, for instance, to identify the fatigue and inattention of a vehicle driver, detect the level of asymmetry in patients with facial paralysis or recognize the emotions of children with autism. Such tools demand a high level of robustness in a wide range of situations, such as varying illumination, large head rotations and partial occlusion, in addition to being able to operate with low computational costs and with real-time capabilities.

This thesis focuses on the analysis of the rigid and non-rigid motions of the head, with the aim of targeting existing challenges in assistance tools and assistive technologies. The proposed approaches are based on intensity and RGB images, and cover the estimation of the head pose and the detection and tracking of facial landmarks. Additionally, facial expression transfer for avatar animation and the generation of sign language images are explored, where the face and upper body play a crucial part in conveying information.

The main contributions on this thesis are grouped in four categories: *(i)* the study of head pose estimation from a monocular system, to obtain a robust estimate of the rigid head pose in real time. To that end, multiple architectures are proposed, where the head pose is formulated as an optimization problem based on the 3D-2D correspondences of facial features; *(ii)* the detection and tracking of fiducial facial landmarks, to model the non-rigid facial motion. The sparse set of facial landmarks are computed using a deep-learning-based generative pipeline, extending the detection to a wide range of facial expressions, including faces with substantial asymmetry. Tracking of 3D landmarks is also investigated, where a method to exploit the landmark connectivity and confidence score is introduced; *(iii)* the development of a real-time framework for performance-driven facial animation using a monocular system. This method aims to transfer the rigid head pose and facial expression from 2D video sequences to a 3D head model with limited resources; and *(iv)* a rendering pipeline to extend current RGB-based sign language datasets, to include multiple types of annotated data such as segmentation masks, normal and depth maps, and 3D-2D body joints. A sign language image synthesis architecture based on generative models is also investigated, conditioned on pose and appearance. Our architecture is trained and evaluated on synthetic and real data.

The proposed methods achieve outstanding performance compared to related work on publicly available benchmarks, as well as in datasets introduced in this thesis. Furthermore, the presented approaches for head pose estimation and performance-driven facial animation perform in real time on CPU, while the proposed pipelines for face alignment and sign language image synthesis achieve real-time capabilities on GPU.

Kurzfassung

Die Analyse des Menschen, insbesondere des Kopfes und des Gesichts, ist ein wichtiger Untersuchungsgegenstand in einer Vielzahl von wissenschaftlichen Bereichen. Mimik, Blick, Gestik und Kopfhaltung liefern nonverbale Hinweise auf den körperlichen und geistigen Zustand von Personen, ihre Emotionen, ihr Bewusstsein und ihre Aufmerksamkeit. In der Computer Vision wurden diese Hinweise genutzt, um Gesichtsbilder zu untersuchen, wobei sich Bereiche wie die Erkennung von Emotionen, die Identifizierung von Gesichtern und die Schätzung von Blicken als Eckpfeiler für die Modellierung und das Verständnis des Menschen herausgebildet haben.

Die Forschung auf dem Gebiet der Gesichtsanalyse hat die Entwicklung von Hilfsmitteln ermöglicht, die zum Beispiel die Müdigkeit und Unaufmerksamkeit eines Autofahrers erkennen, den Grad der Asymmetrie bei Patienten mit Gesichtslähmung feststellen oder die Emotionen von Kindern mit Autismus erkennen können. Solche Hilfsmittel müssen in einer Vielzahl von Situationen, z. B. bei wechselnder Beleuchtung, starken Kopfdrehungen und teilweiser Verdeckung, sehr robust sein und mit geringem Rechenaufwand und in Echtzeit arbeiten können.

Diese Arbeit konzentriert sich auf die Analyse der starren und nicht-starren Bewegungen des Kopfes, mit dem Ziel, die bestehenden Herausforderungen in Assistenzwerkzeugen und assistiven Technologien zu bewältigen. Die vorgeschlagenen Ansätze basieren auf Intensitäts- und RGB-Bildern und umfassen die Schätzung der Kopfhaltung sowie die Erkennung und Verfolgung von Gesichtsmerkmalen. Darüber hinaus werden die Übertragung von Gesichtsausdrücken für die Animation von Avataren und die Erzeugung von Gebärdensprachbildern untersucht, bei denen das Gesicht und der Oberkörper eine entscheidende Rolle bei der Übermittlung von Informationen spielen.

Die wichtigsten Beiträge zu dieser Arbeit sind in vier Kategorien unterteilt: *(i)* die Untersuchung der Kopfposenschätzung aus einem monokularen System, um eine robuste Schätzung der starren Kopfposition in Echtzeit zu erhalten. Zu diesem Zweck werden mehrere Architekturen vorgeschlagen, bei denen die Kopfhaltung als Optimierungsproblem auf der Grundlage der 3D-2D-Korrespondenzen von Gesichtsmerkmalen formuliert wird; *(ii)* die Erkennung und Verfolgung von Referenzpunkten im Gesicht, um die nicht-starre Gesichtsbewegung zu modellieren. Der spärliche Satz von Gesichtsmerkmalen wird mit Hilfe einer Deep-Learning-basierten generativen Pipeline berechnet, wodurch die Erkennung auf ein breites Spektrum von Gesichtsausdrücken, einschließlich Gesichtern mit erheblicher Asymmetrie, ausgeweitet wird. Die Verfolgung von 3D-Landmarken wird ebenfalls untersucht, wobei eine Methode zur Ausnutzung der Konnektivität der Landmarken und des Konfidenzwerts eingeführt wird; *(iii)* die Entwicklung eines Echtzeit-Frameworks für die darstellungsgetriebene Gesichtsanimation unter Verwendung eines monokularen Systems. Diese Methode zielt darauf ab, die starre Kopfhaltung und den Gesichtsausdruck von 2D-Videosequenzen auf ein 3D-Kopfmodell mit begrenzten Ressourcen zu übertragen; und *(iv)* eine Rendering-Pipeline zur Erweiterung aktueller RGB-basierter Gebärdensprachdatensätze, um mehrere Arten von annotierten Daten wie Segmentierungsmasken, Normalen- und Tiefenkarten und 3D-2D-Körpergelenke einzubeziehen. Außerdem wird eine Architektur für die Bildsynthese von Gebärdensprachen auf der Grundlage generativer Modelle untersucht, die auf Pose und Aussehen basieren. Unsere Architektur wird auf synthetischen und realen Daten trainiert und evaluiert.

Die vorgeschlagenen Methoden erreichen eine hervorragende Leistung im Vergleich zu verwandten Arbeiten bei öffentlich verfügbaren Benchmarks sowie bei den in dieser Arbeit vorgestellten Datensätzen. Darüber hinaus arbeiten die vorgestellten Ansätze für die Kopfposenschätzung und die darstellungsgetriebene Gesichtsanimation in Echtzeit auf der CPU, während die vorgeschlagenen Pipelines für die Gesichtsausrichtung und die Gebärdensprachbildsynthese Echtzeitfähigkeit auf der GPU erreichen.

Acknowledgments

Before falling short in acknowledgments, I would like to thank all my family members, friends, classmates, colleagues and teachers that have been part of my academic journey from an early age. I consider myself really lucky to have found along the way such great teachers, who have shaped my analysis and problem-solving skills and ignited my curiosity.

For my PhD journey, I am forever grateful to Prof. Didier Stricker, who gave me the opportunity to be part of the Augmented Vision family at DFKI. The AV group does not only provide a great research environment where academia meets industry, but is also a place with top researchers, where I have found long-lasting friendships. During my journey as a PhD student and researcher, I have had the chance to learn from my brilliant supervisors and colleagues, strengthen my technical and academic abilities and push my experience and knowledge beyond my own boundaries.

I also extend my gratitude to my friends, colleagues and co-authors at the AV department that have made this path more enjoyable, filled with fantastic memories and great collaboration outcomes. My most sincere thanks to Dr. Gerd Reis, with whom I have been working in the last years and from whom I have learned a lot. Many thanks to David Haase, who has been really helpful in the most critical times when my computer seemed overwhelmed by loads of work.

I would also like to thank the members of the PhD committee, Prof. Dr. Annette Bieniusa and Prof. Dr. Luís Gonzaga Magalhães, for the very interesting and insightful discussion during my PhD defense, along with Prof. Dr. Didier Stricker, and to Dr. Bernd Schürmann and Ms. Sabine Owens for the administrative support during the PhD course.

My warmest thanks to Mahdi, Mohamed, Moni, Elisabeth, Stephan, Sai and Simon, for taking care along with Jason in the preparation of my PhD cart, hat and presentation. They were the cherry on top of my PhD journey. Many thanks to Jason and Mohamed for the excellent feedback to my presentation and to all my friends and colleagues who also joined me in the celebration of this milestone. It was really comforting to have you all on that day.

A mis queridos padres, Faride y Guillermo, les doy millones de gracias por ser unos padres maravillosos. Desde que tengo uso de razón, me han guiado en cada paso, me han incentivado a dar lo mejor de mí y siempre me han alentado a seguir mis sueños. Además, me enseñaron que las metas se logran con disciplina y dedicación. También les agradezco a mis hermanos, que han sido mi gran compañía y parte fundamental de mi vida, y que me dan ánimos para seguir adelante. Muchas gracias a todos mis familiares y amigos en Colombia, que a pesar de la distancia, me siguen brindando cariño y apoyo. ¡Los quiero mucho!

Finally, my deepest appreciation to my wonderful husband Jason, who has been my biggest supporter and most fantastic adventure partner; and to our beloved baby, who accompanied me during my thesis defense and has given me the final push and extra strength to finish successfully this journey. I love you both with all my heart.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Problem Formulation and Approach . . . 3
1.3 Contributions . . . 4
1.4 Organization of the Thesis . . . 7
1.5 Publications and Patent . . . 8

2 Background **9**
2.1 Computer Vision . . . 10
2.1.1 Camera Model . . . 10
2.1.2 Camera Calibration . . . 12
2.1.3 Triangulation . . . 13
2.1.4 Feature Detection and Tracking . . . 14
2.1.5 Perspective-n-Point Problem . . . 16
2.1.6 Distance Transform . . . 17
2.2 Statistical Filtering . . . 17
2.3 Deep Learning Models . . . 19
2.3.1 Convolutional Networks Architectures . . . 19
2.3.2 Transformers . . . 21
2.3.3 Generative Models . . . 26
2.4 Face Analysis Tasks . . . 29
2.4.1 Face Detection . . . 29
2.4.2 Face Alignment . . . 30
2.5 Evaluation Metrics . . . 32

3 Head Pose Estimation **35**
3.1 Introduction . . . 36
3.2 Problem Formulation . . . 38
3.3 Related Work . . . 38
3.4 Head Pose Estimation from Keypoints . . . 42
3.4.1 Overview of the Proposed HPE Framework . . . 43
3.4.2 System Initialization . . . 43
3.4.3 Robust 2D Keypoint Detection and Matching . . . 45
3.4.4 3D Keypoint Computation . . . 46
3.4.5 Pose Estimation from Feature Points . . . 47
3.4.6 ROI for 2D Keypoint Detection . . . 47
3.5 Head Pose Estimation by Measurement Fusion from Facial Landmark Detection and Keypoint Tracking . . . 49
3.5.1 Overview of the Proposed HPE Framework . . . 49

3.5.2 Facial Landmark Detection . . . 49
3.5.3 Keypoint Detection and Tracking . . . 51
3.5.4 System Initialization . . . 51
3.5.5 Sparse 3D Reconstruction . . . 52
3.5.6 HPE and Measurement Fusion . . . 52
3.6 Head Pose Estimation by Fusion of Global and Local Motion . . . 54
3.6.1 Overview of the Proposed HPE pipeline . . . 54
3.6.2 Computation of 2D and 3D Keypoints . . . 55
3.6.3 Computation of Facial Landmarks . . . 55
3.6.4 Initialization . . . 57
3.6.5 Head Pose Estimation . . . 57
3.7 Experiments and Results . . . 59
3.7.1 Datasets . . . 59
3.7.2 Evaluation Metrics . . . 60
3.7.3 Results . . . 60
3.7.4 Ablation Study . . . 63
3.7.5 Runtime Analysis . . . 67
3.7.6 Discussion . . . 67
3.8 Concluding remarks . . . 69

4 Facial Landmark Detection and Tracking 71
4.1 Introduction . . . 72
4.2 Related Work . . . 74
4.2.1 2D Face Alignment . . . 74
4.2.2 3D Face Alignment . . . 75
4.2.3 Face Alignment for Palsy . . . 76
4.2.4 3D Landmark Tracking . . . 76
4.3 Face Alignment on Symmetrical and Asymmetrical Expressions . . . 77
4.3.1 Overview of the Proposed Face Alignment Approach . . . 77
4.3.2 Self-Supervised Stage . . . 78
4.3.3 In-Domain Inversion . . . 79
4.3.4 Supervised Stage . . . 80
4.3.5 Datasets . . . 81
4.3.6 Implementation Details . . . 82
4.3.7 Evaluation Metrics . . . 83
4.3.8 Results on 3DA-2D and 3D Face Alignment . . . 83
4.3.9 Evaluation on ParFace . . . 85
4.3.10 Runtime and Model Parameters . . . 88
4.3.11 Ablation Study . . . 88
4.3.12 Results on 2D Face Alignment . . . 91
4.3.13 Discussion and Limitations . . . 94

4.4 Facial Landmark Tracking . . . 94
4.4.1 Overview of the Proposed Landmark Tracking Pipeline . . . 95
4.4.2 Noise Adaptive Clustered Landmark Kalman Filters . . . 95
4.4.3 Confidence Adaptive Kalman Filter . . . 96
4.4.4 Activation of Face Detector . . . 97
4.4.5 Datasets . . . 97
4.4.6 Evaluation Metrics . . . 98
4.4.7 Ablation Study . . . 98
4.4.8 Results on 3D Face Tracking . . . 100
4.4.9 Discussion . . . 102
4.5 Conclusions . . . 103

5 Performance-Driven Facial Animation 105
5.1 Introduction . . . 106
5.2 Related work . . . 107
5.3 Overview of the Proposed Pipeline . . . 108
5.3.1 Blendshape Representation . . . 109
5.3.2 Registration of Blendshape Targets . . . 109
5.4 Target Energy Functional . . . 110
5.4.1 Non-Rigid Tracking . . . 111
5.4.2 Rigid Head Pose Estimation . . . 111
5.4.3 2D-3D Transfer of Facial Expressions . . . 111
5.4.4 Energy Minimization . . . 112
5.5 Experiments and Results . . . 112
5.5.1 Datasets . . . 113
5.5.2 Runtime Analysis . . . 113
5.5.3 Parameter Tuning . . . 115
5.5.4 Qualitative Results . . . 117
5.5.5 Head Pose Evaluation . . . 117
5.5.6 Discussion . . . 117
5.6 Conclusions . . . 117

6 Sign Language Production 119
6.1 Introduction . . . 120
6.2 Related work . . . 122
6.2.1 Sign Language Recognition and Translation . . . 122
6.2.2 Sign Language Production . . . 122
6.2.3 Sign Language Datasets . . . 123
6.2.4 Synthetic Datasets for Human Analysis . . . 124
6.3 SynthSL and Rendering Pipeline . . . 125
6.4 Sign Language Image Synthesis . . . 128
6.4.1 Generator and Discriminator Networks . . . 128

6.4.2 Generator Network . . . 129
6.5 Experiments and Results . . . 130
6.5.1 Datasets . . . 131
6.5.2 Data pre-processing and model architecture . . . 131
6.5.3 Baseline architectures . . . 131
6.5.4 Metrics . . . 132
6.5.5 Evaluation . . . 132
6.5.6 Ablation Study . . . 134
6.5.7 Runtime Analysis . . . 136
6.5.8 Limitations and failure cases . . . 136
6.5.9 Discussion . . . 136
6.6 Conclusions . . . 137

7 Summary and Conclusion **139**
7.1 Summary . . . 139
7.2 Concluding Remarks . . . 140
7.3 Future Work . . . 141

Bibliography **143**

List of Abbreviations **171**

List of Tables **175**

List of Figures **177**

1 Introduction

The human face has been the center of research in multiple fields of science, as it conveys crucial information from individuals, such as identity and emotions. Furthermore, face perception has played a key role in evolution, as it molds how humans interact with each other [1, 2]. The rigid motion of the head can be used to determine the focus of attention and engagement in a given task, while facial expressions provide non-verbal insight about the emotional state and intentions. In computer vision, the recognition of these visual cues has been explored in face analysis, with multiple applications in education [3], entertainment [4], healthcare [5], marketing [6], biometrics [7], surveillance [8], among others.

In the entertainment industry, face analysis has been leveraged in the development of interactive user interfaces or expression transfer pipelines to design and animate characters in movies, video games, social media or advertising booths. More recently, the use of generative models have broadened the spectrum of applications for entertainment, which have garnered praise as well as concerns, particularly over privacy matters. Related concerns apply for technologies in biometrics and surveillance [7].

Nevertheless, research on face and body analysis has led to the development of assistance tools and assistive technologies with undoubted benefit to humanity. These technologies encompass healthcare applications, *e.g.* to track the recovery of stroke patients at home [9] or in emotion recognition of patients with mental or neurological disorders, such as depression or autism [10, 11]. Assistance tools are also advantageous in everyday activities such as driving and teaching, or targeted to improve the quality of life of communities with limitations, *e.g.*, by enhancing communication in people with hearing impairment [12].

Many assistance tools for face analysis demand low computational costs and resources, besides robustness in a wide range of settings. That is the case, for instance, in tools for patients in rehabilitation programs at home, where the recovery can be tracked using smartphones or web-cameras. In addition, research in face analysis is confronted with multiple challenges due to the high variability in term of pose, appearance and facial expression, as shown in Table 1.1. In data collected in the wild, they are affected by external factors such as illumination, occlusion, blurriness, low quality and make up.

1.1 Motivation

Despite significant advances in face analysis research, many existing pipelines face several limitations when deployed in real world applications. One example is the detection of the engagement level of a user, when performing a particular task. In these scenarios, many systems are not able to track the head in challenging instances in real time, *e.g.*, under varying light, partial occlusion, or large head rotations [3]. Moreover, in applications where the head pose is computed as an upstream task, the method should be able to deliver a robust performance with restricted resources, to enable the integration of more time-consuming tasks.

Another limitation in current works is their reduced capacity to handle cases that are con-

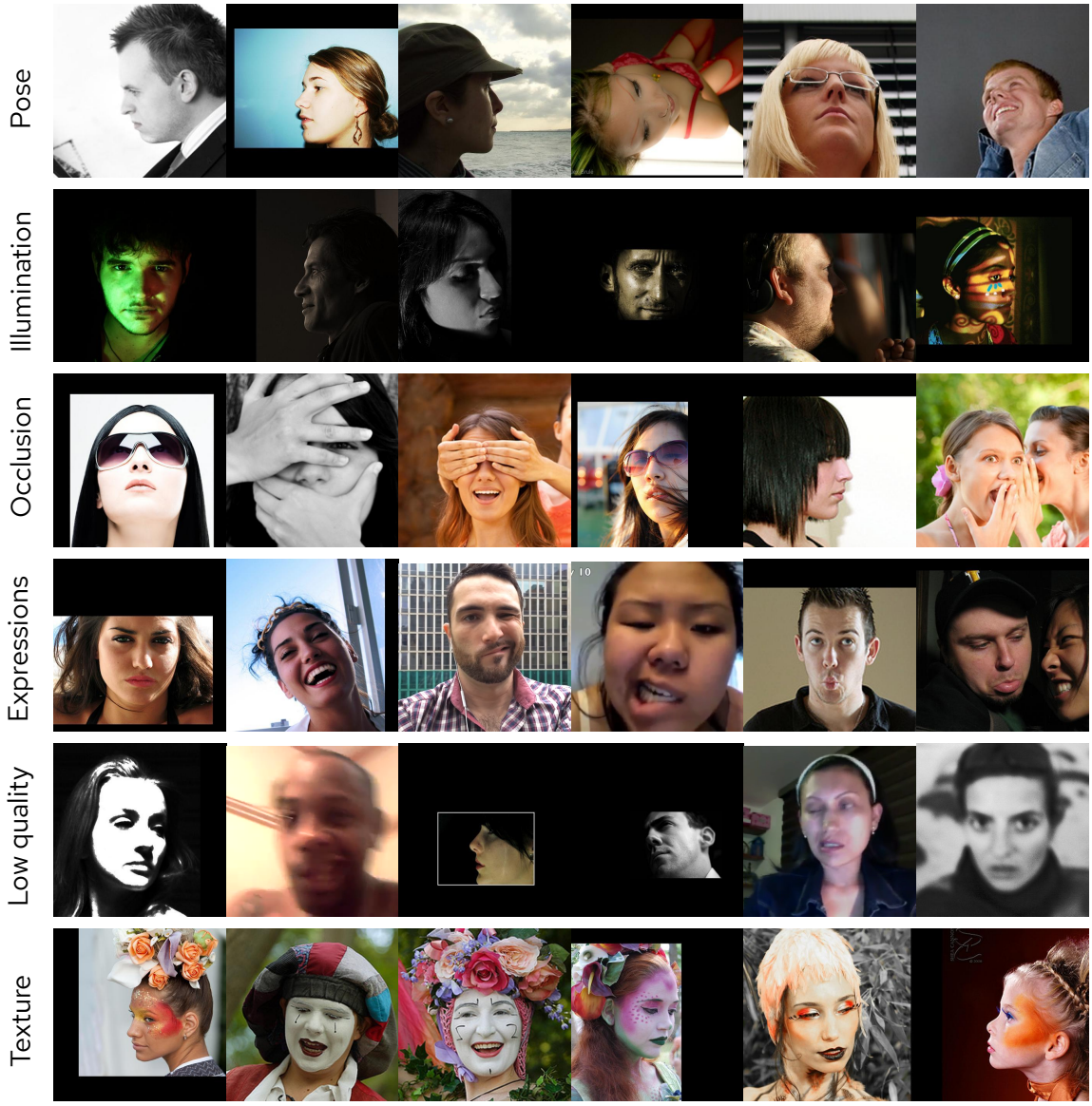


Table 1.1: Challenging images from AFLW2000-3D [13] and ParFace.

sidered outliers or are not covered in the training data. That is the case for unusual facial expressions with large asymmetry, *e.g.*, in patients with unilateral facial paralysis. Most pipelines are trained on data from healthy individuals, limiting the development of mechanism to automatically evaluate the recovery rate of patients.

Tools to improve communication are also part of assistive technologies, as communication plays an important role in human interaction. Instruments for lip reading, avatar animation, and sign language recognition and production are representative cases of such technologies.

The animation of avatars which mimic the facial expressions of users has gained popularity in recent years, with the rise of virtual and mixed reality. These technologies require robust pipelines, which are able to perform in real time and with limited resources, *e.g.*, based on monocular images from web-cameras.

In sign language, the automatic generation of images has been scarcely researched. Related works propose generative models that usually require training data from private datasets, to be able to generate high quality sign images. In addition, sign language is not universal and sign language recognition pipelines can only be deployed with data from the intended language. Therefore, a mechanism to extend current sign language datasets is of great relevance.

In this thesis, we aim to target previous limitations and contribute to bridge the gap in several fields within face analysis research, to attain the successful integration into assistance tools and assistive technologies.

1.2 Problem Formulation and Approach

In face analysis, we have identified the following limitations in existing works when applied in real scenarios:

- *The continuous estimation and tracking of the head pose in real time, in tasks that evaluate user engagement.* Many related works suffer due to sensitivity to illumination changes, large head poses and partial occlusion. In addition, their need of specialized hardware with increased costs, specific user-oriented calibration setups or restrictions to initialize, *e.g.*, using only frontal poses, deter their applicability and usability in real scenarios with limited resources. Therefore, one of our main focus is the estimation of the head pose using a robust approach that can be integrated into consumer devices. Herein, we propose three head pose estimation methods that leverage the pose retrieved via optimization from 3D-2D correspondences of facial features. The presented approaches can handle large head rotations and varying illumination while estimating the pose in real time.

- *Limited capabilities to handle a wide range of facial expressions.* The great majority of face alignment approaches use supervised approaches for the estimation of facial landmarks. Consequently, they have limited performance under unseen facial expressions, *e.g.*, with large asymmetry. In this thesis, we propose a semi-supervised method based on a generative architecture that learns a face representation in the latent space using unlabeled data. The latent code is later refined via in-domain inversion, to improve the landmark detection on a wide range of expressions.

- *Restricted performance of deep face alignment architectures in video sequences.* Facial landmarks are usually detected on a frame basis, with a reduced performance in frames with large occlusion, blurriness or varying illumination. We propose to integrate landmark tracking in existing deep learning frameworks, and exploit the confidence score retrieved for every landmark, to improve the detection in challenging frames.

- *The animation of virtual characters in real time with limited resources.* Pose and expression transfer to animate a character is a time consuming task, usually requiring large datasets for training or specialized hardware such as GPU. We explore performance-driven facial animation in real time on a lightweight optimization-based pipeline, that is able to run on CPU using a consumer camera.

- *The development of accessible pipelines for high-quality sign language image synthesis.* Current works in sign language image synthesis need to be trained on large datasets, many of which are not publicly available. Additionally, to improve the synthesis of hands and faces, the models need dedicated pipelines with manually curated datasets. We propose a rendering pipeline that extends current image-based datasets with multiple types of annotation data. Moreover, we introduce a synthetic dataset in sign language, which provides several advantages such as signer variability, sharp hands and varied annotations. In addition, a pipeline for sign language synthesis based on generative networks is proposed, conditioned on given body poses and appearance.

1.3 Contributions

The primary contributions of this thesis are classified in four categories, as described below.

Head pose estimation

We first investigate head pose estimation under real-time constraints with limited resources. The main contributions in this topic correspond to:

(i) A real-time pipeline for head pose estimation in video sequences, which is able to handle large head rotations and overrides the need of face detection in every frame. The pose is computed from 2D keypoints tracked with optical flow and their reconstruction on a 3D geometric model.

(ii) A framework that uses facial landmarks detected in every frame along with tracked keypoints to estimate the pose. The pose from each set of features are combined through measurement fusion in a dedicated Kalman filter. We additionally introduce a new method to compute an initial pose in cases where the face is not frontal.

(iii) A novel approach for head pose estimation, which exploits the local head motion estimated from keypoint tracking along with the global motion from facial landmark detection. The landmarks are hypothesized in the 3D space using triangulation and refined by means of a Kalman filter.

The proposed frameworks are evaluated on a publicly benchmark on head pose estimation and compared to related works. These contributions are covered in Chapter 3 and are based on the work published in:

- Jilliam M. Díaz Barros, F. Garcia, B. Mirbach and D. Stricker, "Real-Time Monocular 6-DOF Head Pose Estimation From Salient 2D Points", in IEEE International Conference on Image Processing (ICIP), Beijing, China, pp. 121-125. 2017.
- Jilliam M. Díaz Barros, F. Garcia, B. Mirbach, K. Varanasi, and D. Stricker. "Combined Framework for Real-time Head Pose Estimation using Facial Landmark Detection and Salient Feature Tracking", in International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Madeira, Portugal, pp. 123-133. 2018.

- Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker. "Fusion of Keypoint Tracking and Facial Landmark Detection for Real-Time Head Pose Estimation", in IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, pp. 2028-2037. 2018.
- Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, D. Stricker, "Real-Time Head Pose Estimation by Tracking and Detection of Keypoints and Facial Landmarks", in Computer Vision, Imaging and Computer Graphics Theory and Applications. VISIGRAPP. Communications in Computer and Information Science (CCIS), vol 997. Springer, Cham. 2018.

Facial landmark detection and tracking

We additionally study the detection and tracking of facial landmarks, with the following contributions:

(i) A generative approach for facial landmark detection on a wide range of expressions, including asymmetrical expressions due to unilateral facial paralysis. Our method integrates in-domain inversion to an encoder-decoder architecture, to refine the embedding space and improve the detection of the landmarks.

(ii) A newly proposed dataset and benchmark on face alignment for patients with facial paralysis, ParFace. The dataset contains videos of patients in the wild, performing different facial expressions in varying settings.

(iii) A confidence-aware framework for landmark tracking in sequences, based on Kalman Filter. Our framework enables tracking in single-frame deep-learning-based pipelines, leveraging the connectivity and confidence score of every landmark, to tackle occlusion and other visual artifacts in videos, such as blurriness.

The robustness of the proposed pipelines are evaluated on publicly available datasets, in addition to ParFace. These approaches are described in Chapter 4 and they are based on the following work:

- Jilliam M. Díaz Barros, J. Rambach, P. Murthy and D. Stricker, "In-Domain Inversion for Improved 3D Face Alignment on Asymmetrical Expressions", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024.
- Jilliam M. Díaz Barros, C-Y. Wang, D. Stricker and J. Rambach, "Confidence-Aware Clustered Landmark Filtering for Hybrid 3D Face Tracking", in IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia. 2023.

Performance-driven facial animation

We explore expression and pose transfer to animate an avatar in real time, using a consumer camera. The framework is detailed in Chapter 5, with the main contribution being:

(i) A new real-time framework for performance-driven facial animation from a monocular setup. We formulate 3D character animation as a lightweight energy-based optimization problem solved with non-linear least-squares, where the head is modeled using expressive blendshapes.

This Chapter is based on the work presented in:

- Jilliam M. Díaz Barros, V. Golyanik, K. Varanasi and D. Stricker, "Face It!: A Pipeline for Real-Time Performance-Driven Facial Animation", in IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 2209-2213. 2019.

Sign language production

Finally, we investigate the synthesis of images for sign language production. The following contributions are covered in detail in Chapter 6:

(i) A novel generator based on transformers to synthesize sign language images with realistic facial expressions and hand poses. The synthesizing method is based on a generative framework that takes as input a target and source pose, plus a base signer with a defined style, and outputs the base signer in the target pose.

(ii) A new pipeline for rendering body images, which integrates the shape and pose of the body, hands and face, in addition to facial expressions. Our rendering pipeline enables the generation of multiple annotation data, such as RGB images, segmentation masks, depth and normal maps, and 3D-2D body joints.

(iii) SynthSL, a large-scale synthetic dataset for research on sign language. Our dataset expands the *PHOENIX 2014T* dataset [14] with ground truth on rendered models, segmentation masks, depth and normal maps, 2D/3D body joints, body pose and shape, which are provided on a frame basis.

We evaluate the pipeline on publicly available benchmarks and on our own dataset, SynthSL. The results of this Chapter are based on the work in:

- Jilliam M. Díaz Barros, C-Y. Wang, J. Malik, A. Arafa and D. Stricker, "SynthSL: Expressive Humans for Sign Language Image Synthesis", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024.

1.4 Organization of the Thesis

The content of this thesis is divided in seven chapters, with the following structure:

1. Chapter 2 provides an introduction to relevant concepts and models employed in subsequent chapters. Kalman filter and computer vision methods such as camera calibration and triangulation, feature detection and pose estimation are described. Related face analysis tasks, generative deep learning architectures and the evaluation metrics used in this thesis are covered as well.

2. Chapter 3 introduces the proposed approaches for head pose estimation, based on the detection and tracking of facial features. The evaluation on a common public benchmark, with multiple ablation studies and a runtime analysis of every framework are also presented.

3. Chapter 4 describes our deep-learning-based architecture for the detection of facial landmarks in a wide range of facial expressions. In addition, the pipeline to integrate landmark tracking in single-frame face-alignment frameworks is presented. Facial landmark detection and tracking are evaluated in well-known datasets, in addition to our newly introduced dataset, ParFace.

4. Chapter 5 presents our optimization-based method for performance-driven facial animation. This approach aims at transferring the head pose and facial expression to an avatar in real time. Quantitative and qualitative results are presented, in addition to a runtime analysis.

5. Chapter 6 introduces a generative pipeline for sign language image synthesis, given a pose representation. Our synthetic dataset for sign language, SynthSL, and the rendering pipeline are also described. This dataset includes multiple types of ground truth data such as RGB images, segmentation masks, depth and normal maps, and 2D and 3D joints.

6. Chapter 7 summarizes the work presented in the thesis, the conclusions and a discussion of future work.

1.5 Publications and Patent

This thesis collects the work presented in peer-reviewed conferences and journals. The publications covered in this document correspond to:

1. Jilliam M. Díaz Barros, F. Garcia, B. Mirbach and D. Stricker, "Real-Time Monocular 6-DOF Head Pose Estimation From Salient 2D Points", in IEEE International Conference on Image Processing (ICIP), Beijing, China, pp. 121-125. 2017, [15].

2. Jilliam M. Díaz Barros, F. Garcia, B. Mirbach, K. Varanasi, and D. Stricker. "Combined Framework for Real-time Head Pose Estimation using Facial Landmark Detection and Salient Feature Tracking", in International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Madeira, Portugal, pp. 123-133. 2018, [16].

3. Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker. "Fusion of Keypoint Tracking and Facial Landmark Detection for Real-Time Head Pose Estimation", in IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, pp. 2028-2037. 2018, [17].

4. Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, D. Stricker, "Real-Time Head Pose Estimation by Tracking and Detection of Keypoints and Facial Landmarks", in Computer Vision, Imaging and Computer Graphics Theory and Applications. VISIGRAPP. Communications in Computer and Information Science (CCIS), vol 997. Springer, Cham. 2018, [18].

5. Jilliam M. Díaz Barros, V. Golyanik, K. Varanasi and D. Stricker, "Face It!: A Pipeline for Real-Time Performance-Driven Facial Animation", in IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 2209-2213. 2019, [19].

6. Jilliam M. Díaz Barros, P. Murthy and D. Stricker, "In-Domain Inversion for Improved 3D Face Alignment on Asymmetrical Expressions", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024, [20].

7. Jilliam M. Díaz Barros, C-Y. Wang, D. Stricker and J. Rambach, "Confidence-Aware Clustered Landmark Filtering for Hybrid 3D Face Tracking", in IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia. 2023, [21].

8. Jilliam M. Díaz Barros, C-Y. Wang, J. Malik and D. Stricker, "SynthSL: Expressive Humans for Sign Language Image Synthesis", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024, [22].

Patent

1. B. Mirbach, F. Garcia Becerro, Jilliam M. Díaz Barros. Method and system for head pose estimation. US Patent No. 16632689, 2021, [23].

2 Background

Contents

2.1 Computer Vision 10
2.2 Statistical Filtering 17
2.3 Deep Learning Models 19
2.4 Face Analysis Tasks 29
2.5 Evaluation Metrics 32

Description

In this Chapter, the main relevant concepts and techniques used throughout the thesis are summarized. This Chapter is divided in five main sections:

1. Computer vision. This section provides the theoretical foundation for the proposed approaches for head pose estimation and performance-driven facial animation in Chapters 3 and 5. The camera model, camera calibration, triangulation, feature detection and tracking, and the Perspective-n-Point problem are formally defined.

2. Statistical filtering. The linear Kalman Filter is covered in this section, where the general equations are introduced. Kalman Filter is leveraged for head pose estimation and facial landmark tracking, in Chapters 3 and 4, respectively.

3. Deep learning models. This section introduces convolutional neural network architectures such as U-Net and ResNet, and generative models for image synthesis, relevant in Chapters 4 and 6. The Swin Transformer architecture used in Chapter 6 is also described.

4. Face analysis tasks. This section presents the definition of face-related techniques such as face detection and face alignment, where relevant computer vision and deep-learning-based approaches are introduced. These techniques are upstream tasks for the proposed approaches in Chapters 3 to 6.

5. Evaluation metrics. This last section includes the metrics to evaluate the proposed methods for head pose estimation, face alignment and image synthesizing. These methods are covered in Chapters 3 to 6.

2.1 Computer Vision

2.1.1 Camera Model

Camera models describe the relationship between the world and camera coordinate systems using different projective models, based on orthographic, affine or perspective transformations. In our pipelines for head pose estimation and performance-driven facial animation, we assume the projection is given by the pinhole camera model.

The pinhole camera model defines the mapping from 3D points in the wold coordinate system **W** to the 2D image plane, **I**, using a perspective transformation. The pinhole camera captures the light from the 3D world through an aperture, or pinhole, and projects it into an inverted image **I** in the camera obscura.

The image generated by the camera is parallel to the plane given by the **X** and **Y** axes of the camera center **C** (see Figure 2.1). The focal length f is the distance from **C** to the principal point **c** in the image plane **I**. The line passing from **C** through **c** is termed as the optical or principal axis. For practical purposes, a virtual image plane **I**′ is defined, which inverts the image **I** 180 degrees and places it in front of **C**. This virtual image plane has no actual meaning in the physical world, but facilitates the understanding of the projection model. **I**′ is parallel to **I** and is located at the same distance f w.r.t. **C**. Since **I** is equivalent to **I**′, the terms are used interchangeably. However, an additional step must be performed to rotate the image 180°.

Given a point in the 3D world coordinate system, the respective 2D point in the image plane **p** can be estimated in three main steps: $1)$ Mapping from the world coordinate system **W** to the camera **C** coordinate system; $2)$ Projecting the 3D point in the retina coordinate frame; and $3)$ Translating from the retina coordinate frame to the image coordinate frame **I**.

Mapping from W to C

This step maps the transformation from a point $\mathbf{P}_W$ in the world coordinate system to the camera coordinate system, *i.e.*, $\mathbb{R}^3 \rightarrow \mathbb{R}^3$, given by a 3D rotation **R** and a translation **t**. The new coordinates of the point in the camera coordinate system $\mathbf{P}_C$ are given by

$$\mathbf{P}_C = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \mathbf{P}_W. \tag{2.1}$$

$\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix, $\mathbf{t} \in \mathbb{R}^{3\times1}$ is the translation vector, and $\mathbf{P}_W$ are the homogeneous coordinates in the world coordinate system, *i.e.*, $\mathbf{P}_W = [W_x, W_y, W_z, 1]^\top$.

Projection from C to retina

The retina is a 2D plane parallel to the image I, with origin at the principal point $\mathbf{c} = [c_x, c_y]^\top$ (see Figure 2.1). Therefore, the projection of the 3D points to the retina is a transformation from $\mathbb{R}^3 \rightarrow \mathbb{R}^2$. Given a point $\mathbf{P}_C = [P_x, P_y, P_z]^\top \in \mathbb{R}^3$, the coordinates of the projected point $\mathbf{p}_R = [r_x, r_y]^\top \in \mathbb{R}^2$ in the retina coordinate system are computed as

$$r_x = f \cdot \frac{P_x}{P_z} \text{ and } r_y = f \cdot \frac{P_y}{P_z}. \tag{2.2}$$

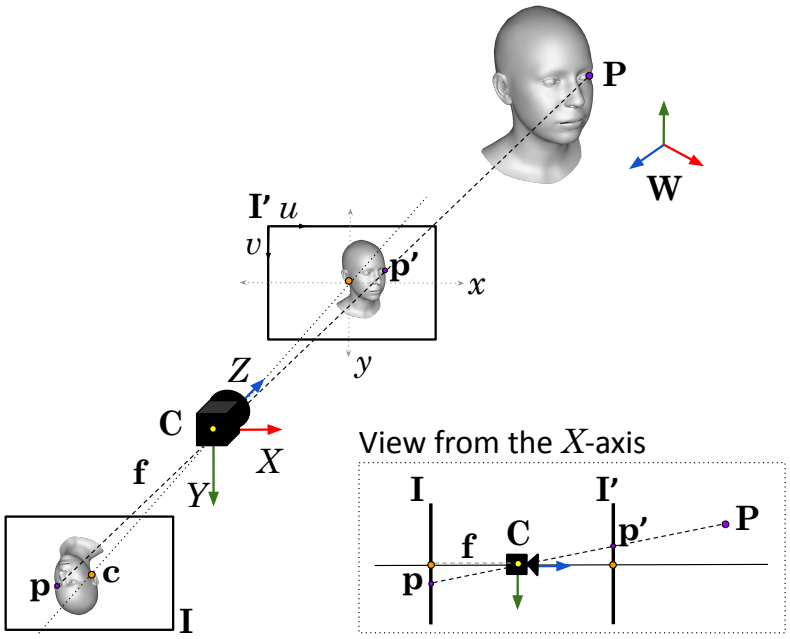


Figure 2.1: Pinhole camera model.

In matrix form, the homogeneous coordinates can be expressed as

$$\begin{pmatrix} r_x \\ r_y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}, \tag{2.3}$$

where f is the focal length.

Translation from retina to the image origin

In this last step, the coordinates of the 2D points in the retina coordinate system are translated to the image coordinate system **I**, *i.e.*, $\mathbb{R}^2 \rightarrow \mathbb{R}^2$. **I** has its origin at the top left corner of the retina. A point $\mathbf{p} = [p_x, p_y]^\top \in \mathbb{R}^2$ in the image plane is computed as follows

$$p_x = k_x r_x + c_x \quad \text{and} \quad p_y = k_y r_y + c_y, \tag{2.4}$$

with k_x and k_y being the density of pixels along x and y, and $\mathbf{c} = [c_x, c_y]^\top$ the principal point. In matrix form, these equations are represented in homogeneous coordinates as

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & 0 & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_x \\ r_y \\ 1 \end{pmatrix}. \tag{2.5}$$

Integrated transformation from W to the image plane

Based on equations (2.1), (2.3) and (2.5), the mapping from a 3D point in the world coordinate

system to the image plane can then be expressed as

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & 0 & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} & \mathbf{R} & & \mathbf{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} W_x \\ W_y \\ W_z \\ 1 \end{pmatrix}. \tag{2.6}$$

2.1.2 Camera Calibration

The terms in Eq. (2.6) can be rearranged as in Eq. (2.7). In this case, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is defined as the camera calibration matrix, which represents the intrinsic camera parameters.

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} fk_x & 0 & c_x \\ 0 & fk_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} & \mathbf{R} & & \mathbf{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} W_x \\ W_y \\ W_z \\ 1 \end{pmatrix}. \tag{2.7}$$

The previous equation can be approximated as follows

$$\mathbf{p} \sim \underbrace{(\mathbf{KR}\ \mathbf{Kt})}_{\mathcal{P}} \mathbf{P}_W \tag{2.8}$$

where $\mathcal{P} \in \mathbb{R}^{3\times4}$ is the perspective projection matrix and $\mathbf{R}$ and $\mathbf{t}$ represent the extrinsic camera parameters, *i.e.*, the camera pose.

Taking into account the image skew s, the intrinsic camera matrix is given by five parameters: $f_x = f \cdot k_x$, $f_y = f \cdot k_y$, s, c_x and c_y. f_x and f_y are the focal length in pixels, s is the skew of the two image axes and c_x and c_y are the coordinates of the principal point in the image. The calibration matrix then can be written as

$$\mathbf{K} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.9}$$

Camera calibration aims to find the intrinsic and/or extrinsic parameters of the camera, *i.e.*, $\mathbf{K}$, $\mathbf{R}$ and $\mathbf{t}$. In the literature, there exists different methods for camera calibration, using a calibration object as in [24], or without one in rigid [25] or non-rigid scenes [26].

In this thesis, where the camera calibration is required, these parameters are found using Zhang's calibration method [24]. Zhang's algorithm first compute the transformation between the calibration target and the image plane, using Direct Liner Transform (DLT) [27]. In our case, the calibration target is a checkerboard pattern with known dimension. Given a set of N images of the pattern, the respective transformations from the calibration target to the images, in addition to the intrinsic and extrinsic parameters of the camera are calculated by means of Singular Value Decomposition (SVD).

Lens distortion

The radial distortion introduced by the lens can be modeled using the polynomial radial distortion model in Eq. (2.10). This model is equivalent to the n-order Taylor series expansion of the

ideal distortion function. A second-order model with coefficients $\mathbf{k} = [k_1, k_2]^\top$ can be solved using Linear Least Squares (LLS). A third-order model with coefficients $\mathbf{k} = [k_1, k_2, k_3]^\top$ is considered sufficient for most real lenses [28, 29, 30].

$$\begin{aligned} x_d &= x_u(1 + k_1 r + k_2 r_u^2 + k_3 r_u^3 + ...), \\ y_d &= y_u(1 + k_1 r + k_2 r_u^2 + k_3 r_u^3 + ...) \end{aligned} \tag{2.10}$$

(x_d, y_d) and (x_u, y_u) correspond to the distorted and undistorted points, respectively, and $r_u = \sqrt{x_u^2 + y_u^2}$. In these equations the center of distortion is at the origin and the image coordinates are normalized.

The calibration and distortion parameters are later refined using maximum-likelihood estimation, with objective function

$$J = \sum_{i=1}^{N} \sum_{j=1}^{M} ||\mathbf{x}_{ij} - \hat{\mathbf{x}}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{P}_j)||^2, \tag{2.11}$$

where $\hat{\mathbf{x}}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{P}_j)$ represents the projection of 3D point $\mathbf{P}_j$ on the image plane based on Eq. (2.8), followed by the radial distortion on Eq. (2.10).

2.1.3 Triangulation

Triangulation estimates the scene geometry from two or more camera views, where the camera geometry is known in advance [31]. In addition to the known camera calibration, triangulation requires a set of point correspondences. Given a 2D point **p** in image **I** and its correspondence **p′** in a different view **I′**, the coordinates of a point **P** in the 3D space results from the intersection of the rays from the respective camera center **C** through the given 2D point, as shown in Figure 2.2.

Recalling Eq. (2.8), the projected point **p** from a 3D point **P** is proportional to the perspective projection matrix $\mathcal{P}$. With homogeneous coordinates, an scale factor α is added, *i.e.*,

$$\mathbf{p} = \alpha \mathcal{P} \mathbf{P}. \tag{2.12}$$

Given the nature of the problem, triangulation can be solved using the DLT algorithm. Since vectors **p** and **P** are parallel, Eq. (2.12) can be formulated as

$$\mathbf{p} \times \mathcal{P}\mathbf{P} = \mathbf{0}. \tag{2.13}$$

This equation is equivalent to

$$\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} \times \begin{pmatrix} \rho_1^\top \mathbf{P} \\ \rho_2^\top \mathbf{P} \\ \rho_3^\top \mathbf{P} \end{pmatrix} = \begin{pmatrix} p_y \rho_3^\top \mathbf{P} - \rho_2^\top \mathbf{P} \\ \rho_1^\top \mathbf{P} - p_x \rho_3^\top \mathbf{P} \\ p_x \rho_2^\top \mathbf{P} - p_y \rho_1^\top \mathbf{P} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{2.14}$$

where ρ_1, ρ_2 and ρ_3 are the first, second and third rows from $\mathcal{P}$. The third row is discarded, as it is a linear combination of the other two. Redefining and combining Eq. (2.14) for points **p** and

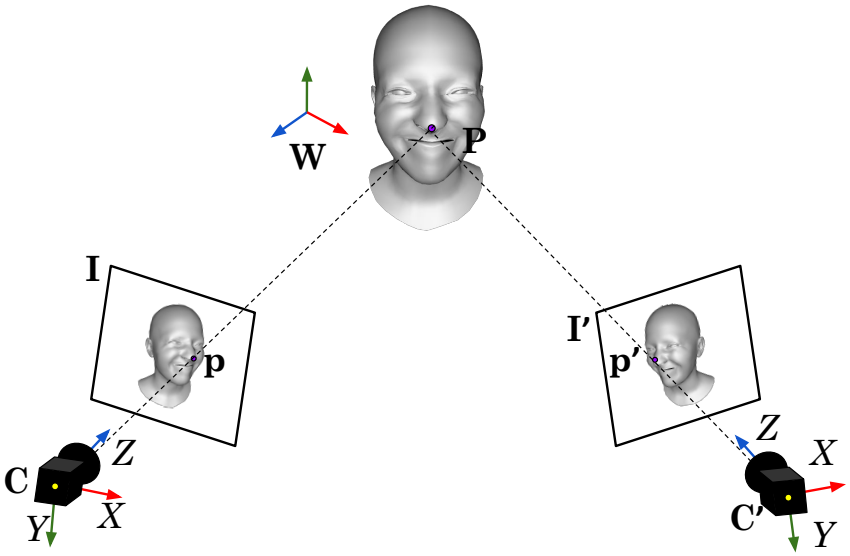


Figure 2.2: Triangulation.

$\mathbf{p}'$, the resulting system corresponds to

$$\begin{pmatrix} p_y\rho_3^\top - \rho_2^\top \\ \rho_1^\top - p_x\rho_3^\top \\ p'_y\rho'^\top_3 - \rho'^\top_2 \\ \rho'^\top_1 - p'_x\rho'^\top_3 \end{pmatrix} \mathbf{P} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.15}$$

The previous linear system, with form $\mathbf{AP} = \mathbf{0}$, can be solved using SVD. $\mathbf{P}$ is given by the eigenvector associated to the smallest eigenvalue of $\mathbf{A}^\top\mathbf{A}$.

2.1.4 Feature Detection and Tracking

The detection of image features and their correspondences across multiple images are fundamental problems in computer vision, with applications in motion estimation, 3D reconstruction, object recognition, among others.

Feature correspondences refers to the problem of identifying the projection of a point $\mathbf{P}$ in N images, *i.e.*, $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N$. The correspondences are matched after finding the set of feature points and their respective descriptors.

Feature correspondence matching can be challenging due to ambiguities in the images, under different view viewpoints and large variation in the appearance. Features are usually detected from distinctive regions in the image such as corners, such as in the Harris corner detector [32]. This detector is robust to rotation and partially invariant to affine intensity changes, but it is not scale invariant. This limitation has been targeted by using feature descriptors for correspondences matching, instead of using features directly. Descriptors such as the scale-invariant feature transform (SIFT) [33], based on the histogram of the image gradients, are more robust under variations in the image scale, illumination and noise.

FAST feature detector

Features from Accelerated Segment Test algorithm (FAST) is a machine learning method for corner detection [34]. Its main characteristic is the low computation inference time, in contrast to the Harris detector or SIFT, making it suitable for real-time applications.

FAST defines a circular array of pixels $\{\mathbf{c}_i\}_{i=1}^{N}$, with $N = 16$, surrounding a candidate corner point $\mathbf{p}$. If a set of M adjacent pixels in the circular array, $\mathbf{C} = \{\mathbf{c}_j\}_{j=1}^{M}$, have larger intensity than $\mathbf{p}$ plus a threshold value λ, *i.e.*,

$$I_{\mathbf{c}} > I_{\mathbf{p}} + \lambda, \forall \mathbf{c} \in \mathbf{C}, \tag{2.16}$$

$\mathbf{p}$ is classified as corner. The same applies if all the intensities are smaller than $\mathbf{p}$ minus λ, *i.e.*,

$$I_{\mathbf{c}} < I_{\mathbf{p}} - \lambda, \forall \mathbf{c} \in \mathbf{C}. \tag{2.17}$$

A decision tree is then trained on data labeled using the previous approach, to detect corners. In FAST-ER [35], the detector is optimized for repeatability using simulated annealing.

Kanade-Lucas-Tomasi feature tracker

The motion of an object or the motion from the camera can be exploited for feature matching and tracking. Given the motion of an object in the 3D space, *i.e.*, its motion field, optical flow refers to the projection of this motion onto the 2D image plane. For a point $\mathbf{p} = (x, y)$ moving across an image, optical flow aims to find its velocity $\mathbf{v} = (v_x, v_y)$, *i.e.*, the speed and the direction of the motion, by assuming that its intensity I is constant across frames:

$$I(x, y, t) = I(x + dx, y + dy, t + dt), \tag{2.18}$$

for a displacement $\mathbf{d} = (dx, dy)$ from time t to time $t + dt$. From this equation, it can be derived

$$I_x u + I_y v = -I_t, \tag{2.19}$$

where $I_\alpha = \frac{\delta I}{\delta \alpha}$, $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. This assumption implies that optical flow is constrained to a line, hence retrieving the flow component only in the direction of the spatial image gradient. This problem is known as the aperture problem.

In addition to the brightness constancy assumption, the motion can also be constrained to a constant flow that is locally smooth, *i.e.*, that neighbouring pixels follow the same motion. The Kanade-Lucas-Tomasi feature tracker (KLT) [36] leverages this assumption to define a method for feature matching. Given a set of N neighbouring pixels with constant velocity, a system based on Eq. (2.19) with form $\mathbf{Au} = \mathbf{b}$ can be defined, *i.e.*,

$$\begin{pmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \dots & \\ I_{xN} & I_{yN} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_{t1} \\ I_{t2} \\ \dots \\ I_{tN} \end{pmatrix}. \tag{2.20}$$

This system can be solved by means of LLS, with solution:

$$\hat{\mathbf{u}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}. \tag{2.21}$$

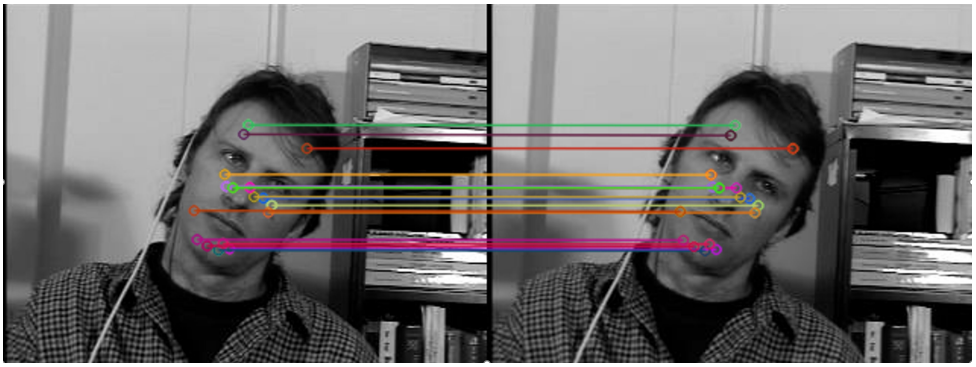


Figure 2.3: Feature detection and tracking. The features in the frames were detected with FAST-ER in the region of the head and tracked with the KLT feature tracker. Images from the Boston University dataset [37].

LLS is viable if $\mathbf{A}^\top\mathbf{A}$ is invertible. To that end, corner features are used for tracking, since the gradients in corners are different and have large magnitudes in both directions. For large motions, KLT uses a multi-scale approach based on Gaussian pyramids of the given images.

Figure 2.3 shows an example of FAST-ER feature points detected in the face and matched in consecutive frames using the KLT feature tracker.

2.1.5 Perspective-n-Point Problem

Perspective-n-Point (PnP) refers to the problem of estimating the camera pose, where a set of N 3D points $\{\mathbf{P}_i\}_{i=1}^N = \{\mathbf{P}_i \in \mathbb{R}^3 \mid 1 \leq i \leq N\}$ and their 2D correspondences in the image plane $\{\mathbf{p}_i\}_{i=1}^N = \{\mathbf{p}_i \in \mathbb{R}^2 \mid 1 \leq i \leq N\}$ are given. This problem assumes a calibrated camera to estimate the 6 degrees-of-freedom (DOF) camera pose. In short, PnP aims to estimate the extrinsic camera parameters, *i.e.*, the rotation $\mathbf{R}$ and translation $\mathbf{t}$ of the camera in Eq. (2.8). By using homogeneous coordinates, an scale factor is added, as in Eq. (2.12).

Based on the characteristic of the given system, different approaches have been proposed to solve the PnP problem, such as P3P [38, 39, 40], Efficient PnP (EPnP) [41] and SQPnP [42]. In addition, the Random Sample Consensus (RANSAC) iterative approach is usually applied to remove outliers from the point sets.

In this thesis, the camera pose is computed using an iterative approach which minimizes the reprojection error, *i.e.*,

$$\arg\min \sum_{i=1}^{N} \|\Pi(\mathbf{R}\mathbf{P}_i + \mathbf{t}) - \mathbf{p}_i\|_2^2 , \tag{2.22}$$

where $\Pi(\mathbf{P}) : \mathbb{R}^3 \mapsto \mathbb{R}^2$ denotes the perspective projection operator and i the index of the i-th point. Eq. (2.22) is minimized in the least-squared sense with respect to the rotation $\mathbf{R}$ and translation $\mathbf{t}$, using Levenberg-Marquardt iteration [43, 44]. An initial estimation is computed using DLT, where $N \geq 6$.

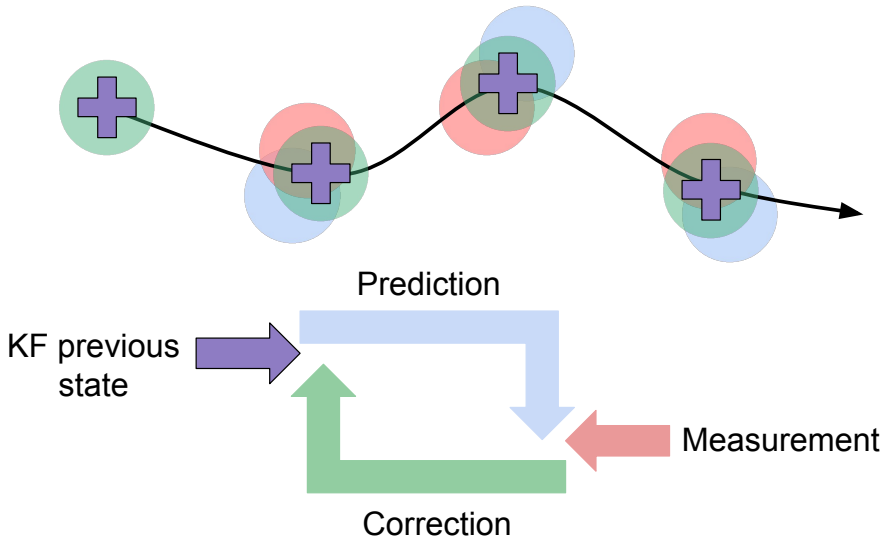


Figure 2.4: Kalman Filter diagram.

2.1.6 Distance Transform

Distance transform is a processing technique used in digital images, which finds the distance from each pixel **p** to the nearest boundary. This technique can be used for different tasks, such as image segmentation and skeletonization. For a binary image, the boundary pixels are defined as those with an intensity value of '0'. The distance transform then computes the shortest path from each pixel to the boundary, based on a defined distance metric d, as in Eq. (2.23). This metric can use the chessboard (Chebyshev) distance, Euclidean, Manhattan, among others. In this work, the distance transform is computed based on the Euclidean distance with a mask of size 3×3.

$$D(\mathbf{p}_i) = \min\{d(\mathbf{p}_i, \mathbf{q}) \mid \mathbf{I}(\mathbf{q})) = 0\}, \tag{2.23}$$

where $\mathbf{q} \in \Omega$, with Ω representing the set of pixels in the image with '0' intensity.

2.2 Statistical Filtering

Filtering refers to the problem of estimating the state of a system, given a noisy set of measurements. Systems can have a linear or non-linear nature, based on the characteristics of the state and the measurement.

The Kalman filter (KF), introduced in [45], proposes an approach for solving the linear optimal filtering problem, specifically for Gaussian dynamical systems. The filter, depicted in Figure 2.4, estimates the state in two consecutive stages: prediction and correction. Firstly, the KF computes a prediction $\hat{\mathbf{x}}_k^-$ based on the state at the previous step, $\hat{\mathbf{x}}_{k-1}$, and a defined dynamic model. In this stage, the filter also computes the covariance or uncertainty of the prediction, $\boldsymbol{P}_k^-$. Secondly, a correction is applied on the prediction and uncertainty based on the measurement $\mathbf{z}_k$ at the current step k, to compute the final estimates $\hat{\mathbf{x}}_k$ and $\boldsymbol{P}_k$.

In the prediction stage, the state and covariance extrapolation equations are given by

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_{k-1} \tag{2.24}$$

and

$$\boldsymbol{P}_k^- = \mathbf{A}_k \boldsymbol{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k, \tag{2.25}$$

where **A** in the state transition matrix, **B** is the control matrix, **u** is an input variable, **w** is the process noise and **Q** is the process noise covariance.

In the correction stage, the measurement innovation or residual is modeled as

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.26}$$

with **H** being the observation matrix, $\mathbf{x}_k$ the true system state and **v** the measurement noise. Kalman Filter assumes the measurement noise **v** has normal distribution with mean μ and standard deviation σ, *i.e.*, $\mathcal{N}(\mu, \sigma^2)$.

We then estimate the Kalman gain **K**, which minimizes the estimate variance. **K** is computed as follows

$$\mathbf{K}_k = \boldsymbol{P}_k^- \mathbf{H}_k^\top \left(\mathbf{H}_k \boldsymbol{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1} \tag{2.27}$$

where **R** is the measurement covariance.

Finally, the filter state and covariance are updated as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \tag{2.28}$$

and

$$\boldsymbol{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \boldsymbol{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top. \tag{2.29}$$

2.3 Deep Learning Models

Precursor learning pipelines such as the Multi-Layer Perceptron (MLP) [46, 47] and artificial neural networks as Neocognitron [48], laid the foundation to modern neural networks and deep learning models around a time when resources were limited and scarce. Due to these limitations, classical machine learning models with hand-crafted features were usually preferred in research. Nowadays, deep learning has gained great popularity, as high computational-powered hardware and large dedicated training datasets have become available.

Neural Network Training

Neural networks are trained using a loss function that estimates the dissimilarity between the predictions and the ground truth data. During training, the main target is to minimize the loss function, to reduce the differences between the prediction and ground truth.

The loss is minimized through an optimization step that adjusts the weights and biases of the neural network in each iteration. This process is performed by calculating the gradient of the loss function with respect to the network parameters. The gradients are computed using back propagation [49], carried out in each layer using the chain rule. The parameters are then updated in the opposite direction of the gradient, gradually moving towards the minimum value of the loss function.

Model Supervision

Deep learning and more general machine learning models can be classified in three categories based on the learning approach: supervised, self-supervised and semi-supervised. In the first case, the models are trained on labeled data, where they learn the mapping from an input X to an output Y, *e.g.*, in classification or regression tasks. Self-supervised approaches are trained without labels, where the models learn the data distribution instead for tasks such as clustering. Semi-supervised approaches, on the other hand, leverage a small set of labeled training data followed by a large set of unlabeled data for their prediction tasks.

2.3.1 Convolutional Networks Architectures

Convolutional Neural Networks (CNN) were proposed as an alternative to MLP networks with Fully Connected (FC) layers. In general, CNNs are composed of: (i) convolutional layers, which extract local features from the input data; (ii) pooling layers, to reduce the dimension of the feature maps; (iii) fully connected layers, which map the feature maps to output classes; and (iv) activation functions, which introduce non-linearity to the output, improving the modeling capacity of the network. Traditionally, sigmoid and tanH were commonly used as activation functions, while more recent architectures mainly use rectified linear unit (ReLU) or leaky ReLU.

In this Section, the convolutional models U-Net and residual networks (ResNet) are described. An encoder-decoder based on ResNet is used in our proposed pipeline for face alignment, in Chapter 4. In Chapter 6, we compare our Swin-Transformed-based generator with two baseline architectures widely used for image-to-image translation. The generator of the first architecture is based on U-Net, while the second generator is based on residual networks.

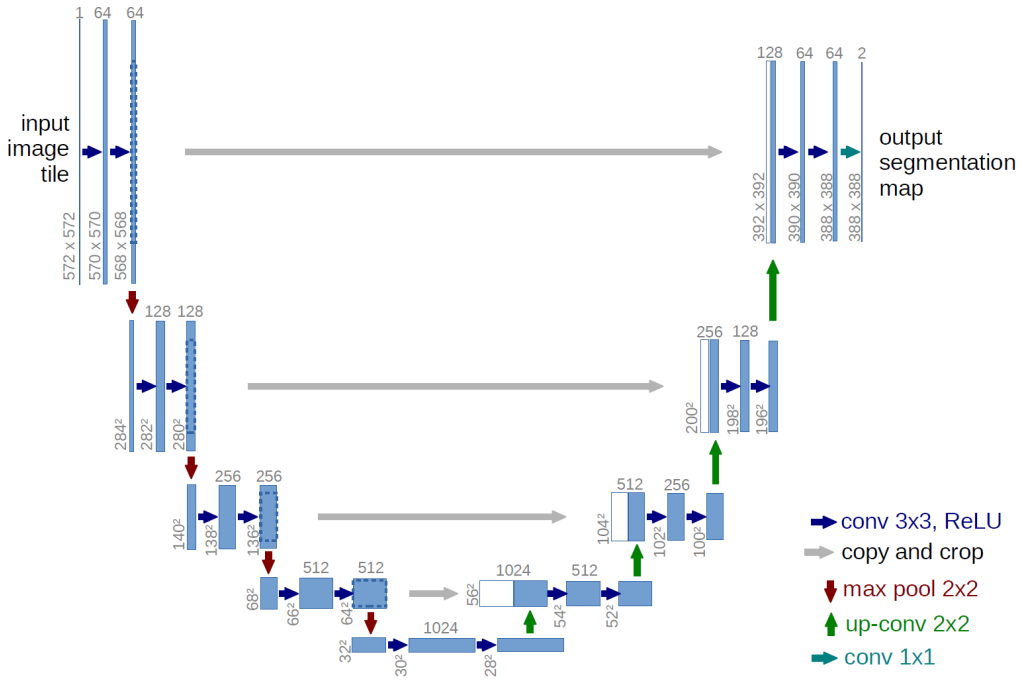


Figure 2.5: U-Net architecture. Source [50].

U-Net

U-Net was introduced in [50] for medical image segmentation with limited data. As depicted in Figure 2.5, this architecture encompasses an encoding path (left) and a symmetrical decoding path (right), with skip connections between the respective downsampling and upsampling blocks.

The encoding path, also known as contracting path, consists of consecutive blocks composed of convolutional layers and ReLU, followed by a max pooling layer. In each block, the feature channels are doubled and later cropped to be passed to the respective block in the decoding path. This skip connections enable the propagation of relevant information to high-resolution layers.

The decoding, or expansive path, maps the intermediate feature space to an output of similar size of the input data. This process occurs through a series of upsampling blocks composed of transposed convolutions, that halve the feature channels. The cropped feature maps from the encoding path are concatenated to the upsampled features of every block, which are then passed to convolutional layers and ReLU. The final layer applies a 1×1 convolution, to reduce the number of channels to the respective number of classes.

For segmentation tasks, U-Net is trained using a weighted cross entropy loss, given by

$$E = \sum_{x\in\Omega} w(\mathbf{x}) \log(p_{l(\mathbf{x})}(\mathbf{x})). \tag{2.30}$$

This loss follows a soft-max operation p_k for every pixel $\mathbf{x} \in \Omega$, with $k \in [1, K]$ being the respective channel and K the total number of classes. In Eq. (2.30), $l : \Omega \to [1, K]$ corresponds

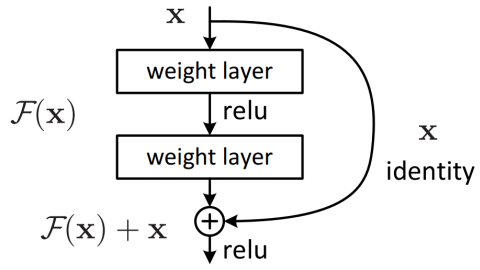


Figure 2.6: ResNet block. Source [51].

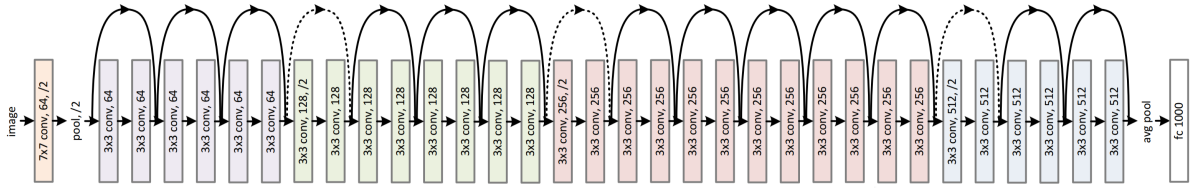


Figure 2.7: ResNet-34 architecture. Source [51].

to the true label of each pixel and $w : \Omega \to \mathbb{R}$ is a weight map to assign higher weights to the border of the segmented regions.

Residual Neural Network

Residual networks were proposed in [51] to address the vanishing gradient problem in deep convolutional neural networks for image recognition. Their main feature is the use of consecutive residual blocks, each composed of a residual connection linking the input to the output of the block, as depicted in Figure 2.6. Therefore, the input of each residual block is passed through two separate paths: on one side, the input is processed through a series of convolutional layers, batch normalization and ReLU; on the other side, the input is passed directly to the output, where it is added to the output of the processing layers. This process can be expressed as

$$\mathbf{y} = F(\phi(\mathbf{x})) + \mathbf{x}, \tag{2.31}$$

where **x** and **y** represent the input and output vectors of a ResNet block, $F(\cdot)$ is a residual function and ϕ is the non-linear activation or normalization applied.

An example of a deep residual network architecture is illustrated in Figure 2.7. This network is named ResNet-34, as it is composed of 34 layers. By connecting their input and output, residual blocks allows the design of deeper neural networks, with more efficient training.

2.3.2 Transformers

Transformers, introduced in [52], were proposed as an alternative architecture to recurrent neural networks (RNN) for natural language processing (NLP) tasks, such as machine translation. Their main feature is the multi-head self-attention (MSA) mechanism, targeted to processing sequential data in NLP, as in text or audio sequences. The attention mechanism aims at finding

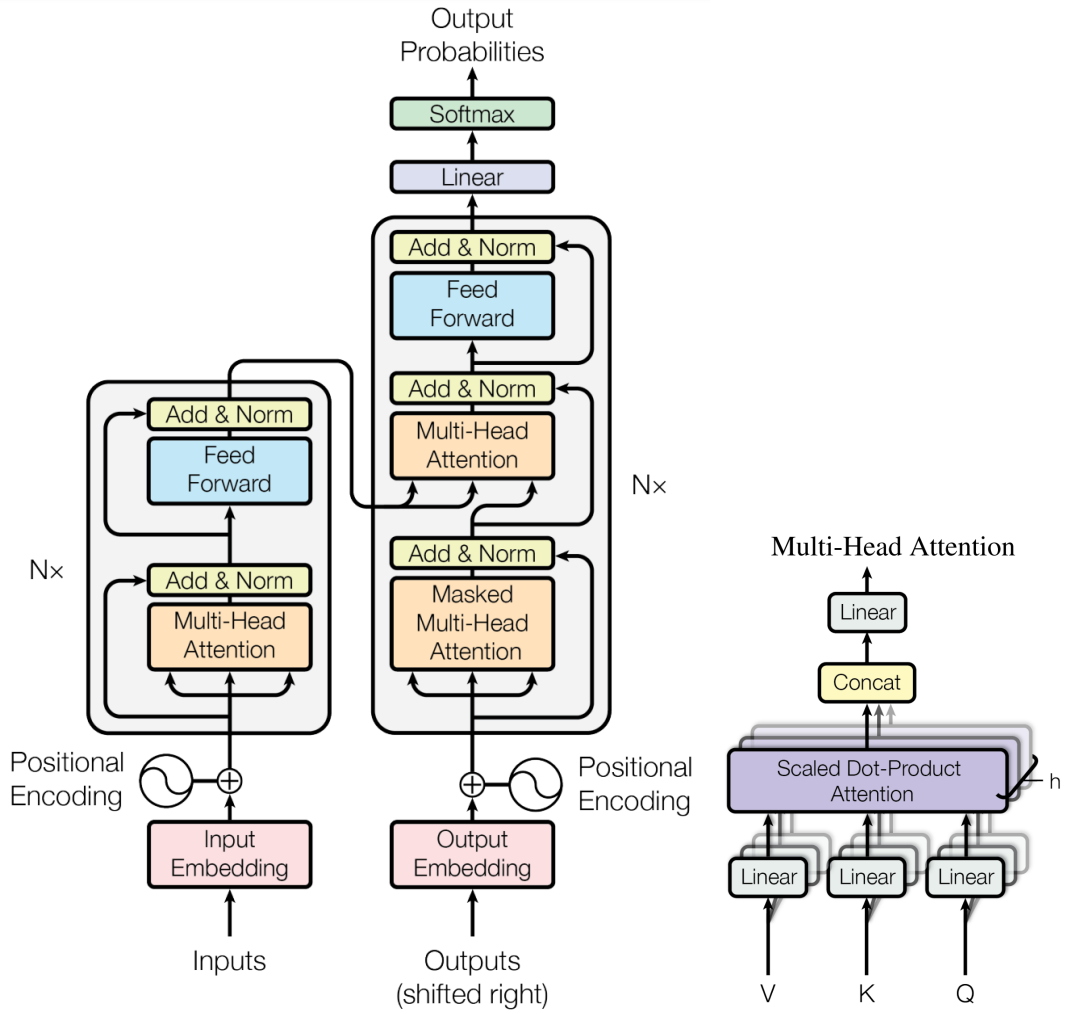


Figure 2.8: Left: Transformer architecture. Right: Multi-head attention. Source [52].

the highest correlation between words in a sequence, based on the training data. Given an input sequence, this mechanism computes the attention weights to each word embedding in parallel, in contrast to RNNs where they are computed sequentially. The architecture proposed by [52] is shown in Figure 2.8 (left). The Transformer encoder, illustrated on the left side, is composed of two residual blocks stacked consecutively.

Transformers take an input text and divide it into n-grams, which in turn are converted to tokens. The tokens are then mapped into vectors using a word embedding table, integrating the positional information of each token. These tokens are weighted using the attention mechanism, to emphasized relevant parts of the given text. To that end, each token is associated with a corresponding query Q, key K and value V.

The attention weights are computed by calculating the similarity between queries and keys, as follows

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V. \tag{2.32}$$

The output of each query results from the dot product of the weights.

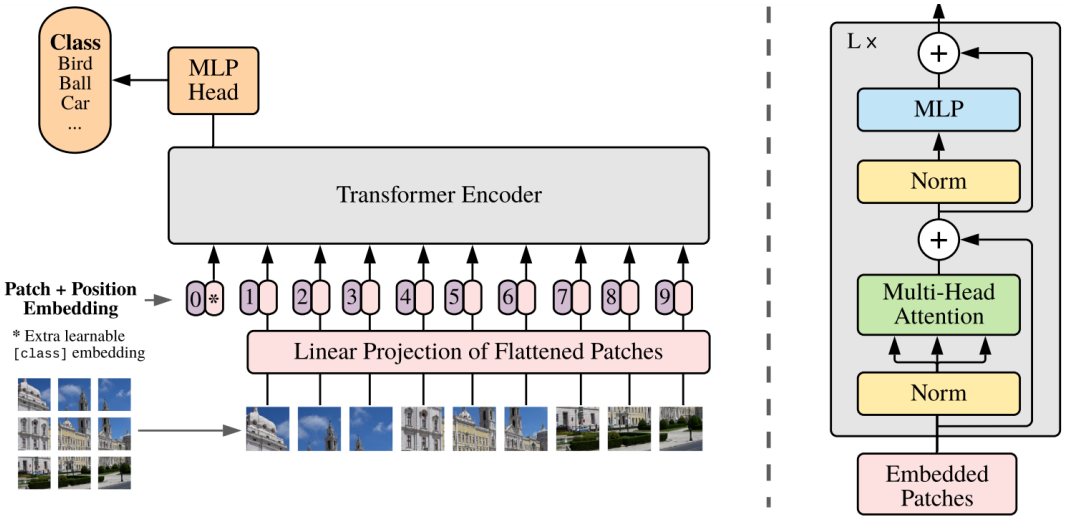


Figure 2.9: Left: Vision Transformer (ViT) architecture. Right: ViT encoder. Source [54].

In MSA with N heads, the model employs multiple set of queries, keys and values, *i.e.*

$$\text{Multi-head}(Q, K, V) = \text{Concat}_{i \in N}(\text{Attention}(XW_i^Q, XW_i^K, XW_i^V))W^O, \tag{2.33}$$

as depicted in Figure 2.8 (right). X is the concatenation of word embeddings, W_i^Q, W_i^K and W_i^V are projection matrices for attention head i, and W^O is the final projection matrix of the MSA head.

To improve the network convergence, [53] proposed to integrate the normalizing layers before the attention mechanism.

Vision Transformers

Vision Transformers (ViT) [54] were designed specifically to target image processing tasks. In contrast to NLP-based transformers which split text into tokens, ViT divides an input image into patches and map these patches to a vector. This vector is then projected into a feature space and later passed to a Transformer encoder, which processes it similarly to the token embeddings, as shown in Figure 2.9 (left). The Transformer encoder for ViT is based on the transformers for NLP, as depicted in Figure 2.9 (right).

More in detail, ViTs take an input image of size $W \times H \times C$, where W, H and C denote the width, height and number of channels. The image is divided into patches of size $P \times P \times C$, *e.g.* $16 \times 16 \times 3$ for an RGB image, to reduce the computational cost of handling single pixels. These patches are processed by a linear operator, to reduce their dimension and generate a vector per patch, named as patch embedding. Likewise, the position of each patch is mapped to a vector, using position encoding. The patch and positional embeddings are added and fed to the Transformer encoders.

The attention mechanism introduces semantic relations between image patches gradually, similarly to NLP transformers. To that end, a weighted sum of the input data is computed, where each weight is estimated by comparing query and key vectors.

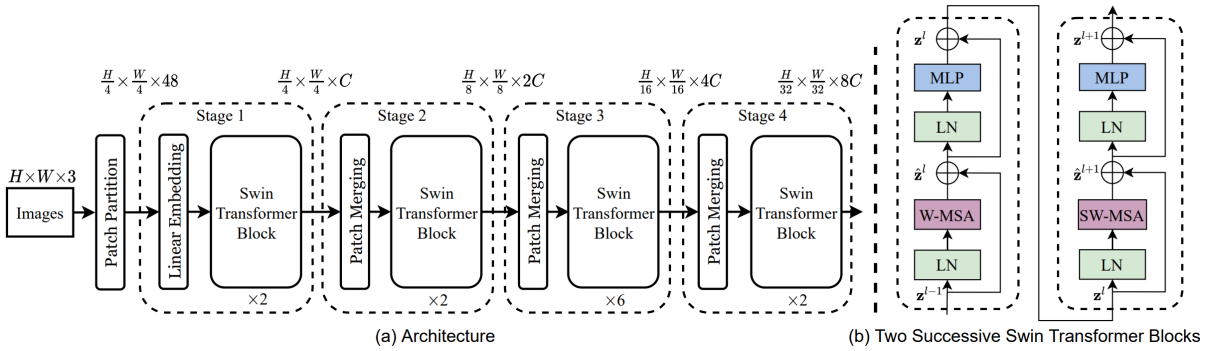


Figure 2.10: Swin-Transformer architecture (Swin-T). Source [55].

Swin Transformers

ViTs are limited due to their computational complexity, which depends on the sizes of the images and patches. Hence, when the image size increases or the patches size decreases, the complexity of the model increases with quadratic cost w.r.t. the number of patches. Swin Transformers were proposed in [55, 56] to target this limitation and enable the scalability of ViTs. This architecture, depicted in Figure 2.10, introduces a hierarchical Transformer with shifted windows that has linear computational complexity w.r.t. the image size.

Swin Transformers propose a modified attention mechanism, based on W-MSA (Window-based Multi-Head Self-Attention) and SW-MSA (Shifted-Window Multi-Head Self-Attention), shown in Figure 2.10 (right). These modules replace the Multi-Head Attention of ViTs in Figure 2.9, which follows the Layer Normalization (LN) step.

The W-MSA and SW-MSA modules are each part of Swin Transformer blocks that are always consecutive. Hence, each stage in the architecture consists of L Swin Transformer blocks, where L is always a multiple of 2. Output features $\hat{\mathbf{z}}$ and $\mathbf{z}$ of W-MSA, SW-MSA and MLP modules for consecutive blocks l and $l+1$ are estimated as

$$\begin{aligned}\hat{\mathbf{z}}^{l} &= \text{W-MSA}(\text{LN}(\mathbf{z}^{l-1})) + \mathbf{z}^{l-1},\\ \mathbf{z}^{l} &= \text{MLP}(\text{LN}(\hat{\mathbf{z}}^{l})) + \hat{\mathbf{z}}^{l},\\ \hat{\mathbf{z}}^{l+1} &= \text{SW-MSA}(\text{LN}(\mathbf{z}^{l})) + \mathbf{z}^{l},\\ \mathbf{z}^{l+1} &= \text{MLP}(\text{LN}(\hat{\mathbf{z}}^{l+1})) + \hat{\mathbf{z}}^{l+1},\end{aligned} \tag{2.34}$$

as depicted in Figure 2.10 (right).

Similarly to MSA, W-MSA divides the image into patches of size $P \times P \times D$, in the 'Patch Partition' stage in Figure 2.10 (left). Patch partition is followed by a linear embedding layer, which projects the patch tokens to a given dimension C. The feature map at this step is fed to a pair of Swin Transformer blocks, which preserves the number of patch tokens.

One of the novelties of Swin Transformers is merging patches in deeper layers, where MSA is applied within a window, instead of the entire feature map. This process is performed at the 'Patch Merging' step in the subsequent stages in Figure 2.10 (left), followed by a linear embedding layer to reduce the number of tokens. Patch merging is applied on the feature maps, on groups of 2×2 neighboring patches.

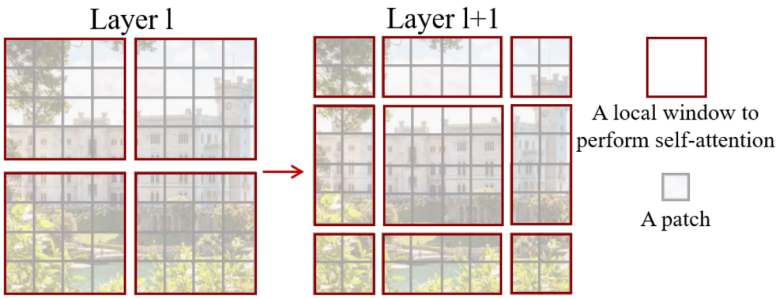


Figure 2.11: Shifted windows between adjacent layers. Source [55].

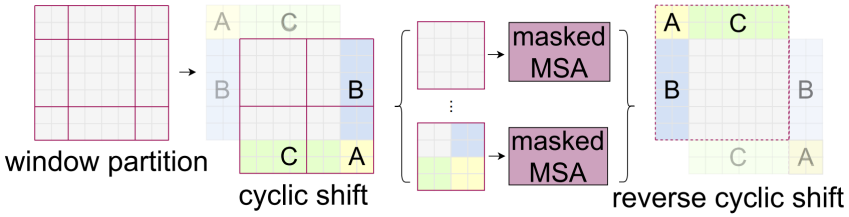


Figure 2.12: Windows padding using cyclic shift. Source [55].

Another novelty is the SW-MSA to exploit the connectivity between consecutive windows. After patch merging and W-MSA, each window is shifted in the following layer, as shown in Figure 2.11. Since some resulting windows have different dimensions, a cyclic-shifting toward the top-left direction approach was proposed to apply the attention mechanism. This process, illustrated in Figure 2.11, introduced a scheme to pad the window with a cyclic shift and apply the MSA using a masking mechanism within the desired sub-window only. A reverse cyclic shift is later apply, to recover the initial arrangement of sub-windows.

Swin Transformers additionally include a position bias B in Eq. (2.32) to each head during the computation of the self-attention, as follows

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}} + B\right)V, \tag{2.35}$$

following related work such as [57]. In this equation, $Q, K, V \in \mathbb{R}^{M^2 \times d}$, with M^2 being the total number of patches in a window and $B \in \mathbb{R}^{M^2 \times M^2}$.

An updated version V2 of Swin Transformers was introduced in [56], to improve their scaling capabilities to higher resolution images, with less training data. The main differences lie within the attention block and Swin Transformer arrangement, as shown in Figure 2.13.

One improvement corresponds to the bias B in Eq. (2.35), which is replaced with a log-spaced continuous position bias (Log-CPB), as depicted in Figure 2.13. Furthermore, the attention mechanism introduces a scaled cosine function applied to the query and key vectors, Q and K, in the pixel level, to restrict the influence of few pixel pairs in the computation of the attention maps. Finally, the structure of a Swin Transformer block is adjusted to apply the layer normalization step after the attention and MLP, instead of before as in ViTs and Swin Trans-

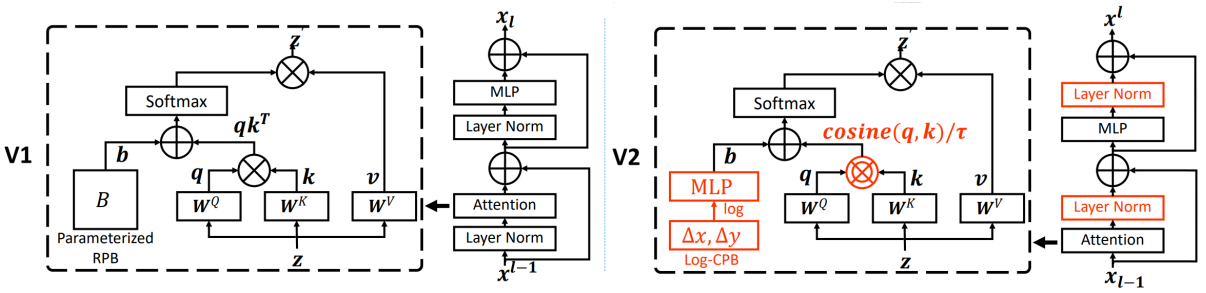


Figure 2.13: Architecture of Swin Transformers V1 vs. V2. Source [56].

former V1. This configuration limits the activation amplitudes, particularly in deeper layers, improving stability during training.

Our sign image synthesis architecture in Chapter 6 leverages a Swin Transformer V2 as the backbone of the generator in the proposed pipeline.

2.3.3 Generative Models

Generative models learn to model the input data distribution [58], in contrast to discriminative models, which learn to identify and classify instances.

Based on [59], existing deep generative models can be grouped mainly in the following categories: variational autoencoders, generative adversarial networks, diffusion probabilistic models, energy-based models and normalizing flows. This Section focuses on the first three categories, which comprise the most common architectures for generative tasks.

Variational Autoencoders (VAEs)

VAEs, introduced in [60], are probabilistic generative architectures that aim to produce a reconstruction that matches an input signal. A VAE consists of an encoder **E** and a decoder **D**, as shown in Figure 2.14. The encoder **E** ensures the distribution of the latent code **z** follows a variational distribution, such as standard Gaussian distribution with mean μ and standard deviation σ, while the decoder maps the latent code to the input space. To that end, the encoder learns the mapping to output the parameters of the given distribution.

VAEs are trained to maximize the log likelihood function, to generate an output $\mathbf{x}'$ that closely resembles input $\mathbf{x}$. The training loss is composed of a reconstruction error $\mathcal{L}_{rec}$ and the Kullback–Leibler (KL) divergence function D_{KL}, *i.e.*,

$$\mathcal{L}_{VAE} = \mathcal{L}_{rec} - D_{KL}(q_\phi(z|x) \,||\, p_\theta(z|x)). \tag{2.36}$$

The KL divergence targets to maximize the probability that distributions $q_\phi(z|x)$ and $p_\theta(z|x)$, computed by the encoder and decoder respectively, overlap. The reconstruction error is usually given by the L2 or cross entropy loss.

Generative Adversarial Networks (GANs)

GANs were proposed in [61] as an alternative generative model. The architecture is composed of a generator **G** and a discriminator **D**, as depicted in Figure 2.15. **G** learns the training data

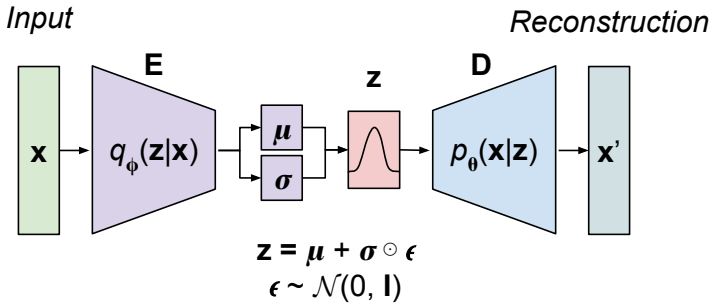


Figure 2.14: Variational Autoencoder (VAE) architecture. The VAE aims to generate a reconstruction $\mathbf{x}'$ that is ideally identical to input $\mathbf{x}$. The input goes through an encoder **E**, which maps $\mathbf{x}$ to a low-dimensional compressed latent representation $\mathbf{z}$ of the input. $\mathbf{z}$ follows an standard normal distribution with mean μ and standard deviation σ. The latent representation is mapped to the size of the original input $\mathbf{x}$ via a decoder **D**.

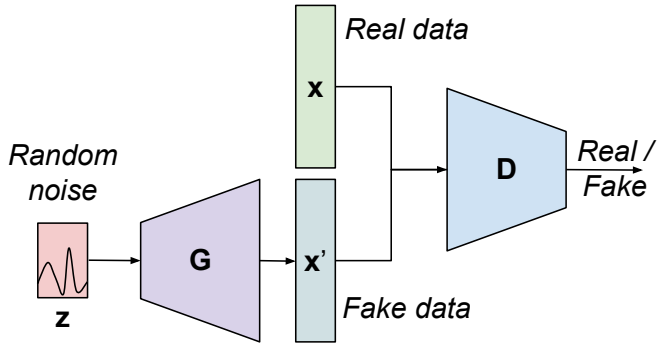


Figure 2.15: Generative Adversarial Network (GAN) architecture. From a random noise signal $\mathbf{z}$, the generator **G** tries to produce a fake sample $\mathbf{x}'$ that can deceive the discriminator **D**. The generator and discriminator are trained in tandem, until the discriminator is not longer able to recognize real from fake data.

distribution and is able to generate fake samples. On the other hand, **D** learns to identify real data from the data generated by **G**. While training, **G** and **D** compete with each other: **G** learns to create more instances that fool **D**, and **D** penalizes **G** if the generated data is implausible.

GANs training occurs in two stages, discriminator and generator training, that happens in an alternating manner. In one stage, **D** is trained with an adversarial loss, as in Eq. 2.37, that penalizes **D** for every misclassification. At this stage, the weights of the generator are fixed, while **D** tries to maximize Eq. 2.37, to differentiate the classification of real and fake instances.

$$\mathcal{L}_{\mathbf{D}} = \mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x}}[\log(\mathbf{D}(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}}[\log(1 - \mathbf{D}(\mathbf{G}(\mathbf{z})))] \tag{2.37}$$

$\mathbf{D}(\mathbf{x})$ corresponds to the probability given by **D** that instance $\mathbf{x}$ is real. $\mathbf{G}(\mathbf{z})$ indicates the output of **G** given random noise $\mathbf{z}$ as input, while $\mathbf{D}(\mathbf{G}(\mathbf{z}))$ is the probability estimated by **D** that a fake instance is real. $\mathbb{E}_{\mathbf{x}}$ and $\mathbb{E}_{\mathbf{z}}$ are the expected values for all the real and fake data instances.

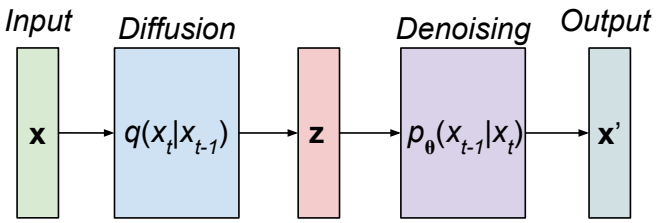


Figure 2.16: Diffusion model architecture. Given a sample input **x**, the model uses a fixed forward diffusion module that gradually adds Gaussian noise to **x** from step $t-1$ to t. The denoising step reverses the forward diffusion process progressively, where a neural network is trained to predict an image at step $t-1$, from the noisy image at step t.

During the generator training, **G** produces an output **x**′ from a random noise **z** . The weights of **D** are set constant, while it classifies **x**′ as real or fake. The generator is trained by minimizing the adversarial loss from Eq. 2.37, computed at the discriminator level. This equation can be rewritten as

$$\mathcal{L}_{\mathbf{G}} = \mathbb{E}_{\mathbf{z}}[\log(1 - \mathbf{D}(\mathbf{G}(\mathbf{z})))], \tag{2.38}$$

as the distribution of real instances has no effect on the generator training. Then, backpropagation follows from the output of **D** until **G**, to compute the gradients and adjust the weights of the generator.

Different variations of GANs architectures propose alternative loss functions for training **G** and **D**, or modified frameworks targeted to improve the quality of the reconstructions or to integrate additional information, such as labels.

Diffusion Models

Diffusion models [62, 63] are composed of two main processes: forward diffusion and reverse diffusion, or denoising, as shown in Figure 2.16. In the first process, noise is added gradually to the input following a Markov chain, until only noise remains. This process is defined as

$$q(x_t, x_{t-1}) = \mathcal{N}(x_t, \mu, \sigma^2), \tag{2.39}$$

where q is the forward process, x_t is the output at step t, x_{t-1} the input, and $\mathcal{N}$ is a Gaussian distribution with mean μ and variance σ^2.

The reverse diffusion process, $q_\theta(x_{t-1}, x_t)$, aims to restore the input back to its original form in a set of steps, through a neural network. In that sense, the output at t is used as the input in $t-1$. The noisy sample **z**, generated in the forward diffusion step, acts as at the latent code of the architecture, keeping the same dimension of the input data. During the denoising, the network is trained while maximizing log of likelihood. These loss is reduced to the L2 loss, where the generated image and image at time step $t-1$ are evaluated.

2.4 Face Analysis Tasks

This Section describes face detection and alignment, two tasks widely researched in face analysis. The off-the-shelf machine-learning and deep-learning-based face detection pipelines used in this thesis are discussed. 2D, 3DA-2D and 3D face alignment is also introduced.

2.4.1 Face Detection

Given an input image I, face detection aims to identify faces and retrieve the coordinates of the bounding boxes enclosing them, *i.e.*, $BB = \{\mathbf{p}_{TL}, \mathbf{p}_{TR}, \mathbf{p}_{BL}, \mathbf{p}_{BR}\} \in \mathbb{R}^2$ for every detected face. The bounding boxes can also be used in related downstream tasks such as facial recognition, alignment, emotion recognition, among others, where each detected face is processed separately.

Earlier machine learning methods exploited hand-crafted feature extractors to train classifiers to detect faces [64, 65]. These methods are usually robust in controlled environments with low occlusion and constant illumination, and frontal or near-frontal faces. More recently, deep learning approaches targeted the detection of faces in the wild, leveraging large scale datasets of faces under varying settings.

Similarly to more general object detectors, most approaches locate the instances by defining one or multiple windows that iterate over the image. As a result, one single object can be located from multiple windows. Therefore, non-maximum suppression is usually applied in overlapping windows to select a single bounding box.

Classical machine-learning-based face detection

Viola and Jones introduced a face detection pipeline in [64], based on a cascade of boosted classifiers constructed from low-level features. The detector is inspired on the Adaptive Boosting (AdaBoost) [66], which classification output results from the linear combination of multiple weak classifiers. Each weak classifier is based on a Haar-like feature and returns a binary output, specifying if the face was detected or not on a sliding window in the image.

The Viola-Jones face detector uses a labeled training set to estimate the best features and adjust the weights of the weak classifiers. The new weights are computed based on their accuracy, assigning larger weights to more robust classifiers. The classifiers are then sorted based on their robustness and used sequentially in every sub-window to be analyzed. If a classifier detects a face in the sub-window, the following classifier is applied to the same sub-window as well. Otherwise, the sub-window is immediately discarded. This process speeds up the detection, while rejecting regions that are less likely to have a face at an early stage.

King proposed in [67] Max-Margin Object Detection (MMOD), an approach that optimizes sliding-window-based object detectors which are linear in their parameters. This method improves the non-maximum suppression pipeline used after the detection, to include windows that might be ignored and cover relevant objects. For face detection, the method is adapted to the HOG sliding window classifier [68], where features are extracted using Histogram of Oriented Gradients (HOG). The Support vector machines (SVM) classifier from [69] is replaced with MMOD, with an improved performance.

These algorithms robustly detect faces with low computational cost in devices with limited resources. Nevertheless, they has several limitations such as requiring frontal faces or not being robust under occlusion. For non-frontal faces, multiple detectors can be trained with different head poses. However, occlusion cannot be handled in a similar manner.

Deep-learning-based face detection

MTCNN [70] introduced a multi-task cascaded CNN for face detection and alignment, which performs in real time on CPU. The architecture integrates three convolutional networks connected in a cascade, for both tasks. The image is first resized on multiple scales, where the first proposal network (P-network) retrieves possible face matches with a low threshold. The second stage refines the previous detection with a R-network, applying non-maximum suppression to remove overlapping faces. In the last stage, the output network (O-network) performs additional refinement on the bounding boxes, to provide accurate detection.

FaceBoxes [71] proposed a CNN-based real-time face detector, which shows high robustness at detecting faces on a single core CPU. The network is composed of two main blocks, the first being the Rapidly Digested Convolutional Layers (RDCL) and the second the Multiple Scale Convolutional Layers (MSCL). In the RDCL, two convolutional layers with large stride are applied sequentially to the input image, each one followed by a pooling layer. This step largely reduces the input image by a factor of 1/32. The second block is based on RPN, with three Inception [72] blocks followed by convolutional layers. This block generates multi-scale feature maps and can handle faces with multiple dimensions. An anchor densification framework is proposed, to increase the number of anchors around the receptive field and improve the detection of smaller faces.

2.4.2 Face Alignment

Face alignment refers to the detection of a predefined set of fiducial facial points, or facial landmarks, in a given input image. Many of these landmarks are semantically meaningful, referring, *e.g.*, to the corners of the eyes and lips, the tip of the nose and the contours of the eyebrows. This set of landmarks can be either sparse [73, 74] or dense [75, 76, 77, 78].

In the past years, many researchers strove to unify and standardize the set of keypoints used for face alignment [79, 80, 81, 82, 83]. The most common set defines 68 fiducial points on the eyes, nose, lips, eyebrows and around the boundary of the face, following the convention proposed in Multi-PIE [84] (see Figure 2.17).

Landmarks are defined in the 2D or 3D space, and can be grouped in four categories [83]:

- Semi-frontal 2D landmarks.
- Profile 2D landmarks.
- 3DA-2D (3D Aware 2D) landmarks.
- 3D landmarks.



Figure 2.17: Facial landmarks aligned to a detected face from the Boston University dataset [37].

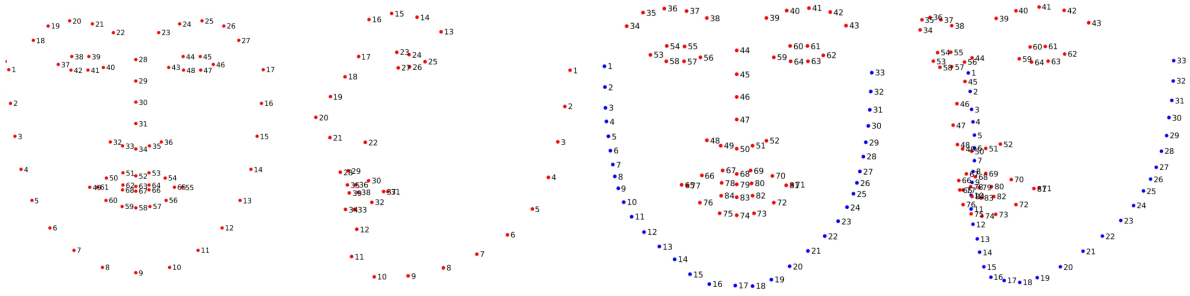


Figure 2.18: Predefined set of landmarks. From left to right: Multi-PIE 2D landmarks for frontal and profile views; Menpo 3DA-3D and 3D landmarks for frontal and left yaw pose. Source [83].

2D landmarks, as in Multi-PIE, are defined around the face contour and suffer from discrepancies for frontal and profile faces [75]. In Multi-PIE, 39 fiducial points are annotated in profile faces, in contrast to the 68 landmarks defined for semi-frontal poses (see Figure 2.18). Although this convention is useful for tasks such as face segmentation, it is error prone for optimization problems, *e.g.*, when minimizing the reprojection error [85, 86].

Recently, a new set of landmarks in the 2D space was defined, namely 3DA-2D landmarks [83], to alleviate the discrepancy problem. These landmarks correspond to the projection of the 3D landmarks in the image space and maintain one-to-one correspondences across different poses, similarly to 3D landmarks [87] (see Figure 2.18). The collection of datasets with 3DA-2D and 3D landmarks annotations has been possible thanks to 3D Morphable Models (3DMMs) [88] and deep architectures [73, 13, 89, 90].

Landmarks are commonly represented with coordinates in the image space, heatmaps [73] or random variables [87]. In addition to the landmark positions, the last two representations

provide information about the uncertainty of each landmark.

Face alignment has been widely researched in the computer vision community. Before the extended use of deep learning, classical regression-based approaches were leading alignment competitions [91, 81]. With the introduction of large datasets, deep-learning based methods outperformed previous approaches and boosted their robustness under different challenges, including illumination variation, motion blur and large head rotations. Relevant 2D and 3D face alignment approaches are described in Chapter 4.2.

2.5 Evaluation Metrics

In this thesis, the performance of the proposed approaches for head pose estimation and face alignment in Chapters 3 to 5 are evaluated based on the difference between the ground truth and predicted values. Furthermore, the quality of the synthesized images by the generative models in Chapter 6 are evaluated using the structural similarity index measure, peak signal-to-noise-ratio, position error and Fréchet inception distance.

Mean and Root Mean Squared Error

The mean squared error (MSE) and root mean squared error (RMSE) are metrics that measure the performance of a predictor. The MSE is given by the average of the square of the difference between the observed and predicted values, *i.e.*,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i)^2, \tag{2.40}$$

where n represents the number of samples, $\mathbf{s} = \{s_i\}_{i=1}^{n}$ are the ground truth values and $\hat{\mathbf{s}} = \{\hat{s}_i\}_{i=1}^{n}$ are the estimated values. On the other hand, the RMSE is calculated as the square root of the average of the square of the difference, *i.e.*,

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(s_i - \hat{s}_i)^2}. \tag{2.41}$$

Mean Absolute Error

The mean absolute error (MAE) is defined as the average of the sum of absolute errors. Following the previous notations for ground truth $\mathbf{s}$ and predicted values $\hat{\mathbf{s}}$, the MAE is computed as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|s_i - \hat{s}_i|\,. \tag{2.42}$$

Normalized Mean Error

The normalized mean error (NME) corresponds to the average of the sum of the L2 distance between the ground truth and estimated values, where each distance is scaled by a normalization factor d, *i.e.*,

$$NME = \frac{1}{n}\sum_{i=1}^{n}\frac{||s_i - \hat{s}_i||_2}{d}. \tag{2.43}$$

Standard Deviation

The standard deviation (STD) is given by Eq. (2.44), where μ corresponds to the mean of $(\mathbf{s}-\hat{\mathbf{s}})$ and is given by Eq. (2.45). Eq. (2.44) uses the Bessel's correction to remove the bias in the estimation.

$$STD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left|(s_i - \hat{s}_i) - \mu\right|^2} \tag{2.44}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i) \tag{2.45}$$

Structural Similarity Index Measure

The structural similarity index measure (SSIM) is a perceptual metric that evaluates the image quality, based on a weighted combination of the luminance (lum), contrast ($cont$) and structure ($struct$) between samples x and y. It is computed as follows

$$SSIM(x,y) = lum(x,y)^{\alpha} \cdot cont(x,y)^{\beta} \cdot struct(x,y)^{\gamma}. \tag{2.46}$$

The luminance, contrast and structure are computed from Eq. (2.47), (2.48), (2.49), respectively. $\mu_{(.)}$ is the sample mean, $\sigma^2_{(.)}$ is the variance and σ_{xy} the covariance of x and y. c_1, c_2 and c_3 are scalars with $c_1 = 0,0001D^2$, $c_2 = 0,0009D^2$, $c_3 = c_2/2$ and D is the dynamic range of the samples. For images, $D = 2^b - 1$, with b being the number of bits per pixel.

$$lum(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{2.47}$$

$$cont(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{2.48}$$

$$struct(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{2.49}$$

If α, β and γ are set equal to 1, Eq. (2.46) can be rewritten as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \tag{2.50}$$

Peak Signal-to-Noise Ratio

The peak signal-to-noise-ratio (PSNR) assesses the quality of images that have been subjected to compression, *e.g.*, in reconstruction tasks. The PSNR is expressed as

$$PSNR = 10 \cdot \log_{10}\frac{I^2}{MSE}, \tag{2.51}$$

with I being the maximum possible intensity value in the image and MSE the mean square error between the ground truth and the reconstructed images, defined in by Eq. (2.40). I is equivalent to the dynamic range of the samples D, where $I = 2^b - 1$, and b is the number of bits per pixel. For 8-bits images, I is equal to 255.

Position Error

The position error (PE) corresponds to the mean Euclidean distance between the ground truth and estimated body joints. It is equivalent to the RMSE in Eq. (2.41), where n is the total number of joints. In this work, the body joints are estimated in the image frame, instead of the 3D space.

Fréchet Inception Distance

The Fréchet inception distance (FID) was introduced in [92] to evaluate the quality of the images produced by generative models, based on the statistical distribution of the generated images, g, in contrast to the real images, r. To that end, a pre-trained Inception network [72] is used to extract feature maps from r and g independently. These embeddings are modeled as multivariate normal distributions, with mean μ and covariance matrix Σ. Given these parameters, the FID score is computed using Eq. (2.52), where $\mathrm{Tr}(\cdot)$ represents the trace of a matrix.

$$FID = ||\mu_r - \mu_g||^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}) \tag{2.52}$$

3 Head Pose Estimation

Contents

3.1 Introduction 36
3.2 Problem Formulation 38
3.3 Related Work 38
3.4 Head Pose Estimation from Keypoints 42
3.5 Head Pose Estimation by Measurement Fusion from Facial Landmark Detection and Keypoint Tracking 49
3.6 Head Pose Estimation by Fusion of Global and Local Motion 54
3.7 Experiments and Results 59
3.8 Concluding remarks 69

Publications

This Chapter covers the work published in:

- Jilliam M. Díaz Barros, F. Garcia, B. Mirbach and D. Stricker, "Real-Time Monocular 6-DOF Head Pose Estimation From Salient 2D Points", in IEEE International Conference on Image Processing (ICIP), Beijing, China, pp. 121-125. 2017.
- Jilliam M. Díaz Barros, F. Garcia, B. Mirbach, K. Varanasi, and D. Stricker. "Combined Framework for Real-time Head Pose Estimation using Facial Landmark Detection and Salient Feature Tracking", in International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Madeira, Portugal, pp. 123-133. 2018.
- Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker. "Fusion of Keypoint Tracking and Facial Landmark Detection for Real-Time Head Pose Estimation", in IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, pp. 2028-2037. 2018.
- Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, D. Stricker, "Real-Time Head Pose Estimation by Tracking and Detection of Keypoints and Facial Landmarks", in Computer Vision, Imaging and Computer Graphics Theory and Applications. VISIGRAPP. Communications in Computer and Information Science (CCIS), vol 997. Springer, Cham. 2018.

3.1 Introduction

Head pose estimation (HPE) denotes the task of calculating the position and orientation of a person's head, with respect to a given coordinate system. The estimation is generally performed for 6 degrees of freedom (DoF), given by 3 rotation angles and 3 translation parameters. HPE is an essential step to address well-known research topics such as face recognition [93], facial expression recognition [94, 95], face alignment [96] and gaze estimation [97, 98], with applications in human-computer interaction, driver monitoring [99, 100] and augmented reality [101].

In particular, monitoring the attention of human users by automatically tracking their head pose is of fundamental importance in many areas, including the detection of the engagement level in online learning, assisting patients with disability and behavior understanding of children with autism. It is also relevant for assessing human performance in areas which warrant critical attention: factory monitoring, production pipelines, security, among others. Indeed, humans naturally suffer from fatigue due to long working hours or information overload, where they show symptoms of errant head movements. Furthermore, it has been demonstrated that the combination of head pose and eye location provides a reference framework to estimate the gaze, from which is possible to determine the level of attention or drowsiness, for instance, of a vehicle driver [102, 103].

The extensive literature on HPE [104], and its use in a wide variety of applications, evidence the importance of this topic. In the recent past, several HPE approaches have been proposed using RGB to depth data, as well as thermal infrared (IR) cameras, in monocular and stereo setups [105, 100, 106, 107, 108, 102, 109, 110]. However, computer vision systems for HPE still face serious challenges in deployment under realistic scenarios. Firstly, such a HPE system has to work for a wide range of human users, without any additional calibration step. Secondly, it should be non-intrusive, requiring no additional sensors or lighting. Thirdly, it needs to be lightweight, with no power demanding devices. This currently rules out computationally intensive models that require GPUs. Fourthly, the system has to be extremely fast in order to assess the situation in real time. Fifthly, it should be able to launch the estimation task without pose constraints, *e.g.*, that the person should be facing the camera. Finally, the system has to work for a wide range of head pose orientations, face occlusions, and external lighting conditions. Hence, a fast and efficient HPE system remains an active area of research.

Considering the previous constraints, this Chapter is focused on estimating the head pose from intensity images, where the gaze could be recovered in a follow-up work, using an efficient system for eye tracking [111, 112, 113]. The advantage of not using additional sensors or depth images is that our method is suitable for applications where only 2D images are available. Herein, we introduce three approaches for HPE in a monocular setup:

1. The first HPE method employs 2D random feature points, or keypoints, to estimate the pose (see Section 3.4). The features are tracked using optical flow and the respective 3D points are computed using a geometric model. The pose is then recovered by minimizing the reprojection error of the 3D features and the 2D correspondences.

2. The second HPE approach combines the head pose estimated from facial landmarks with

the head pose estimated from keypoints, as described in Section 3.5. 3D points for both types of features are also recovered from the intersection on a simple geometric head model. The estimated poses are integrated into a linear Kalman Filter as new measurements in the correction stage.

3. The third HPE method introduces a new framework to fuse the estimated poses from facial landmarks and keypoints (see Section 3.6). In this case, both pipelines operate in parallel: in the keypoint-based HPE method, the pose is computed from 2D tracked keypoints and using a simple geometric model, while in the facial-landmark-based HPE method, the pose is estimated from the detected 2D facial landmarks and 3D facial landmarks refined through triangulation. The latter 3D facial landmarks are extracted from a reference head mesh and used through the entire video sequence. These pipelines are then fused using a tailored Kalman Filter, where the head pose estimated from keypoints are used to update the state transition matrix at the prediction stage, and the pose computed from facial landmarks are used as a new measurement at the correction stage.

The main contributions of this Chapter are:

- A new pipeline for head pose estimation based on 2D keypoints. This approach defines the region of interest (ROI) for feature detection, by projecting the visible surface of a geometric shape modeling the head. Therefore, a face detector is not required in every frame, enabling real-time performance in consumer devices.
- A weighting scheme based on the distance transform, to identify those 2D features that lie close to the edge of the ROI and are therefore prone to self-occlusion or likely to belong to the background.
- A novel framework for HPE combining the relative strengths of facial landmark detection and keypoint tracking, through measurement fusion with a dedicated Kalman filter.
- A new approach for refined initialization of the head pose when it is different to frontal, using the 2D facial landmarks detected in the input image and a set of 3D facial landmarks, when the pose is frontal.
- A novel head pose estimation technique, that combines a local motion estimated by keypoints tracking, with a global head pose estimated from facial landmark alignment. The local motion is integrated at the prediction step of the Kalman Filter, while the global motion is included for the correction step.
- A facial-landmark-based head pose estimation technique, where the 3D facial landmarks are refined over time. The refinement is performed by triangulating 2D facial landmarks extracted from different frames. The 3D points obtained from triangulation are recursively added to a Kalman Filter, where for every new measurement, the observation noise covariance is updated with the covariance matrix computed during the triangulation.

We show that our approaches can perform in real time and are able to estimate the pose also for extreme head rotations.

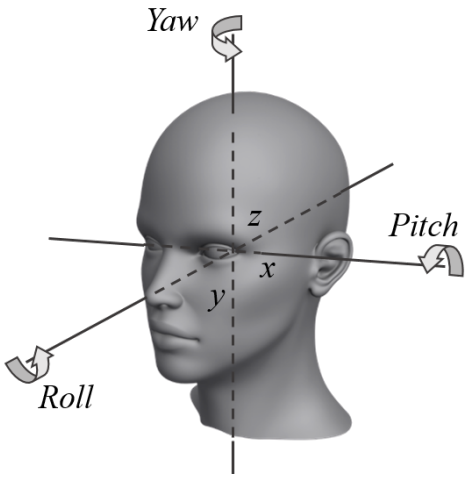


Figure 3.1: We define the head pose through 3 DoF for its orientation (*pitch*, *yaw*, and *roll* angles) plus 3 DoF for its position (*x*, *y*, and *z* translations) with respect to the view of a monocular system.

3.2 Problem Formulation

HPE aims to estimate the 3D position and orientation of the head with respect to a known coordinate system. In computer vision, the origin of this coordinate system is usually given at the optical center of the camera **C**.

In this Chapter, we model the head as a disembodied rigid object [104], where the pose is represented with a transformation, composed of a rotation **R** and a translation **t**. This transformation maps each point from the model coordinate system to the camera coordinate system. Therefore, the pose of every 3D point **P** in the head is updated following a rigid transformation. **R** can also be denoted by the rotation angles $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$ with respect to the X, Y and Z axes of the coordinate system. ω_x, ω_y, and ω_z are usually termed as pitch, yaw and roll angles, respectively, as illustrated in Figure 3.1.

3.3 Related Work

HPE is an extensively researched topic in computer vision, spanning in the literature approaches based on diverse technologies. Methods based on depth data have been proposed in [114, 107, 115, 116, 100, 117], as well as combined techniques using both RGB and depth data [118, 119], or IR and depth data [120]. Video-based methods are comprised in the survey conducted in [104].

HPE approaches can be classified in three categories [100]: feature-based, appearance-based and 3D head model registration approaches. It should be noted that some methods might fall in more than one category.

Feature-based approaches

These approaches are characterized for utilizing facial features, including facial landmarks or face parts for HPE.

Kumano *et al*. [95] presented an approach for pose-invariant facial expression recognition, which estimates the pose and facial expression simultaneously. This method uses a face model based on a variable-intensity template with a particle filter, which depends on keypoints extracted in pre-defined facial parts, such as eyebrows, nose and lips. In [97, 98], Valenti *et al*. introduced a pipeline for simultaneous HPE and eye tracking. Both tasks are calculated and updated based upon a crossed feedback mechanism, which compensates the estimated values and allows to re-initialize the head pose tracker. The eyes are located using an isophote-based method that exploits the semi-circular patterns inside the eye and the head is modeled using a cylindrical head mode (CHM). Jang and Kanade [121, 122] designed an user-specific head tracking framework using a CHM. A Kalman filter is used to combine the motion estimated between consecutive frames and the pose retrieved from a database, which maps SIFT feature points to different head poses. Drouard *et al*. [123] used a Gaussian mixture of linear regression models to map HOG features extracted on a face region to 3D head poses. The approach was suitable for HPE from low-resolution data.

Asteriadis *et al*. [124] performed the estimation task using a facial-feature tracker with Distance Vector Fields (DVF). Kun *et al*. [103] estimated 2D keypoints and their projection onto an ellipsoidal head model (EHM), via ray tracing. The face is detected in each frame, in order to limit the area for feature extraction and correspondences estimation. The head pose is computed by solving the transformation between these 3D features and their corresponding 2D features in the next frame, obtained through the KLT tracker [36]. Two different approaches were evaluated, namely the Perspective-n-Point problem and Pose from Orthography and Scaling by Iterations (POSIT) [125]. Wang *et al*. [126] introduced head tracking based on invariant keypoint descriptors and a 3D head model. The 3D pose was recovered by minimizing the reprojection error, where the 2D correspondences were computed by means of optical flow. This method required user-dependent manual initialization in a pre-processing step. In [127], Prasad and Aravind computed the pose using POSIT from the 3D-2D correspondences from a parameterized 3D face mask and SIFT feature points. Larumbe *et al*. [128] exploit 2D landmarks computed using Random Cascaded-Regression Copse (R-CR-C) [129] and a 3D head model, to estimate the pose via POSIT. On the other hand, Vicente *et al*. [99] proposed an approach oriented to driver's activities monitoring. Facial landmarks were detected and tracked using a face alignment scheme with parameterized appearance models (PAM). The head pose was estimated by minimizing the reprojection error, in this case from a 3D deformable head model and the tracked facial landmarks. Yin and Yang [130] introduced a real-time HPE method optimized for on-board computers. They use a pixel intensity binary test for face detection, with pose regression along with local binary feature for face alignment. The pose is retrieved by solving the 2D-3D correspondences using a rigid face model. Wu *et al*. in [131] presented a pipeline for simultaneous facial landmark detection, HPE and deformation estimation using a cascade iterative procedure augmented with model-based HPE. Similarly, Gou *et al*. [132] proposed a Coupled Cascade Regression (CCR) framework for simultaneous facial landmark detection and HPE.

Appearance-based approaches

These methods use one or multiple classifiers to estimate the head pose, based on visual fea-

tures of the face appearance. They usually have a higher performance on low-resolution face images [133, 123].

A pose-adaptive Constrained Local Model (CLM) framework was proposed by Zamuner *et al*. in [134], which integrates active appearance model (AAM) in order to combine the pose from multiple classifiers and handle large head rotations. Fanelli *et al*. in [114], estimated the head pose as well as facial features with random regression forests from depth data. Patches from different parts of the face were used to recover the pose through a voting scheme. A large dataset with annotated data was necessary for training. Tulyakov *et al*. [135] presented a robust user-specific-model scheme for extreme pose variations from a depth camera. The scheme combines template-matching-based tracking with a frame-by-frame decision-tree-based estimator. Derkach *et al*. [117] introduced a system intended for HPE from depth input data, which integrates three different HPE approaches, two based on landmark detection and one on a dictionary-based method for extreme head poses. Papazov *et al*. [116] introduced triangular surface patch (TSP) descriptors for HPE from depth data. The input head data is divided into several patches and each one is matched with TSP descriptors from synthetic head models stored in a database. A voting scheme is then implemented from the patches to recover the head pose.

Ahn *et al*. [133] proposed a deep-learning-based approach for HPE from RGB images. A particle filter is included to refine and increase stability on the estimated pose. Liu *et al*. [136] presented a pipeline for HPE from RGB images using convolutional neural networks (CNN), where HPE was formulated as a regression problem. The network is trained using a large synthetic dataset obtained from rendered 3D head models. Both [133] and [136] required a GPU to achieve real-time performance. Borghi *et al*. [100] introduced a real-time deep-learning approach for upper-body and HPE from depth images. The proposed regression neural network, POSEidon, integrates depth with motion features and appearance. On the other hand, Schwarz *et al*. [120] presented a deep learning method for HPE under driving conditions, by fusing IR and depth data with cross-stitch units.

3D head model registration approaches

These methods define and register a 3D head model based on the measured input data.

La Cascia *et al*. [37] presented a 3D head tracking approach based on registration of texture map images onto a CHM. An illumination correction term is introduced to increase robustness to light changes among consecutive image frames. HPE is computed by means of least squares. Zhang and Kambhamettu [137] proposed an extended super-quadric model (ESQ) to track the head with motion regularization. Although the method is robust to self-occlusion and non-rigid motion, it is not suited for real-time applications. Choi and Kim [94] introduced a method for HPE based on templates, which combines a particle filter with an EHM. The 3D motion parameters are initialized using AAM and an online appearance model (OAM) is integrated at the observation model. Sung *et al*. [138] combined AAM with a CHM in order to extent the working range of the head motion. AAM is used to initialize the global motion parameters of the CHM. Later, these parameters are employed to update the AAM during the tracking step. An and Chung [93] proposed a method for face recognition where the pose is estimated us-

ing a parameterized EHM. HPE is formulated as a linear system assuming a rigid body motion under perspective projection and it is solved using least squares. From this pose, face images were registered into frontal views, facilitating their recognition. Xiao *et al*. [139] proposed an approach using a CHM, where motion is computed by minimizing an objective function and assuming a perspective projection. Dynamic templates along with a pixel weighting scheme are introduced to track the head pose. In [140], Basu *et al*. presented a framework for rigid motion estimation employing an EHM. Tracking was performed through motion regularization; optical flow was used to compute the model flow and the motion error was minimized using simplex gradient descent.

Meyer *et al*. [115] combined particle swarm optimization (PSO) and the iterative closest point (ICP) algorithm to register a 3D morphable model (3DMM) to a measured depth face. The method runs at 6FPS on a GPU. Yu *et al*. [141] extended the previous work with an online 3D reconstruction of the full head, to handle extreme head rotations. Ghiass *et al*. [142] presented an approach for HPE from RGB-D data. The pose estimation is performed through a fitting process employing a 3DMM. Jeni *et al*. [75] presented an approach for 3D registration of a dense face mesh from 2D images, through a cascade regression framework. The framework was trained using a dataset of high-resolution 3D face scans. The head pose and the 3D shape are iteratively refined, by registering a part-based 3DMM on 2D facial landmarks. Tan *et al*. [109] used RGB-D data to regress the 3D head pose using random forest in a temporal tracking approach. An offline stage is used to learn a 3D head shape from CAD models, while the online stage refines subject-specific features through a tracking scheme.

Morency *et al*. [143] presented a probabilistic scheme named Generalized Adaptive View-based Appearance Model (GAVAM), using an EHM. The pose is extracted by solving a linear system with normal flow constraint (NFC). Baltrusaitis *et al*. [118] proposed an extension of GAVAM, by combining a head pose tracker with a 3D CLM. The framework uses both depth data and intensity information to track facial feature points, but it was also tested using only intensity images. Saragih *et al*. [144] introduced a method for HPE by fitting a deformable model, using an optimization strategy through a non-parametric representation of the likelihood maps of landmarks locations.

The accuracy of feature-based HPE methods that rely on face parts or landmarks, *e.g.*, nose tip or eyes corners, depends on the precision of the estimated features. Since they strongly depend on their detection in every frame, misalignment or missing detection in some frames might lead to erroneous estimations. Hence, these methods might be sensitive to extreme head poses, partial occlusions, facial expressions and low resolution images. Tracking-based methods target such cases by means of a motion model. However, their robustness to estimate an initial HPE when the head is not frontal is not clear [117]. This is also the case for 3D-model-based methods, where a manual initialization is usually required [100]. On the other hand, appearance-based pipelines are robust to large head rotation. Nonetheless, the estimated pose in many of these methods is retrieved from a classifier which uses discrete head poses for training and thus assigns the pose to a specific range, instead of continuous estimation.

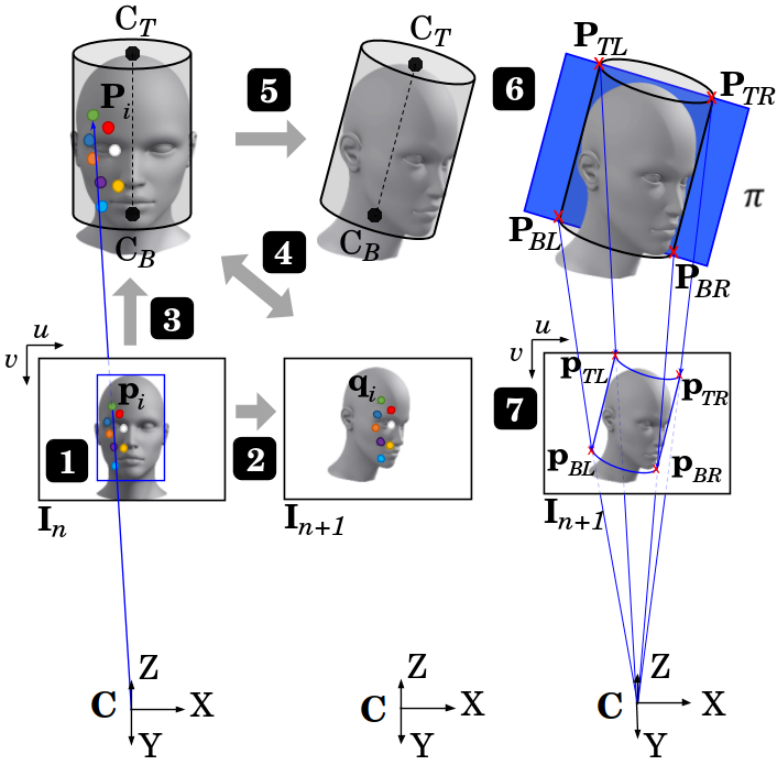


Figure 3.2: Proposed HPE pipeline based on keypoints.

3.4 Head Pose Estimation from Keypoints

In this Section, we estimate the head pose and provide its position and orientation, based on keypoints detected in every frame in a monocular system. Similarly to [94, 139], we propose to use a simple geometric shape to model the head, due to its ease of initialization in comparison to precise and flexible head models [104] or parameterized face models [126, 127]. Indeed, a CHM or EHM only require the estimation of fewer facial landmarks to be initialized. Nonetheless, HPE approaches based on geometric models and intended for monocular systems require precise detection of facial landmarks, or rely on frame to frame face detection in order to continuously update the model. On the contrary, we propose a HPE pipeline that once initialized estimates the head pose with no need of face detection, neither facial landmarks. As a result, the proposed approach is robust to large head rotations regardless of the facial expression and/or self-occlusions.

The experimental evaluation shows that the performance of this approach achieves, and in some cases improves the state of the art performance with a major advantage of not requiring facial landmarks (except for initialization). As a result, our method also applies to real scenarios in which facial landmarks-based methods fail due to occlusion.

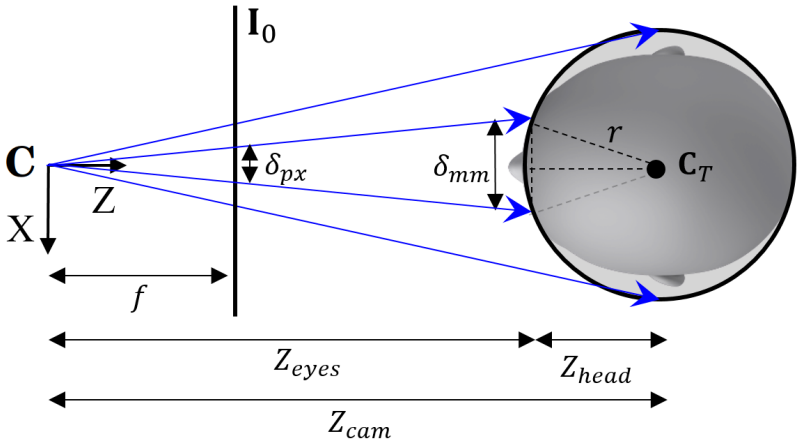


Figure 3.3: Location of the CHM/EHM in the 3D space.

3.4.1 Overview of the Proposed HPE Framework

We introduce a real-time and robust approach to estimate the 3D head pose, depicted in Figure 3.2. Our approach leverages a simple geometric shape to model the head, initialized using facial landmarks, *i.e.*, eye corners. First, 2D keypoints are detected within the region defined by the projection of the visible surface of the geometric head model onto the image, and projected back to the model to generate the corresponding 3D keypoints. Optical flow is used to find the respective 2D correspondences in the next video frame. Assuming that the monocular system is calibrated, it is then possible to solve the Perspective-n-Point (PnP) problem to estimate the head pose. To that end, the set of 3D keypoints on the geometric model surface and their respective 2D correspondences from optical flow in the consecutive frame are used. This process is repeated in the following frames, while updating the pose of the geometric model and the projection of its visible surface.

3.4.2 System Initialization

An essential step for the initialization of the proposed system is to adjust the geometric head model to the head of the user. To that end, the user's face dimensions as well as the distance between the eye's centers are needed. Herein, we use the Viola-Jones face detection method [64], detailed in Section 2.4.1, similarly to related works [118, 142, 15]. However, alternative techniques such as appearance template methods [104] might also work, as soon as the centers of the eyes are also provided.

In the following, we assume a calibrated camera with known intrinsic parameters, following Section 2.1.2, and that the user is facing the camera in the first frame. As shown in Figure 3.3, we define the distance between the optical center of the camera **C** and the baseline of the eyes in the geometric model as

$$Z_{eyes} = f \cdot \frac{\delta_{mm}}{\delta_{px}}, \tag{3.1}$$

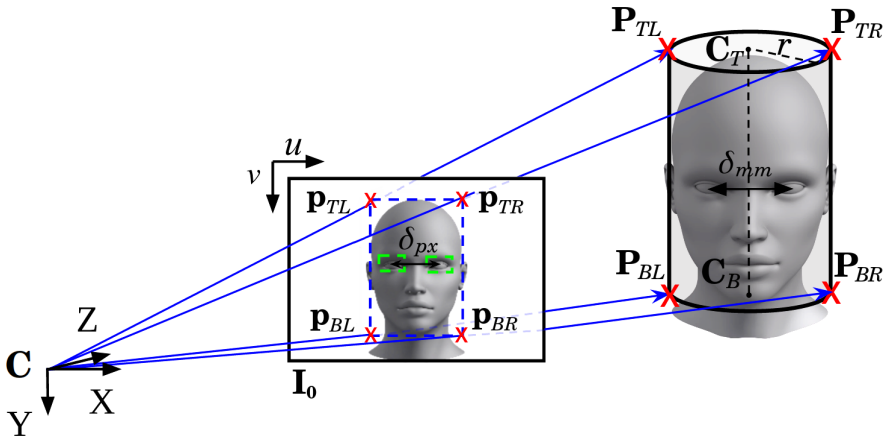


Figure 3.4: Initialization of the CHM to fit the user's head.

with f being the focal length of the camera, δ_{mm} the mean interpupillary distance in the model coordinate system and δ_{px} the estimated distance between the eye's centers in the image space (in pixels). The distance between the eyes is set to 60 mm based on anthropometric measurements, which average δ_{mm} to 64.7 mm for men and 62.3 mm for women [145, 146].

The distance Z_{cam} from **C** to the vertical axis of the geometric model results from the sum of Z_{eyes} and Z_{head}, the distance from the eyes' baseline to the axis of the model, *i.e.*,

$$Z_{cam} = Z_{eyes} + Z_{head}, \tag{3.2}$$

as shown in Figure 3.3. Z_{head} is computed from the radius r of the geometric model as:

$$Z_{head} = \sqrt{r^2 - (\delta_{mm}/2)^2}. \tag{3.3}$$

In the CHM, this radius is calculated as

$$r = \frac{1}{2}|\mathbf{p}_{TR} - \mathbf{p}_{TL}| \cdot \frac{\delta_{mm}}{\delta_{px}}, \tag{3.4}$$

with $\mathbf{p}_{TR}$ and $\mathbf{p}_{TL}$ being the top right and top left corners of the detected face bounding box (see Figure 3.4). In the EHM, r is approximated to the radius of the ellipsoid in the Z axis, r_z, as depicted in Figure 3.5. The radii r_x and r_z are set equal to half of the width of the detected face, and r_y to half of the height of the face, having as a result a prolate ellipsoid, or spheroid, as shown in Figure 3.5. Therefore, r_x and r_z are computed using Eq. (3.4) and r_y is given by

$$r_y = \frac{1}{2}|\mathbf{p}_{TR} - \mathbf{p}_{BR}| \cdot \frac{\delta_{mm}}{\delta_{px}}, \tag{3.5}$$

with $\mathbf{p}_{BR}$ being the bottom right corner of the bounding box. Similarly, the height of the CHM is adjusted to the user's head by

$$h = |\mathbf{p}_{TR} - \mathbf{p}_{BR}| \cdot \frac{\delta_{mm}}{\delta_{px}}. \tag{3.6}$$

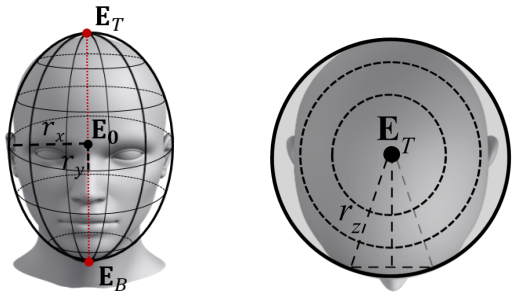


Figure 3.5: Ellipsoidal head model. Left: Frontal view. Right: Top view.



Figure 3.6: Left: Keypoints detected in a single frame using FAST-ER [35]. Right: Distance transform on given ROI.

Once Z_{cam} is estimated, we compute the corners of the face bounding box in the 3D space, *i.e.*, $\{\mathbf{P}_{TL}, \mathbf{P}_{TR}, \mathbf{P}_{BL}, \mathbf{P}_{BR}\}$, together with the centers of the top and bottom bases of the geometric model, *i.e.*, $\mathbf{C}_T$ and $\mathbf{C}_B$ for the CHM and $\mathbf{E}_T$ and $\mathbf{E}_B$ for the EHM. Since we assume that the user is facing the camera, both rotation angles $(\omega_x, \omega_y, \omega_z)$ and initial translation values (t_x, t_y, t_z) are set to zero.

3.4.3 Robust 2D Keypoint Detection and Matching

Geometric-based HPE approaches require either facial landmarks like nose tip, eye's and/or lips' corners to be detected frame by frame, or sample and project onto the image plane the visible surface of the CHM/EHM in a grid-like structure for 2D feature detection [98]. On the contrary, we propose to detect 2D keypoints as 2D features within the region of interest (ROI) (see Step 1 in Figure 3.2). This ROI is defined by the projection of the visible surface of the geometric model onto the image plane, as further explained in Section 3.4.6. We denote the set of 2D keypoints by $\{\mathbf{p}_K\}_{i=1}^m$ and their corresponding set of 3D keypoints by $\{\mathbf{P}_K\}_{i=1}^m$. The number of keypoints per frame, m, is not fixed since it depends on the number of feature points tracked between two consecutive frames.

In every frame, we extract 2D keypoints in the area of the head (see left image in Figure

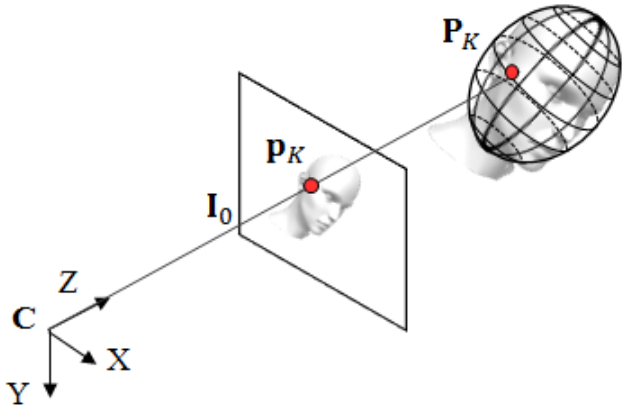


Figure 3.7: Computation of 3D keypoints.

3.6) and find their correspondences from the previous frame. These keypoints can be located in the area of the face, but also on the back side or top of the head, for large head rotations. The 2D keypoints are detected using the Features from Accelerated Segment Test algorithm with Enhanced Repeatability, FAST-ER, described in Section 2.1.4. Aside from being very well suited for real-time applications, it provides accurate corners.

In our approach, each feature is weighted according to its location within the ROI. That is, we have introduced a feature weighting scheme that enables us to discard outliers and/or non-reliable keypoints and thus, to only consider robust features for HPE. This weighting scheme results from applying the distance transform onto the ROI, as detailed in Section 2.1.6, where each pixel within the ROI has a normalized weight related to the Euclidean distance to the closest boundary (see right image in Figure 3.6). Therefore, larger weights are given to those keypoints located in the center of the ROI than to those near the ROI boundaries. Our experiments in Section 3.7.4 show that removing low-weighted features leads to better performance within the HPE process.

Given the set of keypoints extracted from the previous frame $\{\mathbf{p}_K\}_{i=1}^{m}$, we find the 2D correspondences $\{\mathbf{q}_K\}_{i=1}^{m}$ at the new input image using optical flow (see Step 2 in Figure 3.2). To that end, we use the pyramidal Lucas-Kanade feature tracker [147], detailed in Section 2.1.4.

3.4.4 3D Keypoint Computation

As depicted in Figure 3.2 (Step 3) and in Figure 3.7, each 3D keypoint $\mathbf{P}_K$ results from the intersection of the projection line $\boldsymbol{l}$ on the geometric head model. This line passes through the optical center of the camera $\mathbf{C}$ and the corresponding 2D keypoint $\mathbf{p}_K$ in the image plane $\mathbf{I_0}$. The orientation, position and dimension of the geometric model on the 3D space are known in advance, as detailed in Section 3.4.2.

The equation of the projection line is given by

$$\boldsymbol{l} = \mathbf{C} + \lambda \mathbf{d}, \tag{3.7}$$

where $\mathbf{d}$ represents a line parallel to $\boldsymbol{l}$ and λ is a scalar retrieved from the quadratic equation of

the geometric model. In the case of the CHM, the equation is given by

$$|\mathbf{m}|^2 \ \lambda^2 + 2\,(\mathbf{m}\cdot\mathbf{n})\,\lambda + |\mathbf{n}|^2 - r^2 = 0, \tag{3.8}$$

with $\mathbf{m}$ and $\mathbf{n}$ defined as

$$\mathbf{m} = \mathbf{d} - \frac{\mathbf{d}\cdot\mathbf{g}}{|\mathbf{g}|}\,\mathbf{g} \tag{3.9}$$

$$\mathbf{n} = (\mathbf{C} - \mathbf{C}_B) - \frac{(\mathbf{C} - \mathbf{C}_B)\cdot\mathbf{g}}{|\mathbf{g}|}\,\mathbf{g}. \tag{3.10}$$

$\mathbf{g}$ is a vector parallel to the axis of the cylinder, $\mathbf{C}_B$ is the center of its lower base, and r is the radius.

For the ellipsoid, the quadratic equation is given by

$$|\mathbf{a}|^2 \ \lambda^2 + 2\,(\mathbf{a}\cdot\mathbf{b})\,\lambda + |\mathbf{b}|^2 - 1 = 0. \tag{3.11}$$

Given an ellipsoid with radii $\{r_x, r_y, r_z\}$, rotation matrix $\mathbf{R}$ and having its center at $\mathbf{E_0}$, $\mathbf{a}$ and $\mathbf{b}$ are defined as

$$\mathbf{a} = \mathbf{G}\mathbf{R}^T\mathbf{d} \tag{3.12}$$

$$\mathbf{b} = \mathbf{G}\mathbf{R}^T(\mathbf{C} - \mathbf{E_0}), \tag{3.13}$$

with $\mathbf{G}$ being a 3×3 diagonal matrix of the inverse of the ellipsoid radii, *i.e.*, $\{\frac{1}{r_x}, \frac{1}{r_y}, \frac{1}{r_z}\}$.

3.4.5 Pose Estimation from Feature Points

The set of 3D keypoints, $\{\mathbf{P}_K\}_{i=1}^m$, along with the 2D correspondences detected in the following frame, $\{\mathbf{q}_K\}_{i=1}^m$, are used to solve the PnP problem and thus, to estimate the head pose (see Step 4 in Figure 3.2). The pose is computed by minimizing the error between the reprojection of the 3D keypoints onto the image plane and their respective 2D correspondences, following Section 2.1.5.

From the resulting translation $\mathbf{t}$ and rotation $\mathbf{R}$, we update the geometric model (see Step 5 in Figure 3.2) following:

$$\mathbf{P}'_K = \mathbf{R}\cdot\mathbf{P}_K + \mathbf{t}. \tag{3.14}$$

However, we do not update the pose of the corners defining the face bounding box in the 3D space, *i.e.*, $\{\mathbf{P}_{TL}, \mathbf{P}_{TR}, \mathbf{P}_{BL}, \mathbf{P}_{BR}\}$ in a same way, as they need to be recalculated to define the new ROI for feature detection in the following frame.

3.4.6 ROI for 2D Keypoint Detection

As introduced in Section 3.4.3, the ROI for 2D keypoint detection is defined by the projection of the visible surface of the geometric model onto the image, regardless of the orientation of the head (see Step 6 in Figure 3.2). As mentioned earlier, this area might not only be on the face, but also on top or on the back side of the head. However, this is not a trivial task. Alternative geometric-based methods simply keep track of the 3D face bounding box, *i.e.*, $\{\mathbf{P}_{TL}, \mathbf{P}_{TR}, \mathbf{P}_{BL}, \mathbf{P}_{BR}\}$ and define the ROI by its projection onto the image plane. Even so, these methods fail for

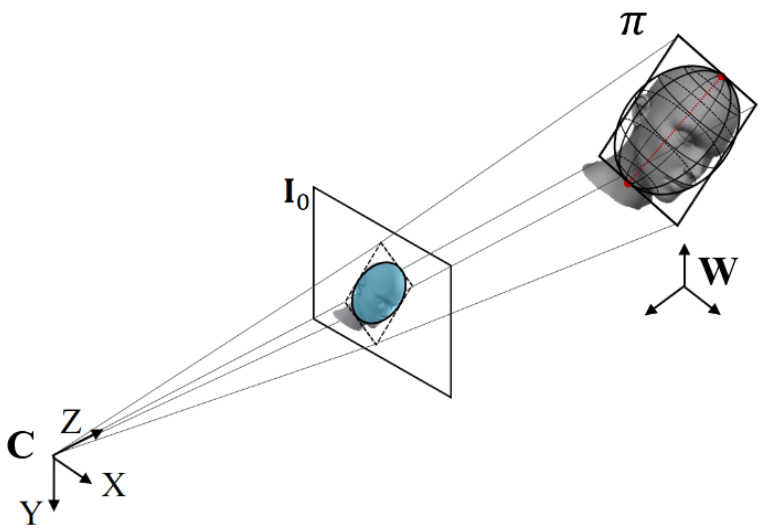


Figure 3.8: ROI computation for 2D keypoint detection.

large head rotations, where the corners of the face bounding box result partially visible or completely occluded. Other authors introduce a grid-like or mesh structure along the geometric model surface, from which the projection of each vertex is used for feature detection. Nevertheless, a dense grid is needed to ensure robust 2D features detection, which is not practical if real-time is required. Some methods propose to extract keypoints from the area given by a face detection algorithm [103]. Besides being time consuming, this approach would fail when the face is not detected, *i.e.*, for large head rotations.

Herein, we present an alternative solution to continuously update the area for keypoint detection, by projecting the 3D geometric model onto the image plane in every frame, as shown in Figure 3.8. In the case of the CHM, only the four corners defining the visible surface of the CHM along with the curvature of the CHM's bases are needed to define the ROI. To do so, we first define a plane π which normal vector results from the cross product of a vector parallel to the X-axis of the camera coordinate system and the vector defined between the two centers of the CHM bases, *i.e.*, $\mathbf{C}_T$ and $\mathbf{C}_B$. The visible surface of the CHM is given by the furthermost intersected points between the CHM and π, *i.e.*, $\{\mathbf{P'}_{TL}, \mathbf{P'}_{TR}, \mathbf{P'}_{BL}, \mathbf{P'}_{BR}\}$, whereas the new ROI results from filling the polygon defined by their projection along with the curvature of the CHM's bases (see Step 7 in Figure 3.2).

Similarly for the EHM, we first estimate the plane π parallel to the image plane and to the vertical axis of the ellipsoid, that divides it in two parts, *i.e.*

$$\pi \in \mathbb{R}^3 \mid \pi \parallel \mathbf{I_0} \text{ and } \pi \parallel \overline{\mathbf{E}_T\mathbf{E}_B}. \tag{3.15}$$

Secondly, we find the elliptical surface that results from the intersection of π and the ellipsoid. Finally, this surface is projected in the image plane, assuming a perspective camera model. For the EHM, the projection corresponds to an elliptical ROI that covers the area of the face.

3.5 Head Pose Estimation by Measurement Fusion from Facial Landmark Detection and Keypoint Tracking

Facial-landmark-based approaches can reach high performance on HPE, as reported in [118, 75]. However, these approaches might fail as soon as the face is partially or totally occluded. On the other side, the use of keypoints provides a solution in case the face detector or alignment fail, *e.g.*, under extreme head poses or occlusion. Nonetheless, they might suffer from cumulative drifting error along frames, a common issue in keypoint-based HPE approaches [121, 122].

We leverage both types of facial features, by exploiting the high accuracy of the facial landmarks while handling extreme head poses and (self-)occlusion by means of the keypoints. In this Section, we introduce a novel HPE framework that combines a tracking-by-detection scheme using facial landmarks with the tracking of keypoints within the head region. In this framework, facial landmarks are detected using a machine learning pipeline that effectively provides the corresponding 2D pixel locations of these points. We additionally integrate keypoints tracked on the acquired video frame sequences, following Section 3.4. These features can be person-specific facial birthmarks, wrinkles, hair or accessories that can provide alternative source of information for HPE.

The proposed approach targets some limitations of the pipeline in Section 3.4, such as the need of a frontal face to initialize the system, and drifting in long sequences. The results of this pipeline are comparable to those from the related literature, as shown in Section 3.7, with the advantage of performing in real time and in real world scenarios.

3.5.1 Overview of the Proposed HPE Framework

The backbone of the proposed HPE approach, shown in Figure 3.9, is given by the detection of 2D facial landmarks (purple box in Figure 3.9) and the tracking of keypoints (green box) on the input face. The landmarks are aligned after detecting the face in the given frame, while the keypoints are extracted in the ROI computed from the projection of a simple 3D geometric shape, either an EHM or a CHM, onto the image plane (orange box). Afterwards, each set of feature points is back projected onto the geometric model via ray-tracing (yellow box), and thus hypothesized in 3D space. Then, the respective 2D correspondences are extracted on the next frame, applying face alignment for the first set and optical flow for the second. Consequently, two head poses are estimated, by minimizing the facial features reprojection error with the 3D-2D correspondences, each with 6 DoF. The resulting head pose is finally computed by combining both poses using a dedicated Kalman Filter (blue box in Figure 3.9).

3.5.2 Facial Landmark Detection

We refer to facial landmarks as a specific set of feature points in the area of the face (see Section 2.4.2). The landmarks are aligned in a face detected using Max-Margin Object Detection (MMOD) [67]. To estimate these landmarks, we propound to use regression trees as in [148], from a sparse subset of intensity values indexed to an initial estimate of the shape. By doing so,

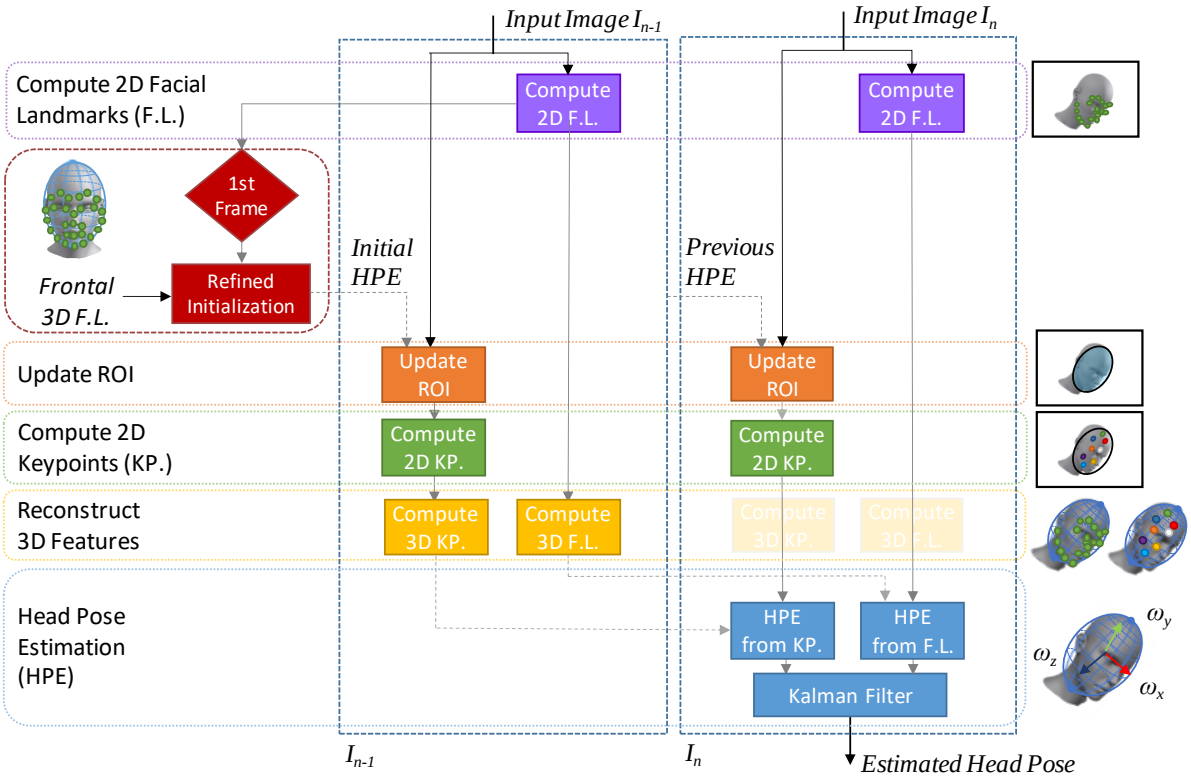


Figure 3.9: Pipeline of the proposed HPE approach.

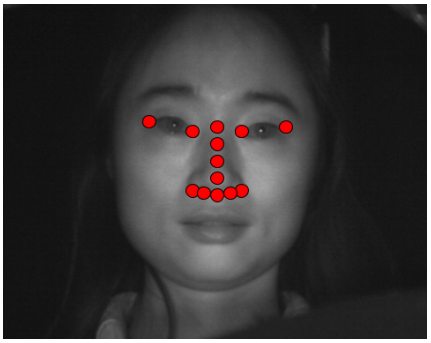


Figure 3.10: Selected facial landmarks in 2D.

we are able to retrieve in real-time the set of facial landmarks displayed in Figure 2.17, and the subsequent correspondences in every new frame, as long as the face is detected.

Since the head is modeled as a rigid body, we chose a set of fiducial features that are robust to non-rigid motions, including facial expressions and blinking. This set of landmarks $\{\mathbf{p}_F\}_{i=1}^{n}$ is composed of n =13 features, which encompasses the corners of the eyes, points in the nasal bridge and points around the nostrils, as shown in Figure 3.10.

However, a problem might arise for extreme head rotations or (self-)occlusions, since the pose cannot be retrieved in case the facial landmarks are not available. We compensate the limitation of facial-landmarks-based approaches, by integrating a set of keypoints which can track the pose of the head, even when the facial landmarks are not available.

3.5.3 Keypoint Detection and Tracking

As depicted in Figure 3.9, one of the cornerstone of our HPE framework is the use of keypoints, which extend the working range of our approach to extreme head poses. These features are extracted only in a region of interest (ROI) in the input image, that corresponds to the head of the user. The ROI is calculated from the projection of the 3D geometric model onto the 2D image plane, as shown in Figure 3.8 and detailed in Section 3.4.6. This ROI is computed in each frame as soon as a new head pose is recovered and the geometric model is updated (orange box in Figure 3.9).

Following Section 3.4.3, we use FAST-ER for the detection of keypoints $\{\mathbf{p}_K\}_{i=1}^{m}$ and the iterative Lucas-Kanade feature tracker with pyramids [147], to track m keypoints and find the correspondences $\{\mathbf{q}_K\}_{i=1}^{m}$. We additionally weight each feature according to its location within the ROI, based on the distance transform. That is, keypoints further away from the ROI border are treated as more reliable and with greater weight, enabling us to reject outliers and/or non-reliable keypoints.

3.5.4 System Initialization

We follow a similar procedure from Section 3.4.2 to initialize the head model. However, instead of using the bounding boxes of the face and eyes provided by the Viola-Jones face detector, we estimate the head dimension and eyes' location from the detected landmarks (see Section 3.5.2). Indeed, the head dimension is linked to the width and height of the detected face. Therefore, in this case the interpupillary distance in pixels δ_{px} is computed by estimating the midpoint of the outline landmarks in each eye and measuring the distance between them.

The 2D bounding box of the face, enclosed by points $\{\mathbf{p}_{TL}, \mathbf{p}_{TR}, \mathbf{p}_{BL}, \mathbf{p}_{BR}\}$, is computed from the outer facial landmarks. We use this box to calculate the width and height of the geometric model. However, as can be observed in Figure 2.17, these landmarks do not cover the upper part of the head. In this case, we set the height of the bounding box as 1.2 of the width of the face, following related works [73]. Once the dimensions of the face are computed, we proceed to initialize the 3D geometric model similarly to Section 3.4.2.

Since the EHM has a better representation of the human head than the CHM, in the following

only the EHM is used as the geometric head model.

3.5.5 Sparse 3D Reconstruction

The 3D features points $\mathbf{P}_i$ from each set of 2D features, facial landmarks $\{\mathbf{p}_F\}_{i=1}^{n}$ and keypoints $\{\mathbf{p}_K\}_{i=1}^{m}$, are computed by means of ray tracing, as previously detailed in Section 3.4.4. As a result, we obtain two set of 3D features, the first one with the 3D facial landmarks $\{\mathbf{P}_F\}_{i=1}^{n}$ and the second set with the 3D keypoints $\{\mathbf{P}_K\}_{i=1}^{m}$.

3.5.6 HPE and Measurement Fusion

Initial HPE

Model-based and appearance-based HPE approaches in which the face must be detected, require the user to be facing the camera during the initialization stage. However, in any real scenario this constraint results in a strong limitation, especially for assistance tools that need to compute the head pose on the fly.

We instead propose a novel scheme to provide an approximate initialization of the head pose as soon as the head is detected and the facial landmarks are visible. The scheme utilizes the 2D facial landmarks retrieved in the first frame $\{\mathbf{p}_F\}_{i=1}^{n}$ along with a prior set of the corresponding 3D facial landmarks $\{\mathbf{P}_G\}_{i=1}^{n}$ on a geometric model when the face is frontal, as depicted in the red box in Figure 3.9. This set is initially normalized and later re-scaled according to the detected face in the 3D space.

Similarly to our approach for pose estimation in Section 3.4.5, we estimate the head pose in the first frame from the 3D-2D correspondences from $\{\mathbf{P}_G, \mathbf{p}_F\}$, by formulating the system as a Perspective-n-Point problem (PnP) and minimizing the reprojection error in the image plane.

HPE for the other frames

Similarly to the initial HPE process, we estimate the pose after formulating a PnP problem from the 3D-2D correspondences and solve by minimizing the reprojection error on the image plane, using the global Levenberg-Marquardt algorithm. To define the 3D-2D correspondences, we use the set of 3D features $\mathbf{P}_i$ that were calculated from the previous frame and the 2D features obtained at the current frame, $\mathbf{q}_i$. This procedure is applied using both set of features, keypoints $\{\mathbf{P}_K, \mathbf{q}_K\}$ and facial landmarks $\{\mathbf{P}_F, \mathbf{q}_F\}$. Consequently, we obtain two measured head poses: $\boldsymbol{\omega}_K$ and $\mathbf{t}_K$ from the keypoints and $\boldsymbol{\omega}_F$ and $\mathbf{t}_F$ from the facial landmarks.

We introduce a method for combining both measurements using a dedicated Linear Kalman Filter. Although Kalman Filter is a very common approach in sensor fusion, it is the first time, to the best of our knowledge, that it is used for fusing measurements given by facial landmarks and by keypoints. As long as we have two different head pose measurements from the two sets of feature points, it is possible to apply this framework.

As described in Chapter 2.2, Kalman Filter is defined on the state-space representation of the system, where the dynamical system is given by Eq. (2.24). In our case, the state transition

matrix **A** is constant and there is no deterministic inputs to the system **u**, reducing Eq. (2.24) to

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{w}_{k-1}. \tag{3.16}$$

$\hat{\mathbf{x}}_k^-$ represents the predicted state of the system at time k and $\mathbf{w}_{k-1}$ is the random noise affecting the system at time $k-1$, commonly termed as process noise. In our framework, the state vector **x** contains the position **t** and orientation $\boldsymbol{\omega}$ of the head, with their respective first and second derivatives, *i.e.*, the velocity and acceleration.

The measurement equation is defined by Eq. (2.26), where $\mathbf{v}_k$ corresponds to the measurement noise, *i.e.*, the random noise in the observation $\mathbf{z}_k$.

$\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be independent, white and having Gaussian zero mean, with covariance matrices given by

$$\mathbf{Q}_k = E[\mathbf{w}_n {\mathbf{w}_k}^T] = \begin{cases} Q_k, & \text{for } n = k \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

and

$$\mathbf{R}_{\boldsymbol{k}} = E[\mathbf{v}_n {\mathbf{v}_k}^T] = \begin{cases} R_k, & \text{for } n = k \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

The measurement (or observation) vector $\mathbf{z}_k$ is composed of both estimated poses. The observation model is given by:

$$\mathbf{H} = \left[\mathbf{H}_K^T, \mathbf{H}_F^T\right]^T \tag{3.19}$$

being $\mathbf{H}_K$ and $\mathbf{H}_F$ the respective observation models from the keypoints and the facial landmarks.

The covariance matrix of the observation noise is then defined by:

$$\mathbf{R} = \begin{bmatrix} \Sigma_K & 0 \\ 0 & \Sigma_F, \end{bmatrix} \tag{3.20}$$

where Σ_K and Σ_F are the observation noise covariance of the poses estimated with keypoints and landmarks, respectively.

It is important to note that facial landmarks are not always detected in every frame. In order to address this problem, we propose to include a dynamic Kalman Filter framework that corrects the state vector from the available measurements. In case the face is not detected, the Kalman Filter is updated using only the pose estimated from the keypoints, that is:

$$\mathbf{H} = \mathbf{H}_K \tag{3.21}$$

and the observation noise covariance as:

$$\mathbf{R} = \Sigma_K. \tag{3.22}$$

Otherwise, the measurement fusion scheme defined by Eq. (3.19) and (3.20) is employed. From this framework, we are able to update the filter and estimate the new pose having either one or two measurements.

3.6 Head Pose Estimation by Fusion of Global and Local Motion

In this Section, we introduce a novel fusion pipeline for HPE from intensity images in monocular video data. Similarly to Section 3.5, a framework based on tracked keypoints and a pipeline relying on the detection of facial landmarks are fused on a tailored Kalman Filter for HPE, extending the working range to large head rotations. This novel pipeline, however, incorporates the head pose computed from the keypoint-based tracking scheme into the prediction step of the Kalman Filter and the head pose computed from the facial-landmark-based detection scheme into the correction step.

The head pose from the tracking scheme is estimated from 2D keypoints tracked in two consecutive frames in the region of the head and their 3D projection on a simple geometric model. In contrast, the head pose from the detection scheme is estimated from 2D facial landmarks detected in each frame and their 3D correspondences retrieved through triangulation. In each scheme, the head pose results from the minimization of the reprojection error from the 3D-2D correspondences. In each iteration, we update the state transition matrix of the filter and subsequently the estimated covariance.

We evaluated our approach on a publicly available dataset and compared with related methods of the state of the art. Our approach could achieve similar performance in terms of mean average error, while operating in real time. Furthermore, we tested our method on our own dataset to evaluate its performance in the presence of large head rotations. The proposed method is robust to extreme head rotations, varying illumination and in cases where facial landmarks are partially occluded.

3.6.1 Overview of the Proposed HPE pipeline

A diagram of the proposed approach is presented in Figure 3.11. The blue region depicts the HPE scheme from keypoints, while the red region describes the facial-landmark-based HPE method. The Kalman filter used to fuse both schemes is depicted inside the green rectangle.

For an input image I_k at time k, we compute separately the head pose using keypoints and the head pose using facial landmarks. For the keypoint-based method (see Figure 3.11), we use a temporal tracking scheme as in Section 3.4, where the correspondences of 2D keypoints are estimated pairwise using optical flow. Then, these 2D keypoints are projected on a simple geometrical head model, to recover 3D keypoints. From the 2D keypoints at the current input image at time k and the 3D keypoints from the previous frame, $k-1$, we obtain an estimation of the head pose, following Section 3.4.5.

For the facial-landmark-based HPE scheme, we align 2D facial landmarks in every input image, independently of the previously alignments (see Figure 3.11). 3D facial landmarks are refined by triangulating 2D facial landmarks detected over time. This differs from the measurement fusion HPE method from Section 3.5, which uses an EHM to estimate the 3D landmarks. Afterwards, we compute the head pose from the 2D facial landmarks at the current frame and the refined 3D facial landmarks at a fixed pose.

The head poses from both independent frameworks are later integrated into a Kalman filter

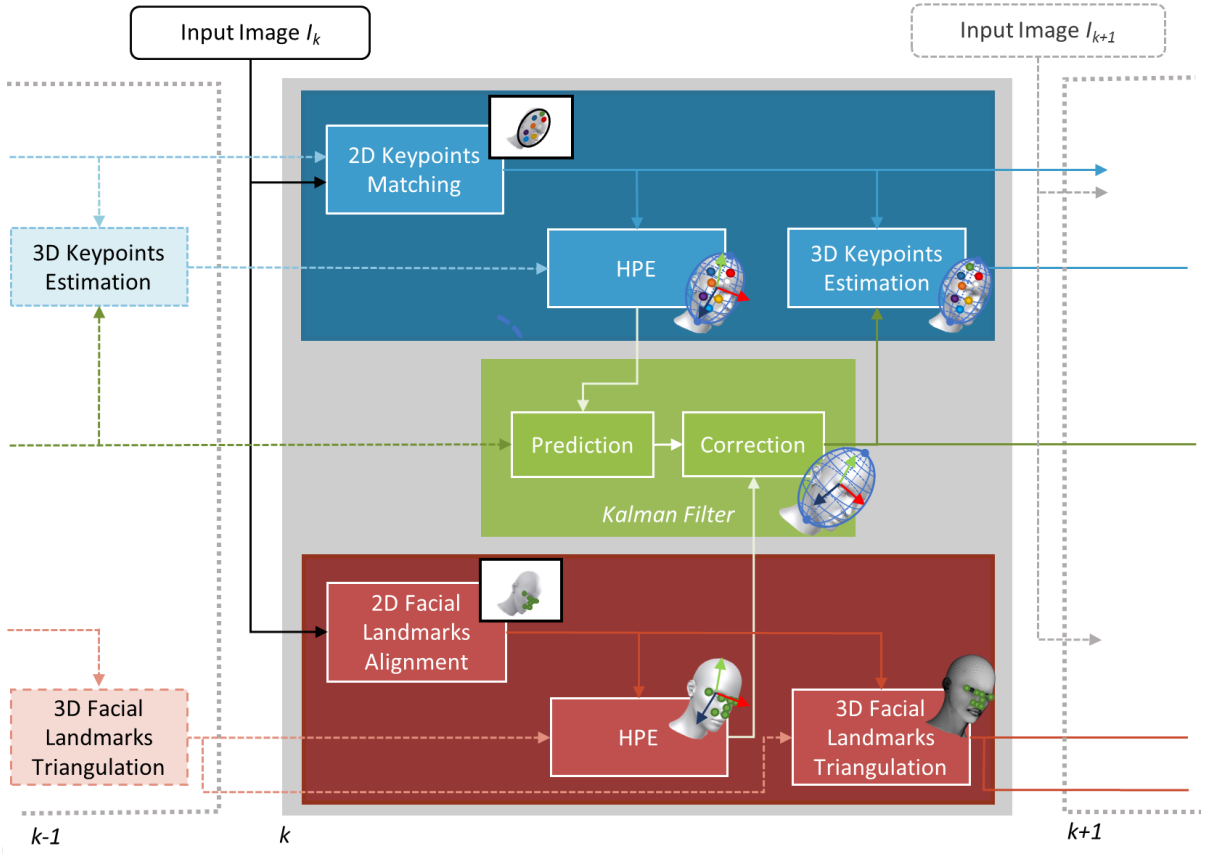


Figure 3.11: Proposed HPE pipeline, from Keypoints (blue) and Facial Landmarks (red).

as follows: the computed pose from keypoints is included at the prediction stage, while the head pose from facial landmarks is used as a new measurement at the correction stage of the filter. As the two different strategies for HPE run in parallel independently of each other, time consumption of the algorithm can be reduced considerably. For our framework, the calibration of the camera is required in advance.

3.6.2 Computation of 2D and 3D Keypoints

We follow a similar procedure as in Sections 3.4.3 and 3.5.3 to extract a set of m 2D keypoints $\{\mathbf{p}_K\}_{i=1}^{m}$, using FAST-ER. Their correspondences $\{\mathbf{q}_K\}_{i=1}^{m}$ are matched by means of the pyramidal iterative Lucas-Kanade feature tracker. The 3D keypoints $\{\mathbf{P}_K\}_{i=1}^{m}$ are calculated in the surface of the geometric head model as detailed in Section 3.4.4.

3.6.3 Computation of Facial Landmarks

Facial landmarks result from the alignment of facial features on a detected face region, as described in Section 3.5.2. Only those facial landmarks that are less prone to be affected by facial expressions or self-occlusion are considered in the set of facial landmarks, *i.e.*, eye corners, nostrils and nasal bridge.

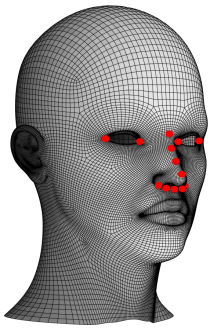


Figure 3.12: Selected facial landmarks in 3D head model.

Following the notations used in this Chapter, the set of n 2D facial landmarks is denoted as $\{\mathbf{p}_F\}_{i=1}^{n}$, while the corresponding set of n 3D facial landmarks is denoted as $\{\mathbf{P}_F\}_{i=1}^{n}$.

2D facial landmarks detection

For every input image, the set of 2D facial landmarks is detected using the method proposed by Kazemi and Sullivan [148]. The resulting facial landmarks are depicted in Figure 3.10.

It should be noted that the face is not always detected in every frame and in such cases the facial landmarks are not aligned. In those frames, the head pose is only estimated from the tracked keypoints, as detailed in Section 3.6.5.

3D facial landmarks triangulation

Given the set of robust 2D facial landmarks described earlier, the corresponding 3D facial landmarks are extracted and refined progressively. For the initial frame, we use 3D points that were retrieved offline on a reference head mesh, as shown in Figure 3.12. These pre-defined 3D features were manually annotated from an open-source 3D face model [149]. Afterwards, as new 2D facial landmarks are detected along the video sequence, 3D facial landmarks are refined using triangulation. This is possible as the extrinsic camera parameters, *i.e.* the camera pose, are known.

Following the linear triangulation method described in Section 2.1.3 [31], from two camera poses $\mathbf{C}$ and $\mathbf{C}'$ and the corresponding sets of 2D facial landmarks, $\{\mathbf{p}_F\}_{i=1}^{n}$ and $\{\mathbf{p}'_F\}_{i=1}^{n}$, the relation of each 2D point $\mathbf{p}$ and its corresponding 3D point $\mathbf{P}$ is defined by Eq. (2.12), where $\mathcal{P} \in \mathbb{R}^{3\times4}$ is the camera projection matrix. Given two views, we can re-write Eq. (2.14) for each point in the form

$$\mathbf{AP} = 0. \tag{3.23}$$

This equation is then solved using the Jacobi's method for finding eigenvalues of symmetric matrices [150, 151]. By using the Jacobi's method, we apply singular value decomposition on matrix $\mathbf{A}$, as follows

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{3.24}$$

where $\mathbf{U}$ is an orthogonal matrix, $\boldsymbol{\Sigma}$ is a diagonal positive definite matrix and $\mathbf{V}$ is an orthogonal

matrix. From this decomposition, we can compute the covariance matrix of $\mathbf{A}$, given by

$$\mathbf{cov} = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T. \tag{3.25}$$

To incorporate new camera poses, and thus refine the 3D facial landmarks, we include a linear Kalman filter where the 3D points are corrected when a new measurement is obtained. This measurement corresponds to the output of the triangulation method from two camera views described before, calculated by solving Eq. (3.23). In each iteration, we use the covariance computed in Eq. (3.25), as the observation noise covariance of the Kalman filter.

With the previous procedure, we avoid the time-consuming process of manual [75] or semi-automatic facial landmarks annotation [118] on large datasets of 3D face scans, yet providing a robust estimated head pose as long as the facial features are visible in the image.

3.6.4 Initialization

Following Section 3.5.4, we adjust the dimension of the EHM according to the size of the user's head. The 2D facial landmarks detected in the first frame are used for this step.

3.6.5 Head Pose Estimation

The head pose estimated from each independent pipeline is computed by minimizing the reprojection error between the 3D features points $\{\mathbf{P}\}_{i=1}^{\eta}$ on the image plane and the 2D correspondences on the image $\{\mathbf{q}\}_{i=1}^{\eta}$ at time k, as detailed in Section 3.4.5. For the keypoint-based scheme, the 3D and 2D features correspond to $\{\mathbf{P}_K, \mathbf{q}_K\}$ respectively, $\eta = m$ and $\{\mathbf{P}_K\}_{i=1}^{m}$ is given at time $k-1$. For the facial-landmark-based scheme, the 3D and 2D features correspond to $\{\mathbf{P}_F, \mathbf{q}_F\}$ respectively, $\eta = n$ and $\{\mathbf{P}_F\}_{i=1}^{n}$ is given at the initial frame. The minimization is expressed by Eq. (2.22).

Before introducing the fusion scheme to combine both estimated head, it is important to understand the difference between both estimates.

For the keypoint-based scheme (Section 3.6.2), we calculate the head pose resulting from tracking keypoints in two consecutive frames. This implies that we compute a transformation from the frame at time $k-1$ to the frame at time k. This frame-to-frame transformation is referred to as a local transformation and is represented by $\mathbf{R}_{k-1}^{k}$ and $\mathbf{t}_{k-1}^{k}$.

Head pose estimated from the facial-landmark-based scheme (Section 3.6.3) is computed with respect to 3D facial landmarks fixed to an initial head pose, $\mathbf{R}_0$ and $\mathbf{t}_0$. Although the 3D facial landmarks are refined over time, the initial pose is not updated. Therefore, the pose retrieved from this scheme maps the head pose from the first given frame at time k_0, to a pose at time k. This transformation is referred to a global transformation and is denoted by $\mathbf{R}_0^k$ and $\mathbf{t}_0^k$.

Given a local and a global transformation, we need to formulate our Kalman Filter in a way that we can integrate both head poses. We define the state vector $\mathbf{x}$ of the filter to be composed of the rotation and translation from the first given frame, *i.e.*, the global head pose. The head rotation is represented using a quaternion $\boldsymbol{\phi} = [\phi_x, \phi_y, \phi_z, \phi_w]^T$, where ϕ_w is the scalar part and $\{\phi_x, \phi_y, \phi_z\}$ the vector part. The translation is denoted in homogeneous coordinates as

$\tilde{\boldsymbol{\tau}} = [\tau_x, \tau_y, \tau_z, 1]^T$. Consequently, the state vector is given by $\mathbf{x} = \left[\boldsymbol{\phi}^T, \tilde{\boldsymbol{\tau}}^T\right]^T$, with a dimension of 8×1.

Initial HPE

For the first given frame, the head pose $\mathbf{R}_0$ and $\mathbf{t}_0$ is computed from facial landmarks only, as 3D keypoints are not available. The pose is recovered by minimizing Eq. (2.22) with the 3D facial landmarks $\{\mathbf{P}_F\}_{i=1}^n$ extracted offline from the reference head model and the 2D facial landmarks $\{\mathbf{p}_F\}_{i=1}^n$ aligned at the first frame. We initialize the Kalman Filter using the computed head pose.

HPE for the other frames

We define a linear Kalman Filter to fuse the pose estimated from keypoints and the pose estimated from facial landmarks. This is possible, as a linear process model can be built by representing the rotation with quaternions and the translation with homogeneous coordinates. The predicted or *a priori* state estimate of the filter $\hat{\mathbf{x}}_k^-$ at time k is calculated using Eq. (3.26), where $\mathbf{A}$ represents the state transition matrix of the process model, with a normal distributed process noise with covariance $\mathbf{Q}$, as described in Chapter 2.2.

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} \tag{3.26}$$

In order to integrate the keypoint-based HPE into the prediction step of the filter, we update $\mathbf{A}$ in every iteration with the pose estimated at the tracking scheme, as shown in Eq. (3.27). The resulting $\mathbf{A}$ is a matrix $\in \mathbb{R}^{8\times8}$.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_\rho & 0 \\ 0 & \mathbf{A}_\mathbf{t} \end{bmatrix} \tag{3.27}$$

$\mathbf{A}_\rho \in \mathbb{R}^{4\times4}$ is the state transition sub-matrix to project the rotation ahead and is computed from the local rotation $\mathbf{R}_{k-1}^k$. We convert this rotation matrix to a quaternion $\boldsymbol{\rho} = \left[\rho_x, \rho_y, \rho_z, \rho_w\right]^T$, and calculate $\mathbf{A}_\rho$ as follows

$$\mathbf{A}_\rho = \begin{bmatrix} \rho_w & -\rho_z & \rho_y & \rho_x \\ \rho_z & \rho_w & -\rho_x & \rho_y \\ -\rho_y & \rho_x & \rho_w & \rho_z \\ -\rho_x & -\rho_y & -\rho_z & \rho_w \end{bmatrix}. \tag{3.28}$$

$\mathbf{A}_\mathbf{t} \in \mathbb{R}^{4\times4}$ represents the state transition sub-matrix to update the translation, and is defined by

$$\mathbf{A}_\mathbf{t} = \begin{bmatrix} \mathbf{R}_{k-1}^k & \mathbf{t}_{k-1}^k \\ 0 & 1 \end{bmatrix}. \tag{3.29}$$

The new translation estimate $\hat{\mathbf{t}}_0^{k-}$ is given by

$$\hat{\mathbf{t}}_0^{k-} = \mathbf{R}_{k-1}^k \hat{\mathbf{t}}_0^{k-1} + \mathbf{t}_{k-1}^k. \tag{3.30}$$

The covariance $\boldsymbol{P}^-$ at the prediction step is computed from Eq. (2.25).

The measurement model of the Kalman Filter is given by Eq. (2.26). The new measurement $\mathbf{z}_k$ at time k corresponds to the head pose retrieved from the facial-landmark-based scheme,

i.e., the global head pose. $\mathbf{H} \in \mathbb{R}^{7\times 8}$ relates the current state $\mathbf{x}_k$ to the measurement and is given by $\mathbf{H} = [\mathbf{I}_7\ 0]$, where $\mathbf{I}_7$ is a 7×7 identity matrix. $\mathbf{v}_k$ denotes the measurement noise in the observation with covariance $\mathbf{R}$.

The updated or *a posteriori* state estimate $\hat{\mathbf{x}}_k$ is calculated from Eq. (2.28), where $\mathbf{K}_k$ represents the Kalman gain, defined in Eq. (2.27).

The covariance is updated at the correction step using Eq (2.29).

Occlusion handling and pose recovery

One challenge in HPE is to compute the head pose when the face is occluded due to large head rotations or when it is not detected at all. The facial-landmark-based scheme fails to provide an estimation in these cases, so our fusion approach uses only the head pose computed from the keypoint-based scheme. This implies that the head pose calculated at the Kalman Filter is given only from the prediction step, as no new measurement is available. Thereby, our approach is able to provide an estimated pose even when the face has not been detected.

When the face is detected again, the output of the facial-landmark-based scheme is integrated again at the Kalman Filter to correct the estimated pose. This step is fundamental in our approach, especially if the face has not been detected for several consecutive frames. If the pose has been computed only from keypoints for a long sequence, the estimation might suffer from drifting, while increasing the state covariance (uncertainty) of the filter (from Eq. (2.25)) over time. When a new measurement is available, the recovery takes place rapidly, since the weight of the predicted state covariance is relatively small. In the proposed pipeline, the covariance of the process noise and measurement noise are set fixed for the entire sequence.

3.7 Experiments and Results

In this section, we describe the methodology implemented to evaluate our HPE approaches and the analysis of the results in comparison to other methods of the state of the art. The proposed methods were implemented in C++, with assistance of the OpenCV library [152] and dlib [153].

3.7.1 Datasets

We evaluated the proposed HPE frameworks on the Boston University (BU) head pose benchmark introduced in [37]. The dataset is divided in two sets of videos recorded with a RGB camera in an office setup. The first set contains 45 video sequences from 5 different subjects performing different motions in an controlled environment under uniform illumination. The second set has 27 video sequences from 3 subjects under varying illumination. The ground truth was acquired using the Flock of Birds magnetic sensor attached to the head, with a nominal accuracy of 1.8 mm in translation and 0.5 degrees in rotation.

Additionally, we evaluated the performance of the HPE fusion approach (Section 3.6) under extreme head poses using our own video sequence. We made this video publicly available for research purposes at [154], and included ground truth with the respective calibration file. The video contains 1263 images of a person sitting, while moving her head with large rotations. In

some frames, the head is partially self-occluded, due to rotations larger that 45 degrees in the X or Y axes (pitch and yaw).

3.7.2 Evaluation Metrics

The performance of the proposed HPE approaches was evaluated using three metrics: root mean square error (RMSE) in Eq. (2.41), mean absolute error (MAE) in Eq. (2.42) and standard deviation (STD) in Eq. (2.44), in Section 2.5. In these equations, n represents the number of frames, s_i the ground truth $\mathbf{s}$ at time i and $\hat{s}_i$ the estimate $\hat{\mathbf{s}}$ of the position or angle, at time i.The same metrics were used for comparison with relevant methods of the state of the art (SOTA).

3.7.3 Results

The results of our approaches and related works on the BU dataset with uniform illumination are reported on Table 3.1. Our best approach, described in Section 3.6 is also evaluated on the BU set with varying illumination (see Table 3.2). In the methods using the 2D feature weighting scheme from Section 3.4.3, we chose a weight threshold of 5.

For the HPE method based on keypoints from Section 3.4, we report the results using the EHM and CHM. We observed that the EHM had lower errors in every metric, as it resembles more the shape of the head.

The HPE pipeline from Section 3.5 is listed as *Measurement Fusion with KF*. The results show an improvement in the estimation of the pitch angle, with respect to the EHM-based HPE from keypoints, but a higher error in the yaw angle. However, in average it has a slightly lower MAE.

The HPE fusion pipeline from Section 3.6 is listed as *Fusion of global & local motion*. We also evaluated the contribution of each HPE scheme in our approach, by assessing them individually. The last three rows of Tables 3.1 and 3.2 present the angular accuracy from the individual HPE approaches, using only keypoints (K.P.) or facial landmarks (F.L.) and HPE from fusion. It should be noted that the latter performs better for every angle estimation than each individual method and that our previous approaches. Only for the varying illumination set, Table 3.2, the performance of the facial-landmark-based method is comparable to the fusion approach.

In the BU dataset, no calibration data is provided. For that reason, most of the related work do not report their translation estimation errors. We have included the results of our HPE method from Section 3.6 in Tables 3.3 and 3.4, for the uniform and varying illumination sets, respectively. We assumed the principal point of the camera to be in the center of the image, with no pixel skew or distortion, and square pixels, and that the users are at a distance of about 1 meter of the camera center. In Tables 3.3 and 3.4 it can be noted that in average, the fusion scheme presents the lowest errors in translation for both sets, with a similar performance of the keypoint-based HPE scheme. This can be explained as this pipeline uses frame-by-frame tracking, so the 2D estimation of the features is robust. In contrast, in the facial-landmark-based scheme the feature detection is not always precise and there are small displacements in the alignments between frames even when the person is not moving.

Table 3.1: Evaluation on the BU benchmark with uniform illumination.

Method and Year	RMSE$^{\downarrow}$ ± STD$^{\downarrow}$			MAE$^{\downarrow}$				Time
	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average	(FPS)
La Cascia *et al*. [37], 2000	-	-	-	9.8	6.1	**3.3**	6.4	-
Sung *et al*. [138], 2008	-	-	-	3.1	5.6	5.4	4.7	-
Morency *et al*. [143], 2008	-	-	-	2.91	3.67	4.97	3.85	6
Jang & Kanade [121], 2008	-	-	-	2.1	3.7	4.6	3.47	-
An & Chung [93], 2008								
Ellipsoid	-	-	-	2.83	3.95	3.94	3.57	-
Cylinder	-	-	-	3.22	7.22	5.33	5.26	-
Plane	-	-	-	2.99	7.32	18.08	9.46	-
Choi & Kim [94], 2008								
Ellipsoid	-	-	-	2.82	3.92	4.04	3.59	14
Cylinder	-	-	-	2.45	4.43	5.19	4.02	14
Kumano *et al*. [95], 2009	-	-	-	2.9	4.2	7.1	4.73	-
Lefevre & Odobez [155], 2009	-	-	-	2.0	3.3	4.4	3.23	3
Asteriadis *et al*. [124], 2010	3.56	4.89	5.72	-	-	-	-	-
Prasad & Aravind [127], 2010	-	-	-	3.6	**2.5**	3.8	3.3	-
Jang & Kanade [122], 2010	-	-	-	2.07	3.44	4.22	3.24	-
Saragih *et al*. [144], 2011	-	-	-	2.55	4.46	5.23	4.08	8
Valenti *et al*. [98], 2012								
Fixed template with eye cues	3.00± 2.82	5.26±4.67	6.10± 5.79	-	-	-		-
Fixed template w/o eye cues	3.85±3.43	6.00±5.21	8.07±7.37	-	-	-		-
Updated template with eye cues	3.93±3.57	5.57±4.56	6.45±5.72	-	-	-		-
Updated template w/o eye cues	4.15±3.72	5.97±4.87	6.40±5.49	-	-	-		-
Wang *et al*. [126], 2012	-	-	-	**1.86**	2.69	3.75	**2.76**	15
Baltrusaitis *et al*. [118], 2012	-	-	-	2.08	3.81	**3.00**	**2.96**	-
Tran *et al*. [156], 2013	-	-	-	2.4	3.9	5.4	3.90	5
Vicente *et al*. [99], 2015	-	-	-	3.2	6.2	4.3	4.56	25
Jeni *et al*. [75], 2017	-	-	-	2.41	**2.66**	3.93	**3.0**	**50**
Wu *et al*. [131], 2017	-	-	-	3.1	5.3	4.9	4.43	-
Gou *et al*. [132], 2017	-	-	-	3.3	4.8	5.1	4.4	-
HPE from keypoints - Section 3.4								
Ellipsoid	3.36±2.98	4.46±3.84	5.09±4.56	2.56	3.39	3.99	3.31	**56**
Cylinder	3.66±3.35	5.73±4.54	6.16±5.42	2.80	4.58	4.87	4.08	42
HPE from Section 3.5								
Integration K.P.+F.L.	3.44±3.11	4.51±3.97	5.84±4.98	2.62	3.59	4.73	3.65	-
Integration with KF	3.37±3.00	4.31±3.60	5.30±4.70	2.56	3.29	4.13	3.33	-
Measurement Fusion with KF	3.36±2.99	4.32±3.62	5.25±4.70	2.54	3.27	4.07	3.29	23
HPE from Section 3.6								
HPE from K.P.	3.42±3.05	4.53±3.78	5.46±4.81	2.61	3.45	4.27	3.44	-
HPE from F.L.	2.46±2.31	4.43±3.61	5.23±4.85	**1.92**	3.49	4.17	3.19	-
Fusion of global & local motion	**2.41±2.20**	**4.11±3.26**	**4.92±4.40**	**1.91**	3.26	3.92	3.03	40

Table 3.2: Evaluation on the BU benchmark with varying illumination.

Method and Year	RMSE$^{\downarrow}$ ± STD$^{\downarrow}$			MAE$^{\downarrow}$				Time
	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average	(FPS)
Lefevre & Odobez [155], 2009	-	-	-	2.3	3.5	**4.1**	3.3	3
Jang & Kanade [122], 2010	-	-	-	2.8	4.25	5.92	4.32	-
Jeni *et al*. [75], 2017	-	-	-	2.24	**2.72**	4.87	3.27	50
HPE from Section 3.6								
HPE from K.P.	3.44±2.77	5.31±4.38	6.67±5.56	2.73	4.20	5.21	4.04	-
HPE from F.L.	2.70±2.17	4.14±3.59	5.21±4.51	**2.18**	3.22	**4.12**	3.17	-
Fusion of global & local motion	**2.66±2.08**	**4.11±3.42**	**5.15±4.05**	**2.16**	3.23	**4.12**	**3.17**	40

Table 3.3: Errors on the translation estimation on BU benchmark with uniform illumination.

Method	RMSE$^{\downarrow}$ ± STD$^{\downarrow}$			MAE$^{\downarrow}$			
	Trans. X	Trans. Y	Trans. Z	Trans. X	Trans. Y	Trans. Z	Average
HPE from K.P.	4.09±3.76	**2.10±1.68**	1.58±1.23	3.27	**1.64**	1.22	2.04
HPE from F.L.	4.30±4.07	2.49± 2.05	1.45±1.19	3.48	1.99	1.13	2.2
Fusion global & local motion	**3.89±3.65**	2.31±1.91	**1.40±1.07**	**3.15**	1.86	**1.13**	2.04

Table 3.4: Errors on the translation estimation on BU benchmark with varying illumination.

Method	RMSE$^{\downarrow}$ ± STD$^{\downarrow}$			MAE$^{\downarrow}$			
	Trans. X	Trans. Y	Trans. Z	Trans. X	Trans. Y	Trans. Z	Average
HPE from K.P.	3.17± **2.63**	**2.41± 1.82**	1.53± 1.09	2.58	**1.94**	1.24	1.92
HPE from F.L.	3.69±3.26	2.66±2.12	1.11±0.89	3.00	2.18	0.91	2.03
Fusion global & local motion	**3.10**±2.85	2.76±2.16	**0.96±0.75**	**2.54**	2.29	**0.77**	1.87

Table 3.5: RMSE of the head orientation based on the chosen weight threshold on the weighting scheme.

Weight thresh.	RSME$^{\downarrow}$							
	CHM				EHM			
	Roll	Pitch	Yaw	Average	Roll	Pitch	Yaw	Average
0	4.22	6.37	6.30	5.63	3.41	**4.36**	5.31	4.36
5	3.66	**5.73**	**6.16**	5.18	3.36	4.46	**5.09**	**4.30**
10	3.23	5.89	6.22	**5.11**	**3.29**	4.79	5.44	4.51
15	3.15	6.19	6.56	5.3	3.34	5.04	5.69	4.69
25	**3.06**	6.32	6.46	5.28	3.31	5.19	5.66	4.72
35	3.16	6.65	6.90	5.57	3.33	5.48	5.76	4.86

3.7.4 Ablation Study

2D Feature Weighting Scheme

We evaluated the performance of the 2D feature weighing scheme based on the distance transform, proposed in Section 3.4.3. To that end, multiple weight thresholds were evaluated on the keypoint-based HPE pipeline detailed in Section 3.4. Table 3.5 reports the RMSE of the estimated head orientation by both CHM and EHM, and given by the proposed HPE approach depending on the chosen weight threshold.

As expected, the overall performance improves in both geometrical models as soon as a weight threshold above 0 is chosen. Indeed, outliers that correspond to 2D features from the background of the head are then discarded. However, in the case of the EHM, the weighting scheme improvement is less significant for the pitch rotation, which might be due to the fact that the EHM fits better to the shape of the head and thus, less outliers are detected. This better fitting also explains why the EHM outperforms the CHM, where errors are introduced when computing the 3D features by ray tracing.

HPE Pipeline by Measurement Fusion Reformulated

Using the measurement fusion HPE method from facial landmark detection and keypoint tracking in Section 3.5, we also investigated the effect of combining facial landmarks with keypoints using two additional pipelines:

- *Integrating feature points in common pipeline.* For this first method, we integrated the set of 2D keypoints $\mathbf{p}_K$ along with the set of 2D facial landmarks $\mathbf{p}_F$, to create one combined set of 2D features points $\mathbf{p}_{K+F}$. The set of 3D feature points $\mathbf{P}_{K+F}$, composed as well of 3D keypoints $\mathbf{P}_K$ and 3D facial landmarks $\mathbf{P}_F$, were computed as described in Subsection 3.4.3, by using ray tracing. The head pose was also retrieved by finding the 3D-2D correspondences and minimizing the reprojection error from the combined set of features, $\{\mathbf{P}_{K+F}, \mathbf{p}_{K+F}\}$. Similarly to the fusion approach, the face ROI was updated as detailed in Section 3.4.6. This method is listed in Table 3.1 as *Integration K.P.+F.L.*.

- *Integrating feature points and adding Kalman Filter.* For the second pipeline, we proceeded similarly to the previous method, *Integration K.P.+F.L.*, but we included an extra step to incorporate a Kalman Filter. In this case, the pose computed from the set of features $\{\mathbf{P}_{K+F}, \mathbf{p}_{K+F}\}$ is integrated as a single measurement in the KF, hence updating the state vector accordingly. The results are reported in Table 3.1 as *Integration with KF*.

In comparison to the method from Section 3.5, *Measurement Fusion with KF*, we observed the lowest performance in every metric when combining the keypoints and facial landmarks (*Integration K.P.+F.L.*) and estimating the pose directly from the correspondences. The results considerably improve after adding the Kalman Filter (*Integration with KF*) and are further enhanced when the poses from the keypoints and landmarks are added as independent measurements in the Kalman Filter. The improvement is more evident in the yaw angle, affected by large head rotations.

Table 3.6: HPE performance with and without facial landmark refinement with triangulation.

Method	Triang.	RMSE$^{\downarrow}$ ± STD$^{\downarrow}$			MAE$^{\downarrow}$			
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average
HPE from F.L.	-	2.58±2.42	5.68±4.31	7.18±6.71	2.01	4.45	5.87	4.11
	✓	2.46±2.31	4.43±3.61	5.23±4.85	**1.92**	3.49	4.17	3.19
Fusion of global & local motion	-	3.06±2.78	4.38±3.76	**4.93**±4.56	2.32	3.41	**3.90**	3.21
	✓	**2.41±2.20**	**4.11±3.26**	**4.92±4.40**	**1.91**	**3.26**	3.92	**3.03**

Table 3.7: Evaluation of HPE from K.P., F.L. and fusion approach on challenging sequence.

Method	RMSE$^{\downarrow}$ ± STD$^{\downarrow}$			MAE$^{\downarrow}$			
	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average
HPE from K.P.	57.63±30.20	55.04±54.02	32.78±25.15	52.94	46.02	25.70	41.55
HPE from F.L.	16.34±16.16	**16.66±16.61**	9.82±9.73	9.72	12.61	5.66	9.33
Fusion global & local motion	**14.87±14.69**	16.85±16.81	**9.69±9.42**	**8.24**	**12.09**	**5.65**	**8.66**

Evaluation of 3D Facial Landmarks Triangulation

In this ablation study, we evaluated the impact of the facial landmark refinement with triangulation in the HPE pipeline introduced in Section 3.6. We compared the performance using the initial landmarks through the entire sequence, in contrast to the refined landmarks. The performance of the HPE methods with and without triangulation are shown in Table 3.6. The experiments include the HPE pipeline using only facial landmarks and the HPE approach based on the fusion of the global-local motion.

We observed that every metric is improved after refining the facial landmarks. Only in the yaw rotation there is a similar performance in the fusion-based approach. This shows that the integration of the keypoints-based HPE framework helps alleviate the error introduced by large head rotations when using only facial landmarks.

Evaluation of Extreme Head Rotations

We additionally evaluated the HPE fusion approach described in Section 3.6 and compare it against both independent schemes using only keypoints and only facial landmarks. The approaches are evaluated using our own video sequence, presented in Section 3.7.1. The most challenging estimations are the yaw and pitch rotation, where facial landmark detectors usually have higher failure rates.

Figure 3.13 presents the results for different frames from the three HPE schemes. For each frame, the results are shown as follows from top to bottom: keypoint-based method; facial-landmark-based scheme; and the fusion approach. The blue area around the face depicts the projection of the ellipsoid onto the 2D image, while the coordinate system representing the estimated pose is displayed with the RGB arrows. For the facial-landmark-based scheme, we define a plane parallel to the image plane with the dimensions of the head and we project it on the 2D image.

We also estimated the RMSE, MAE and STD for the three rotation angles. Results are presented in Table 3.7. Similar to the BU benchmark, the fusion scheme presents the lowest errors.

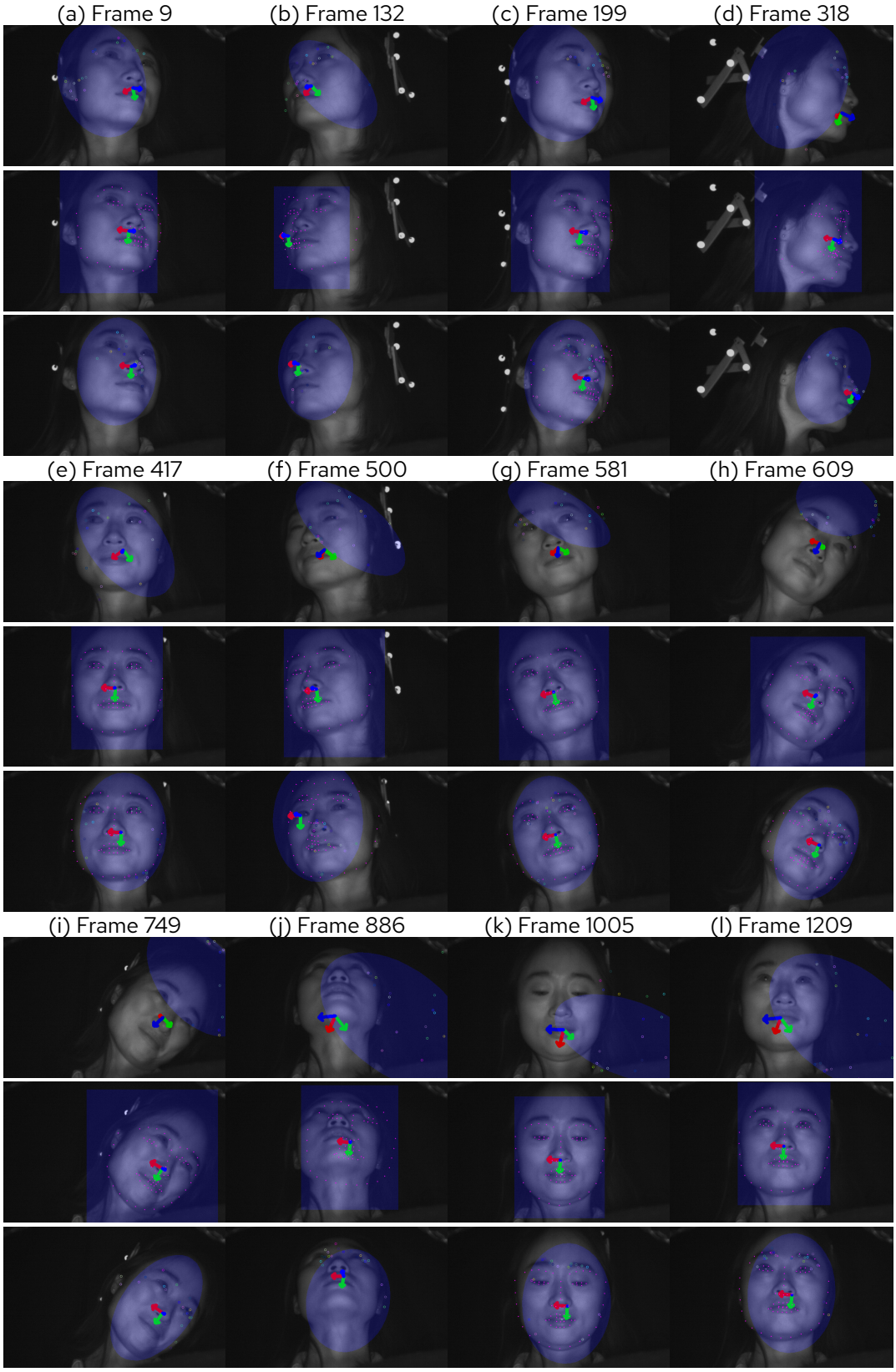


Figure 3.13: Head pose estimation on challenging sequence from K.P. (top), F.L. (center) and the fusion of global & local motion (bottom).

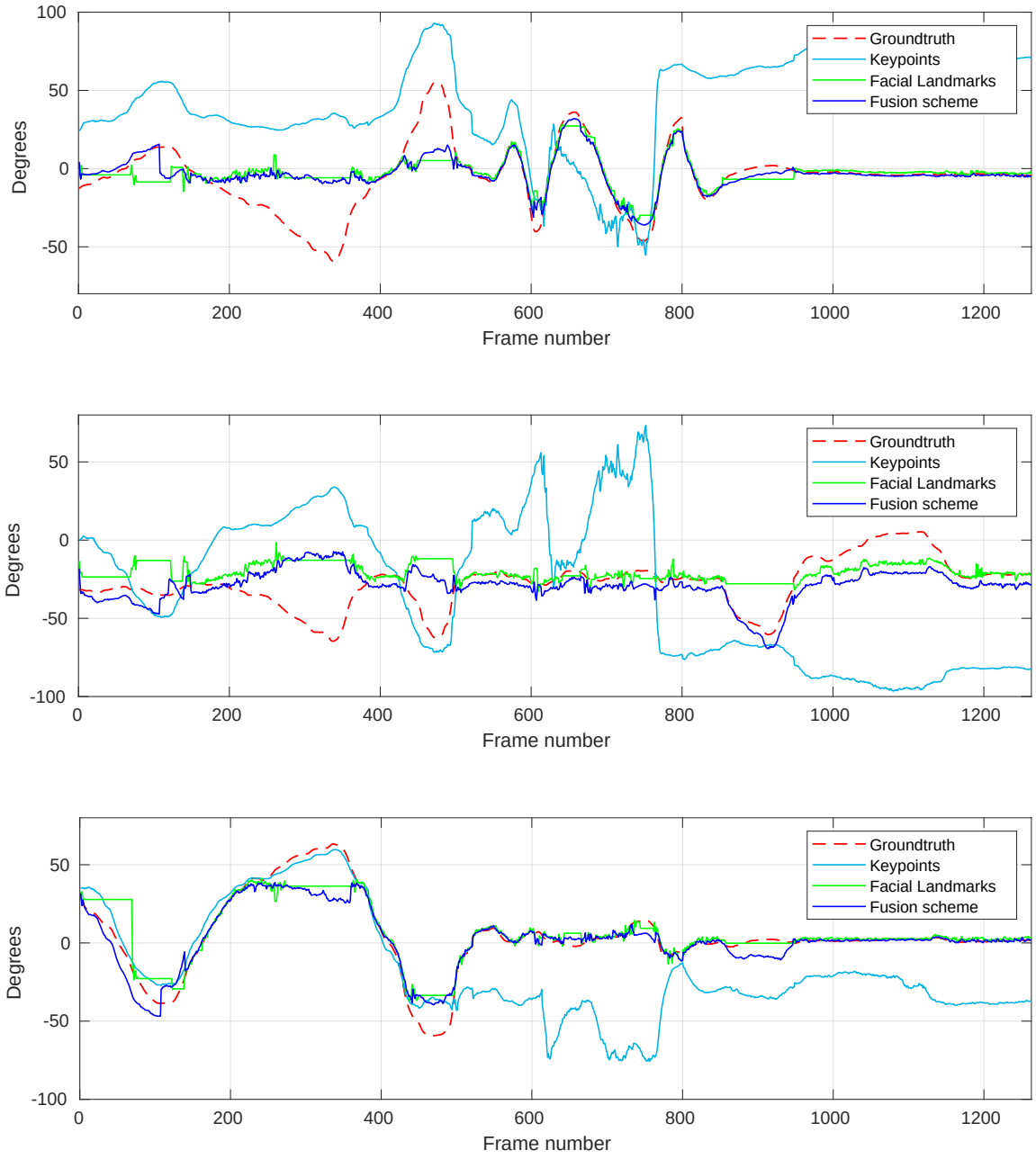


Figure 3.14: Rotation error on challenging sequence. From top to bottom: estimated roll, pitch and yaw angles from K.P., F.L. and the fusion of global & local motion.

The high errors in the keypoint-based scheme is due to the fact that the estimated head pose started drifting and it was not able to recover (see Figure 3.13 from frame 417 (e) onwards).

The estimated angles in degrees for the three schemes are presented in Figure 3.14. For some frames, the pose from the facial-landmark-based method could not be updated from the last estimate, since the face was not detect. This can also be noted in Figure 3.13 in frames 9 (a), 132 (b), 318 (d), 749 (i) and 886 (j). However, as soon as the face is correctly detected again, the facial-landmark-based approach is able to recover. On the other hand, the head pose from the keypoint-based method is continuous, but suffers from drifting in long sequences. The fusion scheme is able to estimate the pose through the entire sequence, even with large head rotations (frames 318 (d) and 886 (j)) and recover in case it starts drifting, exploiting the advantages from both pipelines.

3.7.5 Runtime Analysis

For many applications, it is of great importance the approach performs in real time. We measured the average time to execute each step of the proposed pipelines on the BU dataset. HPE based on keypoints from Section 3.4 and the measurement fusion HPE approach from 3.5 were evaluated using a CPU with an Intel Core(TM) i5-4210U processor. The fusion method based on the global and local motion described in Section 3.6 was tested on an Intel® Xeon(R) W3520 processor with 8 GB of RAM. The results are presented in Tables 3.8, 3.9, 3.10. A comparison with other methods of the state of the art is presented in the last column of Tables 3.1 and 3.2.

Table 3.8 reports the time consumption of each step of the proposed HPE pipeline based on keypoints (Section 3.4). The initialization is the most time-demanding step whereas the rest of the process perfectly suits for real-time applications in which a robust HPE is required, *e.g.*, drowsiness, distraction/attention, or fatigue detection. For the measurement fusion method (Section 3.5), the first frame requires around 50 ms to detect the face and provide an initial estimation of the head pose using only facial landmarks (see Table 3.9). For the rest of the frames, the pipeline is also suitable for real time applications, as it can run at 23 FPS. In the HPE fusion approach based on the global and local motion (Section 3.6), the HPE for both the initialization step and the other frames took around 40 FPS (see Table 3.10).

This is an important advantage with respect to methods of the state of the art that do not perform in real time, as [133, 127, 126]. If we compare to previous real-time approaches, only [75] could reach $>$40FPS, with estimation errors similar to the proposed approach. In contrast to [75], we did not need manual annotation on a large dataset of high-resolution 3D face scans.

3.7.6 Discussion

We evaluated and compared the proposed HPE pipelines using the BU benchmark. We observed that in long video sequences, the keypoint-based method from Section 3.4 might show drifting over time. Hence, the system should re-initialize after a number of frames. This problem is tackled by integrating the head pose computed from facial landmarks aligned frame by frame. Our experiments show that the highest performance is achieved by adding the pose as

Table 3.8: Averaged runtime of keypoints-based method for 150 runs.

Process	Time (ms)	
	CHM	EHM
System Initialization	657	631
Initial HPE	23.51	17.83
Runtime of first frame	680.51	648.83
Robust 2D Keypoints Detection	2.29	1.99
2D Feature Matching	9.89	6.97
Computation of 3D Features	1.04	1.68
Head Pose Estimation	3.43	2.49
ROI for 2D Keypoints Detection	6.87	4.70
Runtime of other frames	23.51	17.83

Table 3.9: Averaged runtime of HPE by measurement fusion for 100 runs.

Process	Time (ms)
Face detection & EHM initialization	31.61
Initial HPE	17.87
Runtime of first frame	49.48
ROI Update	0.49
Feature detection and tracking	23.84
3D Reconstruction	0.27
Head pose estimation	18.6
Runtime of other frames	43.2

Table 3.10: Averaged runtime of HPE by fusion of global-local motion for 100 runs.

Process	Time (ms)
Initial face detection	22.34
Initial HPE	2.01
Runtime of first frame	24.35
2D feature detection and matching	21.27
Estimation of 3D keypoints	0.3
HPE and refinement of 3D facial landmarks	3.12
Runtime of other frames	24.68

new measurement in the correction stage of a Kalman Filter, while the pose from the keypoints is used at the prediction stage (Section 3.6).

From Table 3.1 we can observe that our results are comparable or even better than state-of-the-art approaches such as [98], and with a major advantage of not using eye cues. For the uniform illumination set in the BU benchmark, Table 3.1, only [75] presents a similar average error to our fusion approach from Section 3.6, with a higher estimation rate. Other methods as [126] and [118] present lower MAE, but with much lower estimation rate. On the varying illumination set, Table 3.2, our fusion approach presents the best performance, with the lowest average of the MAE and real-time capability.

We noted that some of the approaches considered for evaluation do not use any type of head model, *e.g.*, [124], requiring the detection of facial landmarks in each frame. As shown in our experiments, landmark-based methods might fail under large head poses, occlusion or failed face detection. On the contrary, [126] and [127] use 3D parameterized face models that are not suitable for real-time applications. That is also the case of [118], even though the approach has a low MAE. [75] on the other hand, can perform on real time, but a large dataset of high-resolution 3D face scans are required for training (the method used around 300,000).

3.8 Concluding remarks

In this Chapter, we investigated real-time 6DoF head pose estimation from a monocular system and with limited resources. We proposed three HPE approaches based on intensity-images, which target large head rotations in video sequences under realistic scenarios.

Our first approach only requires 2D keypoints to accurately provide the head pose, without the need to detect the face or facial landmarks in every frame. Indeed, and in contrast to the alternative HPE approaches, we simply use 2D keypoints that we filter according to an introduced 2D feature weighting scheme based on the distance transform. We model the head using a simple geometric model such as CHM or EHM, and we generate 3D features by projecting the previously computed 2D features onto the model surface. 2D feature correspondences are also detected in the following frame and used to solve the PnP problem, from which results the head pose. By doing so, our approach can handle extreme head rotations in which facial landmarks are not necessarily available and it is robust to self-occlusions.

The second approach estimates the 6 DoF head pose from 2D images, by combining the head poses estimated from facial landmarks detected in each frame and tracked keypoints. Similarly to the previous approach, the head poses are computed by solving the PnP problem for each set of feature points. To that end, we use the detected 2D features and the corresponding 3D points calculated by projecting the 2D points from the previous frame onto an EHM. Both head poses are then integrated as measurements in a dedicated Kalman Filter.

We additionally introduce a method for initializing the head pose when it is different to frontal. To do so, we use the facial landmarks detected in the first frame and their corresponding 3D points on a synthetic head model that is facing the camera. We then compute the initial pose by minimizing the reprojection error from the 3D-2D correspondences.

The third and last approach integrates into a tailored Kalman Filter the head pose given by a global transformation based on detected facial landmarks, with the head pose given by a local transformation based on tracked keypoints. Its particular strength is that it combines the advantages of both HPE methods. On the one hand, it benefits from the robustness of detecting keypoints, which is nearly always possible, regardless of the head pose and (limited) occlusions. On the other hand, it benefits from the absolute (i.e. not relative to a previous head pose) determination of the head pose based on facial landmarks, which is unaffected by drift or similar effects. Another great advantage is that the head pose from keypoints and the head pose from facial landmarks can be determined independently of each other, *i.e.*, the corresponding calculations can be performed in an arbitrary sequence, *e.g.* in parallel. These greatly help to provide a real-time head pose estimation, which is robust to large head rotations.

We evaluated and compared our approaches to other methods of the state of the art obtaining similar and in some cases outperforming results w.r.t. existing methods on the BU benchmark, with real time capabilities. We also demonstrated that our latest approach provides a reliable estimation even for extreme head poses and under varying light conditions.

For future work, we are interested in investigating a method to refine the 3D keypoints, in order to have a more robust estimation from the keypoint-based HPE scheme. An EHM provides a good representation of the human head. However, it might add approximation errors when estimating the pose. For that reason, we are interested in exploring in the future the performance of the pipeline with a refined head model, that adjusts better to the shape of the head.

An additional future direction to improve the framework is to replace the linear Kalman Filter by an Extended Kalman Filter (EKF), as it provides a more accurate modelling of the 3D motion of the head, and thus a better estimation of the rotation angles.

4 Facial Landmark Detection and Tracking

Contents

4.1 Introduction 72
4.2 Related Work 74
4.3 Face Alignment on Symmetrical and Asymmetrical Expressions 77
4.4 Facial Landmark Tracking 94
4.5 Conclusions 103

Publications

This Chapter is based on the following work:

- Jilliam M. Díaz Barros, J. Rambach, P. Murthy and D. Stricker, "In-Domain Inversion for Improved 3D Face Alignment on Asymmetrical Expressions", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024.
- Jilliam M. Díaz Barros, C-Y. Wang, D. Stricker and J. Rambach, "Confidence-Aware Clustered Landmark Filtering for Hybrid 3D Face Tracking", in IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia. 2023.

4.1 Introduction

Facial landmark detection, often termed as face alignment, is a well-studied research problem in computer vision. Its main objective is to register a predefined set of landmarks on a face image, as described in Section 2.4.2. Furthermore, it is a key step to other face analysis tasks, such as head pose estimation [117], face recognition [157], reconstruction [158, 159, 75, 160, 87], performance capture [161, 19] and palsy assessment [162].

With the introduction of large-scale datasets [13, 82, 163] for training deep neural networks (DNN), 2D face alignment gained a performance boost w.r.t. traditional computer vision approaches, especially for challenging images with varying illumination, large head poses and occlusion. These datasets, however, have relatively few samples of large asymmetrical expressions and even less of peripheral facial paralysis, or palsy, affecting current face alignment approaches (see Figure 4.1). This limitation has a negative impact on palsy assessments that rely on face alignment [164, 165]. Such assessments usually require the patient to follow predefined facial expressions, *e.g.* raising the eyebrows, closing the eyes and smiling. Then, an asymmetry index is computed based on measurements between specific areas in the affected side w.r.t. the unaffected side or the face at rest. An automatic method for extracting features or parts of the face used in the evaluations would reduce the associated costs and observer dependence inherent to manual assessment [166, 167]. In addition, 2D-landmark-based palsy assessment requires fully frontal face images [165] or pose correction techniques [167, 168], while the assessment with 3D landmarks is less prone to measurement errors since distances are not affected by the face orientation.

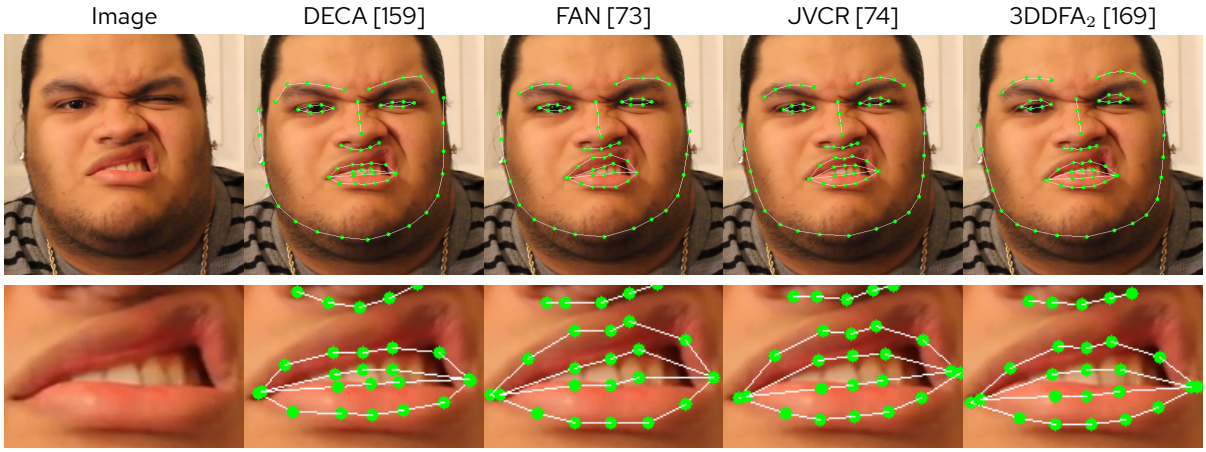


Figure 4.1: Face alignment on patient with palsy. Top row: landmarks extracted from SOTA architectures. Bottom row: Close-up of the landmarks in the mouth. Note that the landmarks are defined around the contours of the lips.

Face alignment methods also depend on the accuracy of the face detectors, and their robustness to locate faces in the wild. Since many of them work on a tracking-by-detection basis, their performance is limited in video sequences when the face is not consistently detected.

While the current state of the art on face alignment for near-frontal and partially-occluded faces has reached high robustness, the detection and alignment still fail on cases with large occlusion, extreme head poses and blurriness [170].

Motivated by the limitations of current approaches to (i) align landmarks on large asymmetrical facial expressions and (ii) exploit the temporal consistency of the landmarks in video sequences, we introduce two new approaches on face alignment and tracking. The first approach aims to detect 3D landmarks on a wide range of facial expressions, including large asymmetrical expressions from patients with facial palsy, alongside healthy subjects. Our second approach integrates landmark tracking into existing deep-neural-network (DNN) face alignment pipelines, to improve the detection of landmarks in video sequences.

The first approach exploits unlabeled face data with and without facial paralysis, to train an autoencoder and create an intermediate representation in a latent vector. In this stage, an in-domain-inversion module is incorporated to ensure a smooth latent space and enhance the representation of the expressions. In the supervised stage, we integrate interleaved transfer layers to the decoder to regress 3DA-2D landmarks, inspired in the state of the art (SOTA) 2D face alignment method, 3FabRec [171]. Our approach additionally enables the detection of 3D landmarks by means of a newly proposed 3D landmark detector. By relying on unlabeled data, our approach seeks to alleviate the cumbersome landmark annotation task, particularly for clinical data.

Our second approach introduces a novel framework based on Kalman Filter (KF), that exploits the correlation between the landmarks by using grouped landmark filters. The framework additionally leverages the uncertainty of the face alignment predictors, by adapting the Kalman Filter measurement noise. Our method is modular and can be combined with any off-the-shelf face alignment approach which retrieves landmark coordinates.

The main contributions of this Chapter are:

- A novel approach for 3D face alignment which encompasses cases with large facial asymmetry. Our approach is the first semi-supervised method that detects 3D landmarks from 2D palsy images.
- The novel integration of in-domain GAN inversion in the self-supervised stage, to enhance the detection of the facial landmarks.
- ParFace, the first dataset for 3D face alignment on persons with palsy. ParFace and our source code is publicly available for research purposes..
- A novel facial landmark hybrid tracking pipeline using grouped landmark Kalman Filters to capture the inter-landmark correlations in the filter state.
- An adaptive Kalman Filter measurement noise derived from the uncertainty of the DNN predictions.
- A mechanism to override the face detection in every frame, based on the behaviour of the Kalman Filter state covariance.

The proposed approaches are supported by multiple experiments and evaluation on public face alignment and tracking datasets, in addition to a newly introduced facial palsy dataset, with improvements w.r.t. the state of the art in 3D face alignment.

4.2 Related Work

In this Section, we review the relevant work on 2D and 3D face alignment, facial landmark detection for palsy and 3D landmark tracking. For a detailed survey on 2D alignment and tracking, we refer the reader to [172] and [173], respectively.

4.2.1 2D Face Alignment

These methods consider the face as a 2D object and detect only visible landmarks. Classical approaches use Active Appearance Models (AAMs) [174, 175, 176], Active Shape Models (ASMs) [177, 178, 179], Constrained Local Models (CLMs) [144, 180] and Cascaded Regression Methods (CRMs) [181, 148, 182, 183]. Since 2D landmarks do not maintain a one-to-one correspondence across large head poses, they are not robust against extreme head rotations.

More recently, several DNN architectures have been proposed for this task. They can be categorized as coordinate-regression or heatmap-based methods. The former includes architectures that regress the mapping between the image and the 2D landmark coordinates and the latter regresses heatmaps for every landmark. Coordinate-based methods are more computationally efficient, while heatmap-based approaches usually have higher accuracy [170].

Coordinate-based methods span DAN [184], a CNN based on cascaded regressors, which uses heatmaps in intermediate stages; LAB [163], which exploits the geometric information in the face boundary, by first detecting boundary heatmaps with adversarial learning, and then regressing the landmarks; DeCaFa [185], a deep convolutional cascade of U-Net [50], which integrates landmark attention maps; and DTLD [186], which uses a cascaded deformable transformer with self-attention.

Heatmap-based methods include SAN [187], that explores the variance of images styles in facial landmark prediction, by training with styled images from a generative adversarial module; HSLE [188], a hierarchical structured which searches the most robust patterns to model the structure of the facial landmarks; AVS [189], which investigates the latent space in face alignment to disentangle style and augment the training data; HRNetV2 [190], an extension to the HRNet architecture, which integrates multi-level representations from the parallel convolutions; LUVLi [191], which introduced a cascaded U-Net with additional modules to estimate the uncertainty and visibility of the landmarks; 3FabRec [171], a reconstruction-based pipeline that learns a representation in the latent space of a GAN, and integrate an additional ResNet [51] for heatmap regression; PIPNet [170], an architecture that regresses a compact representation of the heatmaps, by overriding the upsampling layers and incorporates neighboring pixel predictions; SHR-Fan [192] a pipeline that targets heatmap discretization error and propose a novel encoding-decoding scheme, with a Siamese-based training; H3R [193], which proposes randomized rounding to target sub-pixel localization, improving the quantization error due to

decoding landmark coordinates from heatmaps; and LDEQ [194], an adapted deep equilibrium model (DEQ) for facial landmark localization, which reduces flickering on video sequences.

4.2.2 3D Face Alignment

These methods integrate 3D face models, either implicitly or explicitly, to recover a sparse or a dense set of 3D facial landmarks. Some approaches jointly perform face alignment and reconstruction, with the aid of 3DMMs and large datasets of 3D faces [159, 195, 169, 86, 196, 197, 198, 199, 200, 201, 202, 13].

Classical approaches such as [195], register a 3DMM to a face image, given a model composed of a sparse point cloud and 2D texture patches for each point. The alignment is formulated as a Bayesian inference problem and is solved using the Expectation-Maximization algorithm. CRMs have been extended to 3D face alignment as well [196, 197, 203].

Recent 3D face alignment methods can be grouped in two main categories: (a) model-free [78, 87, 161, 73, 74] and (b) model-based approaches [159, 77, 199, 13, 169, 201, 202]. In the latter, the alignment is aided by a 3D model, such as 3DMM. These methods are less sensitive to large rotations and occlusion in terms of pose [75], but have shown poor generalization capabilities when data is low in quantity or variability [204], particularly for unseen facial expressions and shapes. On the contrary, model-free approaches are more flexible, but suffer in cases with (self-)occlusion.

FAN [73] consists of stacked hourglass (HG) for 2D face alignment and an additional sub-network based on ResNet to estimate the depth. [78] and [74] regress a volumetric representation of the face from stacked hourglass networks (SHN). JVCR [74] additionally integrates additional sub-network to retrieve the 3D coordinates from the voxels. [205] proposes to leverage CLMs, where a CNN-based local detector exploits the advantages of mixture of experts. The detection is performed with two additional components, an Adjustment Network and a Correction Network. [161] incorporates an attention mechanism from a spatial transformer to the regression pipeline network in [206], to refine the landmark detection in eyes, irises and lips. [207] exploits the generative ability of StyleGAN2 [208] to detect 3DA-2D landmarks. The generator is modified based on [171].

PRN [77] uses an encoder-decoder to regress the 3D face from a 2D representation denoted as UV position map, given an input image. 2DASL [199] proposes a self-supervised-learning scheme to integrate datasets with only 2D or 3D annotations. [202] employs autoencoders to model non-linear 3DMMs, by learning joint texture and shape. 3DSTN [204], a spatial transformer network, uses a generic 3D model along with Thin Plate Spline warping to handle unseen faces. [209] proposes to model a non-linear 3DMM from in-the-wild face images without 3D scans, using encoder-decoders. [210] introduces a queried landmark predictor, allowing detection of 3D landmark configurations using a 3D face model reference. [211] proposes a multi-view consistent pipeline for landmark detection that leverages a multi-view dataset built using Neural Radiance Field (NeRF) [212].

[75, 87] leverage the 3D landmarks to fit a 3DMM, while [159, 13, 169] derive the landmarks from the reconstructed face. [87] proposes a dense face alignment approach, where the land-

marks are extracted with a CNN and used to fit a 3DMM via optimization. The landmarks cover the head, in addition to the ears, eyes and teeth. [159, 13, 201] propose analysis-by-synthesis approaches with dedicated networks to learn the 3DMMs parameters. [13] uses cascades of CNN regressors, while [169] uses a backbone based on MobileNet [213], with additional layers for landmark regression and regularization. [159] introduces a UV displacement map, which models dynamic features such as wrinkles and SynergyNet [201] uses [169] for 3DMM fitting. The landmarks are then extracted from the model and refined with multi-attribute feature aggregation.

4.2.3 Face Alignment for Palsy

Facial palsy assessment is performed either by segmenting susceptible regions of the face, such as eyebrows, eyes, nostrils and mouth [162, 214, 215, 216], by locating the muscle activation and exploiting action units (AUs) [217, 218], or by directly detecting facial landmarks. These landmarks have been used as well to locate AUs [218] or face regions heuristically [219, 215] or with more elaborated methods as in [162, 220, 214].

Face alignment for palsy assessment can be divided in two categories, 2D and 3D landmark-based. 3D-based methods usually compute the landmarks from multi-camera systems [221, 222] and 3D sensors such as Kinect [223], deterring their implementation in a clinical setting. 2D landmarks, on the other hand, are extracted from gray-scale or RGB images, captured from easily accessible cameras in smartphones [166], web [167] or digital cameras [224].

Relevant to this work are pipelines based on 2D images, which have been achieved with AAMs [225, 226], ASMs [227], CLMs [164], CRMs such as supervised descent method (SDM) [167], a parallel cascade of linear regressors [166] from [181], an ensemble of regression trees [164, 228, 165, 214, 229] from [148], supervised face alignment networks such as FAN [164, 230, 162], DAN [231], SAN [219] and other DNN [220, 226]. In most cases, the alignment is performed with models that have been trained on images with healthy subjects, with very few or no samples of large asymmetrical facial expressions, limiting their scope. The bias in face alignment on palsy can be tackled with incremental learning on discriminative models such as [181], and retraining or fine-tuning existing regression-based pipelines [165] or face alignment networks, as in [162, 164, 231], with dedicated datasets from the target population, via transfer learning. Note that previous works on palsy face alignment use supervised approaches, while our method is semi-supervised and does not require large labeled datasets with asymmetrical expressions.

4.2.4 3D Landmark Tracking

Classical approaches for 2D face alignment, such as CRMs [232, 85, 75] and CLMs [118] have been extended to 3D landmark tracking. While [232] does not work on still images, since it needs an approximate ground truth shape to predict the landmarks, [118] was aided by a 3D depth sensor. Some methods use a tracking-by-detection approach [75, 76, 233], where a face detector extracts the face in every frame before the face alignment. [75] registers a dense 3D shape frame by frame, using a cascade regression framework trained on a large dataset of 3D

face scans.

DNN architectures use CNN regressors to either estimate the parameters in 3DMMs [76] or use model-free schemes [233, 234]. [76] introduced an updated U-Net to estimate frame by frame dense 3D landmarks, that are then smoothed over consecutive frames. [233] combine local heatmap with global shape regression to design a local-to-global pipeline. A face detector along with the mean shape of 3D landmarks are employed to initialize every frame. [234] proposes a multi-view SHN, where the bounding box from the previous frame is used to initialize the next frame. Based on Re^3 [235], [236] introduced a tracking approach which integrated the temporal dependency with Long Short Term Memory (LSTM).

Some prior works explored the idea of combining 2D face alignment with statistical filtering. [237] and [238] propose to use a Kalman Filter per landmark and a constant acceleration model. [237] uses Active Shape Model for face alignment and additionally introduced an approach for head pose tracking with Kalman Filter. [238] included a 3D model for aiding the landmark tracking. Nonetheless, these methods are based on 2D landmark tracking, which are inconsistent across different poses, *e.g.* for profile and frontal faces.

In contrast to the prior work, we introduce for the first time a 3D face alignment approach using Kalman Filter. Additionally, we propose the novel concepts of grouped-landmark Kalman Filters and face alignment uncertainty integration into the Kalman Filter measurement update.

4.3 Face Alignment on Symmetrical and Asymmetrical Expressions

Face alignment on asymmetrical expressions has been overlooked in the literature, particularly for unusual gestures observed in individuals with unilateral facial paralysis. In this Section, we explore in-domain inversion in a semi-supervised approach for face alignment and target the detection of 3D landmarks on symmetrical and extremely asymmetrical facial expressions due to paralysis. Our approach first leverages unlabeled face data to synthesize face images, while learning a compressed representation in the latent space. Then, it integrates in-domain inversion in the self-supervised stage, to make the latent space semantically meaningful. This is exploited in the supervised stage by a 2D face landmark detector, trained on labeled data. Finally, we extend the pipeline to 3D face alignment and regress the depth coordinate from the intermediate latent space and the predicted 2D landmarks. We evaluate and compare our method to related work on publicly available datasets, and demonstrate that our approach outperforms the state of the art in the detection of 3D facial landmarks in our newly introduced dataset of facial paralysis, ParFace.

4.3.1 Overview of the Proposed Face Alignment Approach

We explore the semantically meaningful latent space in a reconstruction-based deep learning architecture, to improve the detection of facial landmarks in faces with a varying range of expressions. The proposed architecture ParFace-Net is shown in Figure 4.3. In the self-supervised stage, an autoencoder (AE) is trained with unlabeled face datasets, where the en-

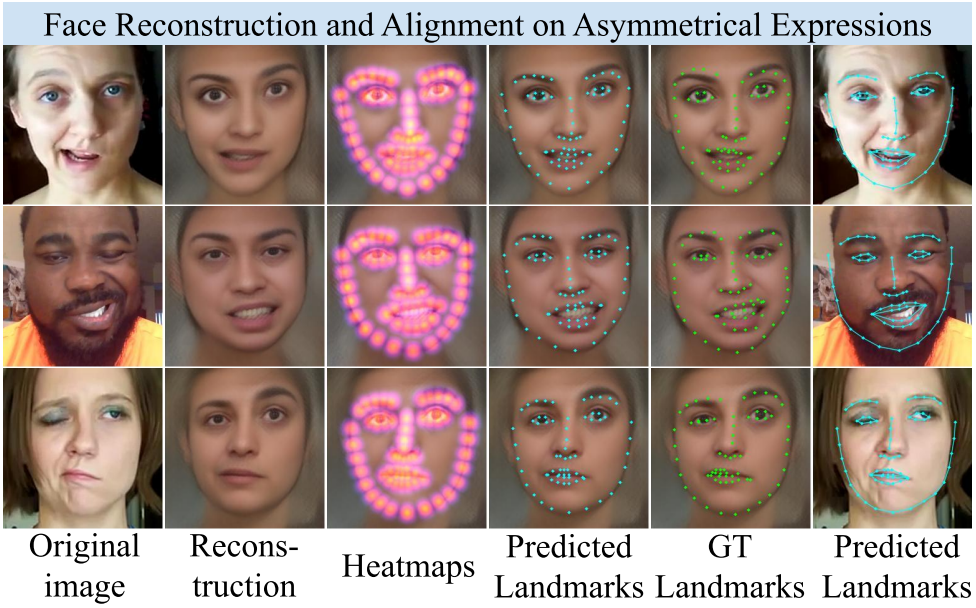


Figure 4.2: Face alignment on asymmetrical expressions. Our approach learns to synthesize faces from unlabeled datasets and exploits the latent code to predict the landmarks.

coder E learns the mapping from the input data to a low-dimensional intermediate vector z. This latent code is further enforced to be semantically meaningful, through the feature disentanglement process introduced by in-domain inversion. This is achieved by means of the discriminator D and adversarial training on the E and D, while the decoder G is frozen. In the supervised stage, a 2D landmark detector learns to regress 3DA-2D landmark heatmaps from the semantically rich latent code, which in turn are used to predict the depth coordinate. We further fine-tune the encoder with the gradients from the landmark heads to improve the results.

4.3.2 Self-Supervised Stage

This stage consists of an adversarial AE, trained on large-scale face datasets. The encoder E learns to capture the most important facial attributes in an intermediate latent vector z, while the decoder G is posed as a generator of a GAN that reconstructs the original image from the latent code. The AE is trained on a combination of three losses, as follows:

$$\mathcal{L}_{AE} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{adv}\mathcal{L}_{adv}, \tag{4.1}$$

where $\mathcal{L}_{rec}$ is the reconstruction loss, given by the $L1$ or $L2$ pixel-wise distance between the input x and the reconstruction $\hat{x}$ at $G(E(x))$; $\mathcal{L}_{perc}$ is the perceptual loss [239], in eq. (4.2); $\mathcal{L}_{adv}$ is an adversarial image loss [61], which enforces the AE to produce realistic faces based on the output from D; and $\lambda_{(.)}$ is the respective weight of each loss.

$$\mathcal{L}_{perc}(x, \hat{x}) = \sum_{i}^{\phi} \frac{1}{C_i H_i W_i} ||V_i(\hat{x}) - V_i(x)||_2^2. \tag{4.2}$$

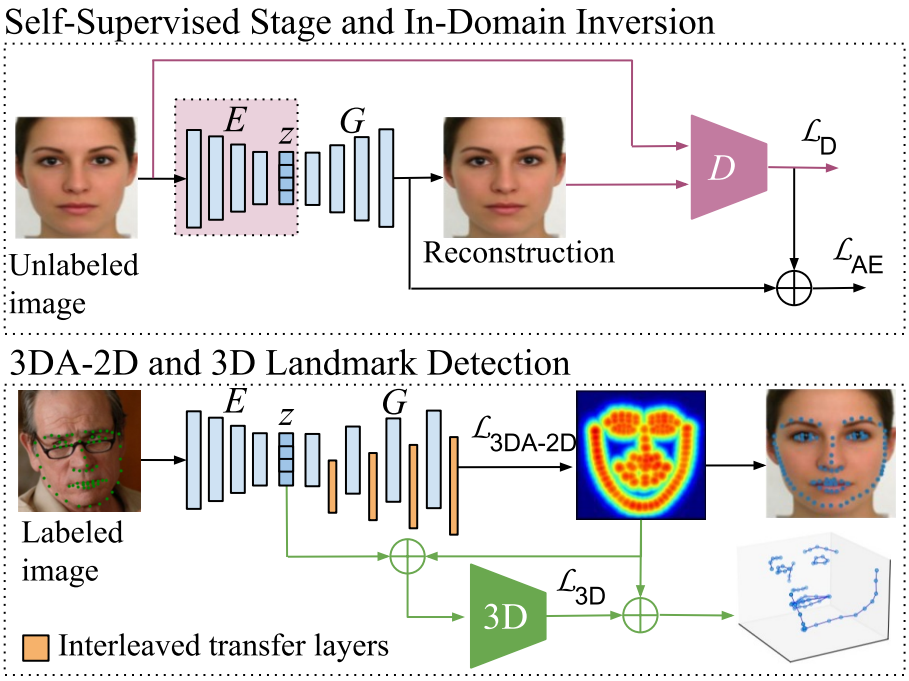


Figure 4.3: Architecture of ParFace-Net (PF-Net).

C_i, H_i, and W_i are the depth, height and width of the feature map $V_i(\cdot)$ at layer i of a VGG network [240]; x and $\hat{x}$ are the input and reconstructed images; and ϕ is the set of layers from the VGG.

We introduce a discriminator D during the reconstruction phase, trained with the Wasserstein Loss with Gradient Penalty (WGAN-GP) [241], formulated as

$$\mathcal{L}_D = \mathbb{E}[D(\hat{x})] - \mathbb{E}[D(x)] + \frac{1}{2}\gamma\mathbb{E}[\nabla D], \tag{4.3}$$

where the last term is the gradient regularization and the hyper-parameter $\gamma = 10$. The AE and D are trained using a similar procedure to GANs, alternating gradient updates.

4.3.3 In-Domain Inversion

We leverage the generative capabilities of the AE by incorporating in-domain GAN inversion. Inspired by Zhu *et al*. [242], we follow a domain-regularized approach that pushes the encoder to create latent code in the semantic domain. In [242], this module enables semantic editing of facial attributes such as expression and pose, while an additional optimization stage improves the reconstructed face in the pixel level. Unlike [242], our approach does not seek to edit facial attributes nor aims to create a faithful reconstruction of the face. Instead, we propose to encode facial attributes in the latent vector that boost the alignment in the landmark detectors for a wide range of expressions.

The inversion is achieved in [242] by introducing a domain-guided encoder to the GANs-based formulation. We instead exploit the pre-trained encoder E from the previous step, as shown in red in Figure 4.3. The discriminator D is then used to compete with E, which acts

as the domain-guided encoder and refines the latent space z to be aligned with the semantic latent space of the reconstruction process. During this stage, the decoder G is fixed, and E and D take turns to train with the loss functions in eq. (4.1) and (4.3), respectively. To that end, the same unlabeled data as in the self-supervised stage is used, where the input of E corresponds to the image x, and the input of D is given by x and the reconstruction $\hat{x}$.

The asymmetrical features in the latent code from palsy patients are refined in this stage, without affecting the reconstruction of symmetrical faces. Hence, the same trained model could be used to align the landmarks during different levels of palsy, to continuously track the recovery.

In contrast to [242], we do not apply the final optimization step to enhance the output of the reconstruction, since we do not aim to create an accurate reconstruction in the pixel level.

4.3.4 Supervised Stage

The supervised stage is composed of a 3DA-2D and a 3D landmark detector, where all the face information learned in the self-supervised stage and refined through the in-domain inversion module is available for generalized usage across various landmark datasets.

3DA-2D Landmark Detector

In this stage, the landmark detector maps the disentangled latent code z to 2D heatmaps that represent the probability map of each landmark location. During training, the parameters of the autoencoder are fixed and the layers of the decoder G are interleaved with 3×3 convolutional layers, inspired by 3FabRec [171]. The last convolutional layer that produces the face image is then superseded by a convolutional layer that provides the heatmaps, as shown in Figure 4.3.

We propose to adopt the adaptive wing loss (AWing) [243] as the heatmap prediction loss, instead of the mean squared error (MSE) from [171]. Since background pixels on a heatmap dominate over foreground pixels, this loss function penalizes small errors on foreground pixels while tolerating small errors on background pixels. It is formulated as

$$\mathcal{L}_{2D}(h, \hat{h}) = \begin{cases} \omega \ln\left(1 + \left|\frac{h-\hat{h}}{\epsilon}\right|^{\alpha-h}\right) & \text{if } |(h-\hat{h})| < \theta \\ A|h-\hat{h}| - C & \text{otherwise,} \end{cases} \tag{4.4}$$

where h and $\hat{h}$ denote the ground truth and predicted heatmap pixel values, and ω, θ, α, and ϵ are positive values. A and C are added to smooth the loss function at $|h-\hat{h}| = \theta$.

3D Landmark Detector

We introduce a 3D landmark detector to regress the depth coordinate of the 3DA-2D landmarks. It takes as input the concatenation of the intermediate latent vector and the predicted 3DA-2D landmark heatmaps. This detector uses the MSE loss, defined as

$$\mathcal{L}_{3D}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{4.5}$$

where N is the number of landmarks, and y and $\hat{y}$ are the ground truth and predicted depth values, respectively.



Figure 4.4: Samples images of ParFace.

Encoder Fine-tuning

This strategy, introduced in [171], proposes to further optimize the encoder E along with the Interleaved Transfer Layers (ITL) in tandem. The fine-tuning encourages the encoder to encode more features in the latent code that enhance the landmark predictions. By integrating this step, the identity of the reconstructed face no longer resembles the original image and instead tends towards an average face, as shown in Figure 4.2. Nonetheless, other attributes such as the expression and pose are enhanced.

4.3.5 Datasets

ParFace

We introduce ParFace, the first dataset on palsy face alignment with 3D landmarks annotations in video sequences. We collected 28 videos from YouTube of 150 frames each, where the subjects are usually talking to the camera or making a wide range of facial expressions. The videos have varying resolution and cover a wide range of ages, ethnicity, poses, illumination settings and backgrounds. We provide 68 landmarks annotations for 1350 frames in 9 videos, for a total of $\sim$92K landmarks.

We developed an annotation tool, which provides an initial 3D landmark estimation by 3D-FAN [73]. Since 3D-FAN was trained on datasets without palsy, each landmark was manually refined to match the asymmetrical facial expressions and provide high quality annotations. This refinement affected most of the 3DA-2D landmarks, and less the depth coordinate. Some sample images are shown in Figure 4.4. The dataset, along with the source code of the proposed architecture is publicly available for research purposes.

The annotated set of ParFace can be used as a benchmark to evaluate palsy alignment, as in Section 4.3.9, or to fine-tune semi- or fully supervised approaches as in Section 4.3.11. The unlabeled set of ParFace can be used for training semi- or self-supervised architectures, similarly to ParFace-Net.

Datasets for Face Alignment

ParFace-Net is trained on well-known public datasets on face analysis. The datasets, annotation types and in which stage they were used are listed in Table 4.1. In the self-supervised stage and during the in-domain inversion, the autoencoder is trained with multiple datasets without any type of landmark annotations. We introduced palsy datasets in these stages, such as the Toronto NeuroFace, MEEI and the unlabeled set of ParFace.

The 3DA-2D and 3D landmark detectors are trained with 300W-LP. We separately train the 3DA-2D detector with 2D landmarks, to investigate the performance on 2D face alignment.

Table 4.1: Publicly available datasets used for training the autoencoder (AE), the 2D and 3D landmark detectors, and for testing the current model. Some datasets such as HELEN, LFPW and WFLW are split into a training and a test set.

Dataset	Images / Frames	Facial landmarks			Train			Test		
		2D	3DA-2D	3D	AE	2D	3D	AE	2D	3D
CelebA [244]	202599	✓	-	-	✓	-	-	-	-	-
AffectNet [245]	291650	✓	-	-	✓	-	-	-	-	-
Menpo 2D [82]	8954	✓	✓ [89]	✓ [89]	✓	-	-	-	-	-
LS3D-W [73]	7200	-	✓	-	✓	-	-	-	-	-
FFHQ [246]	70000	-	-	-	-	-	-	✓	-	-
NeuroFace [164]	3306	-	-	-	✓	-	-	-	-	-
MEEI [224]	12050	-	-	-	✓	-	-	-	-	-
ParFace (Ours)	4200	-	-	✓	✓	-	-	-	-	✓
300W-LP [13]	61225	✓	-	✓	✓	-	✓	-	-	-
AFW [247]	337	✓ [80]	✓ [89]	✓ [89]	-	✓	-	-	-	-
HELEN [248]	2330	✓ [80]	✓ [89]	✓ [89]	-	✓	-	-	✓	-
LFPW [249]	1035	✓ [80]	✓ [89]	✓ [89]	-	✓	-	-	✓	-
300W [79, 80, 81]	600	✓	✓ [89, 73]	✓ [89]	-	-	-	-	✓	-
iBUG [79, 80, 81]	135	✓	✓ [89]	✓ [89]	-	-	-	-	✓	-
WFLW [163]	10000	✓	-	-	-	✓	-	-	✓	-
AFLW2000-3D [13]	2000	✓	-	✓	-	-	-	-	-	✓

4.3.6 Implementation Details

ParFace-Net was implemented in Python using PyTorch. The self-supervised stage was trained on a Nvidia A100, while the face alignment networks were trained on a Nvidia RTX2080-Ti.

The autoencoder takes as input a cropped version of the face. For labeled datasets, we use the ground truth landmarks to compute the bounding box, following related works. For datasets such as FFHQ that have no landmark annotation, we use the MTCNN face detector [70], described in Section 2.4.1, to crop the face. Faces with a height less than 100px are discarded. The data is augmented with random horizontal flipping (50%), translation ($\pm 4\%$), scale jittering (94% to 103%) and rotation (between $\pm 45°$).

Model Architecture

The autoencoder consists of a ResNet-18 [51], which encodes a 99-dimensional latent vector, and an inverted ResNet-18 [250] for decoding, based on [171]. The perceptual loss uses layers $\phi = [3, 8, 15, 22]$ of a VGG-19 [240] pre-trained on ImageNet [251]. In the supervised stage, the 3DA-2D landmark detector is an inverted ResNet-18 that outputs landmark heatmaps of size 128 × 128 with N channels, where N is the number of landmarks. The 3D landmark detector is a ResNet-18, which regresses the depth. This coordinate is normalized to lie between [-1, 1] to achieve faster convergence and numerical stability.

Training Details

We use the Adam optimizer [252] with a learning rate of 2e-5, $\beta_1 = 0.0$ and $\beta_2 = 0.999$. The autoencoder is trained with input and output images of size 256x256. We train for 50 epochs with eq. (4.1), where $\mathcal{L}_{rec}$ is the L2 loss, followed by 50 epochs with the L1 loss as $\mathcal{L}_{rec}$. After that, we fix the decoder G and optimize the encoder E for feature disentanglement against the discriminator D with the L2 loss as the $\mathcal{L}_{rec}$, for 50 epochs.

The 3DA-2D landmark detector is trained for 100 epochs to predict the heatmaps. We fine-tune the encoder with gradients from the landmark head for 100 epochs. A similar procedure is followed in the experiments to train with 2D landmarks. For 3D face alignment, the 3D landmark detector is trained with the ground truth 3DA-2D landmark heatmaps for 50 epochs.

4.3.7 Evaluation Metrics

Following the standard protocol, we adopt the normalized mean error (NME) to evaluate 3DA-2D face alignment on ALFW2000-3D and ParFace. We additionally report the failure rate (FR) and area under the curve (AUC) at 10% of the Cumulative Error Distribution (CED) on ParFace.

3D face alignment is evaluated using the ground truth error (GTE) on AFLW2000-3D and ParFace. The GTE is equivalent to the NME, but evaluates the full 3D coordinates. The GTE is normalized by the inter-ocular (IO) distance, while the NME is normalized by the square-root of the bounding box size enclosing the landmarks, following related works. The standard deviations σ of the NME and GTE in ParFace are also reported. All the metrics are defined in Section 2.5.

4.3.8 Results on 3DA-2D and 3D Face Alignment

3DA-2D and 3D face alignment are evaluated on the widely used benchmark AFLW2000-3D, where the landmark detectors are trained using 300W-LP. We employ the AE with in-domain inversion to train the landmark detectors. Furthermore, we refine the 3DA-2D landmarks with encoder fine-tuning. The results are shown in Figure 4.5 and in Table 4.2.

We observed that our models outperform the SOTA in 3DA-2D face alignment (NME) for frontal and near frontal faces (0 to 30°) and our ParFace-Net with the Awing loss has the 2nd best GTE on 3D face alignment for the reported methods. For larger poses, we noticed a decreased performance of ParFace-Net. This could be attributed to the small portion of non-frontal faces in the self-supervised stage, where face semantics are learned mostly for near-frontal poses. We also observed that model-based approaches tend to be more robust to large



Figure 4.5: 3DA-2D and 3D Face alignment on AFLW2000-3D. Ground truth in red and predictions in green.

Table 4.2: NME (3DA-2D) and GTE (3D) on AFLW2000-3D, for different yaw angles. † reported in [74]. Methods with * are semi-supervised. The NME for 'All' is averaged on the 3 categories.

	Method	NME↓				GTE ↓
		0° to 30°	30° to 60°	60° to 90°	All	
Model-based	3DDFA [13]	2.84	3.57	4.96	3.79	-
	SPDT [198]*	3.56	4.06	**4.11**	3.88	-
	3DDFA_V2 [169]	2.63	3.42	4.48	3.51	-
	2DAL [199]*	2.75	3.46	4.45	3.55	-
	SADRNet [200]	2.66	**3.30**	4.42	3.46	-
	SynergyNet [201]	2.66	**3.30**	4.27	**3.41**	-
Model-free	SDM [203]†	3.67	4.94	9.76	6.12	-
	3D-FAN [73]†	2.77	3.48	4.60	3.62	7.45
	JVCR [74]	2.94	3.46	4.53	3.64	**7.28**
	StyleGAN-FA[207]*	2.65	3.62	4.89	3.72	-
	PF-Net$_{MSE}$*	2.62	3.65	4.80	3.69	7.42
	PF-Net$_{Awing}$*	**2.61**	3.67	4.74	3.67	7.38

head poses, since they are trained with additional 3DMM parameters such as head orientation and face shape. However, as shown in the next section, model-based methods do not cope well with a wide range of facial expressions, including asymmetrical expressions.

4.3.9 Evaluation on ParFace

The annotated set of ParFace is employed to evaluate our models from Section 4.3.8, trained without annotations for palsy. The results are shown in Table 4.3. We additionally report the performance of different SOTA model-based and model-free methods for 3D face alignment, which have been trained on 300W-LP or 3DMMs datasets. To discard alignment errors due to face detection inaccuracies, we replaced the face detectors in every method and provided the bounding box from the ground truth landmarks, to crop the input images.

The CED curves for the normalized 3DA-2D and 3D RMSE are shown in Figure 4.6. Our models achieve the lowest NME and FR, the highest AUC and the 2nd and 3rd lowest GTE on ParFace. As mentioned in Section 4.3.5, for labelling ParFace, an initial landmark prediction was computed using 3D-FAN. However, the 3DA-2D landmarks were heavily refined, while the z coordinates were refined to a lesser extent. As expected, 3D-FAN has the lowest GTE in this dataset. Qualitative results are shown in Figure 4.7 and Figure 4.8. Table 4.3 and Figure 4.7 show that model-free methods have in general a better performance and are more flexible on asymmetrical expressions than model-based pipelines.

Table 4.3: Performance metrics on ParFace. The NME, AUC and FR evaluate 3DA-2D landmarks, while GTE evaluates 3D alignment.

	Method	NME $\pm \sigma^{\downarrow}$	$AUC_{10}^{\uparrow}$	$FR_{10}^{\downarrow}$	GTE $\pm \sigma^{\downarrow}$
Model-based	3DDFA_V2 [169]	4.65 $\pm$ 2.51	55.14	1.11	11.18 $\pm$ 5.98
	SynergyNet [201]	5.49 $\pm$ 4.96	49.48	1.70	13.89 $\pm$ 8.48
	DECA [159]	4.24 $\pm$ **1.19**	57.81	0.88	23.27 $\pm$ 6.27
Model-free	3D-FAN [73]	4.26 $\pm$ 3.64	59.83	0.96	**7.06** $\pm$ 6.20
	JVCR [74]	4.00 $\pm$ 2.79	61.87	1.26	8.79 $\pm$ 4.91
	PF-Net$_{MSE}$	3.83 $\pm$ 3.19	62.74	0.96	8.14 $\pm$ 4.60
	PF-Net$_{AWing}$	**3.79** $\pm$ 2.98	**62.81**	**0.82**	8.05 $\pm$ **4.37**

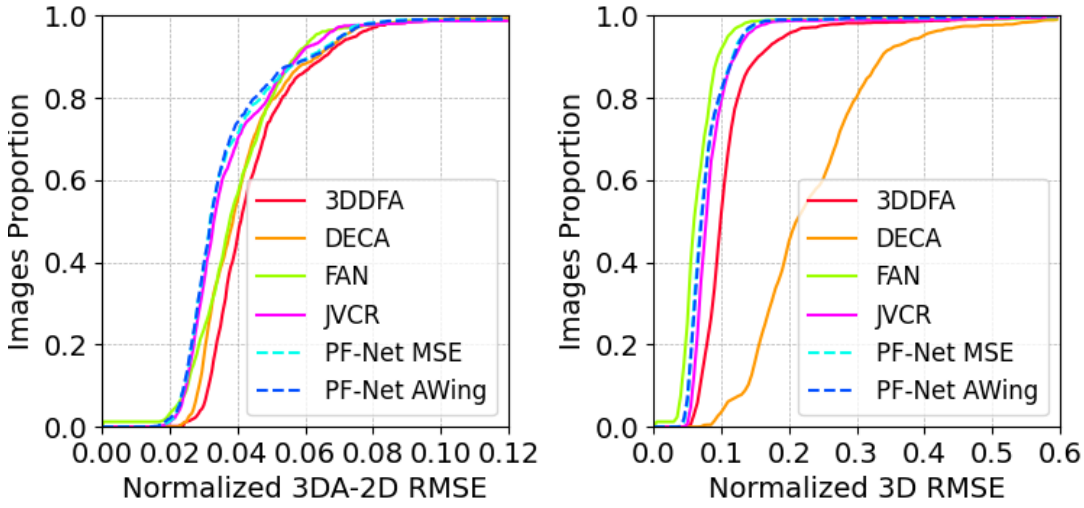


Figure 4.6: CED curves for 3DA-2D and 3D face alignment of models tested on ParFace. Our models are 'PF-Net: MSE' and 'PF-Net: Awing', represented with dotted lines.

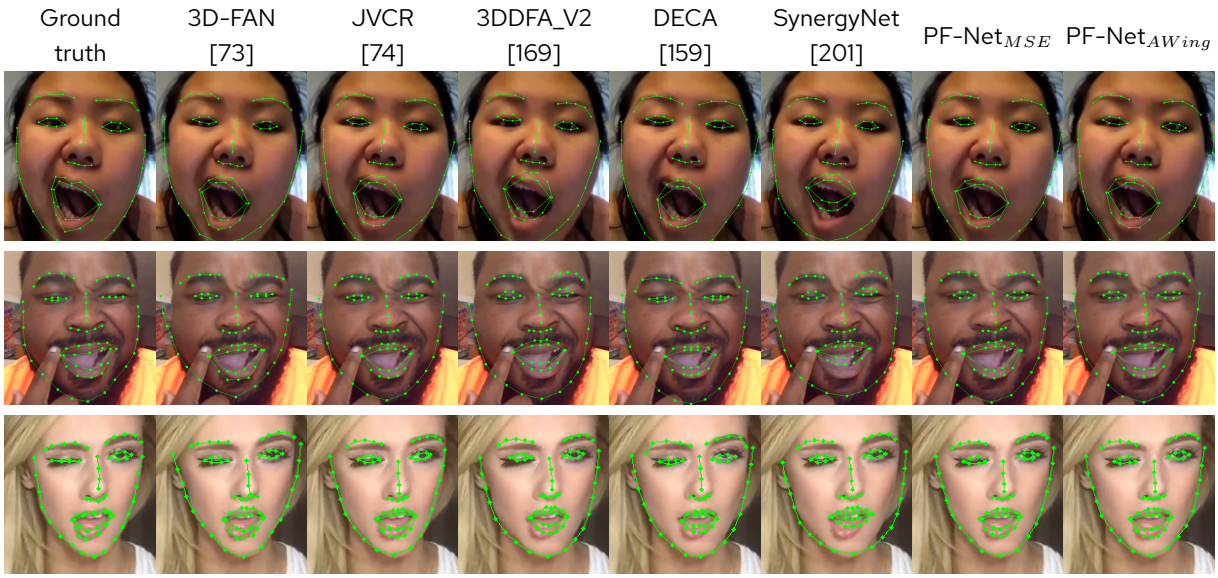


Figure 4.7: Face alignment on ParFace. 3D-FAN, JVCR, 3DDFA_V2 and SynergyNet were trained with 300W-LP, while DECA uses pseudo ground truth from 3D-FAN. 3DDFA_V2, DECA and SynergyNet are trained on datasets for face reconstruction to fit 3DMMs as well.

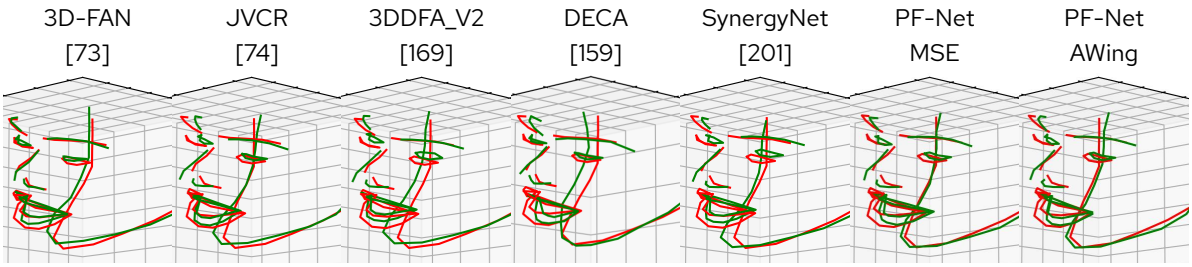


Figure 4.8: 3D face alignment on ParFace with different methods of the SOTA. Ground truth in red and predictions in green.

4.3.10 Runtime and Model Parameters

We measured for ParFace-Net a runtime of ∼230FPS on average for 1K repetitions, for 2D and 3DA-2D face alignment, on a Nvidia RTX2080-Ti. To estimate the full 3D coordinates, ParFace-Net runs at ∼156FPS.

ParFace-Net is composed of two ResNet-18 and an inverted ResNet-18, with ∼24.14M parameters. 3D-FAN is composed of 4 HG networks with ∼24M parameters and a ResNet-152 with ∼58.5M parameters to compute the depth coordinate (in total ∼82.5M). Likewise, JVCR uses 4 stacked HG and an additional network to map the voxels to coordinates, with 32.47M parameters in total. SynergyNet has 4.6M parameters, 3DDFA_V2 3.27M and DECA uses two ResNet-50 with more than 25M parameters each and multiple decoders to retrieve the parameters of the 3DMM.

4.3.11 Ablation Study

We evaluate the contribution of the in-domain inversion module, the AWing loss and unlabeled data in the face alignment process. We additionally fine-tune the models on ParFace, to analyze the performance of our approach.

Training with In-Domain Inversion

We investigated the effect of the in-domain inversion module as well. To that end, we trained the landmark detectors before and after the in-domain inversion is applied. The results for 3DA-2D and 3D face alignment are shown in Table 4.4. We observed that in-domain inversion boosted the performance in every metric w.r.t. the model without inversion. The improvement is more noticeable for ParFace, both in the NME and GTE. Based on these results, we confirm the improvement of the landmark detection task by integrating the in-domain inversion module, both for symmetrical and asymmetrical faces.

In order to visualize the refinement added by the in-domain inversion (IDI) module to the self-supervised stage, we present qualitative results on 3DA-2D face alignment in Figure 4.9. The models used for these experiments were trained with the AWing loss and encoder fine-tuning.

The challenging images in Figure 4.9 are part of the validation set of 300W-LP, used to train the landmark detectors. A detailed inspection in the area of the jaw and lips shows the improvements added by the in-domain inversion module, where the contours are followed more closely. This could be attributed to an improved representation of the pose and expression in the latent code, which in turn is exploited by the landmark detectors.

Effect of AWing Loss

We additionally examined the performance of the 3DA-2D landmark detector using the MSE loss and the proposed AWing loss. The results are reported in Table 4.4. During the experiments, we observed that the MSE loss converged faster, but in overall the AWing loss leads to improved accuracy in most of the metrics. We hypothesize that this is due to AWing loss being more sensitive to foreground pixels than background pixels, considering that background pixels predominate in the heatmaps.

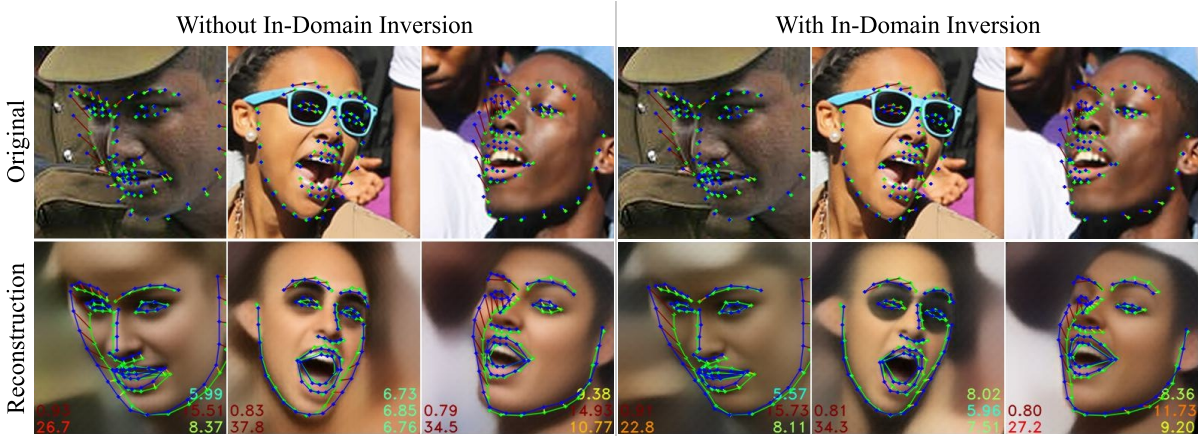


Figure 4.9: 3DA-2D landmarks detected with and without in-domain inversion (IDI) with the AWing loss. Green (ground truth) and blue (prediction). The original and reconstructed images are shown.

Table 4.4: Ablation study on AFLW2000-3D and ParFace. We evaluate the impact of the self-supervised stage, the in-domain inversion module and the loss function for landmark detection. The NME and GTE are averaged in each dataset on the total number of images.

Self-Supervision	In-Domain Inversion	Alignment Loss	AFLW2000-3D NME $\pm \sigma$ ↓	AUC$_{10}$ ↑	FR$_{10}$ ↓	GTE $\pm \sigma$ ↓
-	-	MSE	3.91 ± 3.94	65.31	3.56	8.81 ± 6.87
✓	-	MSE	3.23 ± **2.90**	69.40	1.50	7.77 ± 6.22
✓	✓	MSE	3.15 ± 2.96	70.17	1.50	7.42 ± 5.33
-	-	AWing	3.96 ± 3.97	64.81	3.65	9.03 ± 6.93
✓	-	AWing	3.19 ± 2.95	69.78	1.45	7.45 ± 5.07
✓	✓	AWing	**3.14** ± 2.92	**70.22**	**1.35**	**7.38** ± **5.05**
Self-Supervision	In-Domain Inversion	Alignment Loss	ParFace NME $\pm \sigma$ ↓	AUC$_{10}$ ↑	FR$_{10}$ ↓	GTE $\pm \sigma$ ↓
-	-	MSE	4.24 ± 3.27	58.39	1.11	9.16 ± 4.89
✓	-	MSE	3.96 ± 3.34	61.75	0.96	8.40 ± 4.87
✓	✓	MSE	3.83 ± 3.19	62.74	0.96	8.14 ± 4.60
-	-	AWing	4.46 ± 3.56	56.17	1.04	9.47 ± 5.27
✓	-	AWing	3.94 ± 3.21	61.76	0.96	8.41 ± 4.75
✓	✓	AWing	**3.79** ± **2.98**	**62.81**	**0.82**	**8.05** ± **4.37**

Table 4.5: Ablation study on 3DA-2D and 3D face alignment after training the self-supervised stage with portions of the data. The AE in * was trained only with palsy data.

%	AFLW2000-3D				ParFace			
	NME $\pm \sigma^{\downarrow}$	$AUC_{10}^{\uparrow}$	$FR_{10}^{\downarrow}$	GTE $\pm \sigma^{\downarrow}$	NME $\pm \sigma^{\downarrow}$	$AUC_{10}^{\uparrow}$	$FR_{10}^{\downarrow}$	GTE $\pm \sigma^{\downarrow}$
0	3.96 ± 3.97	64.81	3.65	9.03 ± 6.93	4.46 ± 3.56	56.17	1.04	9.47 ± 5.27
1	3.57 ± 3.36	67.31	2.45	8.20 ± 5.94	4.20 ± 3.42	58.87	1.19	9.06 ± 5.15
3*	3.58 ± 3.24	66.97	2.65	8.16 ± 5.59	4.19 ± 3.31	58.86	0.89	9.06 ± 4.94
13	3.35 ± 3.09	68.94	1.70	7.70 ± 5.42	3.84 ± 2.85	61.89	0.89	8.65 ± 4.34
34	3.26 ± 2.94	69.35	1.55	7.55 ± 5.07	3.96 ± 3.33	61.52	0.82	8.30 ± 4.94
97	3.17 ± 2.96	70.07	**1.30**	7.46 ± 5.14	3.82 ± 2.98	62.38	0.82	8.15 ± 4.41
100	**3.14 ± 2.92**	**70.22**	1.35	**7.38 ± 5.05**	**3.79 ± 2.98**	**62.81**	**0.82**	**8.05 ± 4.37**

Training the AE with Portions of the Data

As an additional ablation study, we explore how the performance of the landmark detectors are affected when the self-supervised stage is trained with different portions of the data. The quantitative results for AFLW2000-3D and ParFace are reported in Table 4.5. To that end, we trained the autoencoder (AE) with multiple combinations of the datasets from Table 4.1, where the total amounts to ∼590K images. Note that only the models with 3% and 100% included palsy data, and that all the models were trained with in domain inversion, the AWing loss and encoder fine-tuning.

As part of the experiments, we trained the AE only with palsy datasets at our disposal: Toronto NeuroFace [164], MEEI [224] and the unlabeled set of ParFace. The results correspond to 3% of the total data in Table 4.5. From this experiment, we observed a comparable performance on ParFace with the model trained with 1%, with a minimal improvement on the model trained with palsy data. However, on 300W (2D face alignment) and AFLW2000-3D, the model trained with palsy data showed in general a slightly lower performance than the model trained with only 1% of the data. The main reason is that the Toronto NeuroFace and MEEI are clinical datasets collected in controlled conditions, with little diversity in terms of pose, lighting and background setting. Therefore, a model trained with relatively few images in the wild (in this case from ParFace), would not perform well on challenging images with large poses, occlusion and varying lighting, such as in 300W and AFLW2000-3D, due to insufficient data to generate a compact face representation embedded in the latent code.

Overall, the alignment performance shows a gradual improvement as more data is added to the self-supervised stage. These results lead to the assumption that the performance of the landmark detectors can be further enhanced as more unlabeled data with large diversity is used for training the AE.

Training with Labeled Palsy Data

The results in Section 4.3.9 were computed from models that were not trained using labeled

Table 4.6: Performance metrics on ParFace after fine-tuning with training sets from ParFace. The number of sequences used for fine-tuning are added as numerals adjacent to the loss.

Num.	MSE				AWing			
Seq.	NME$^{\downarrow}$	AUC$^{\uparrow}_{10}$	FR$^{\downarrow}_{10}$	GTE$^{\downarrow}$	NME$^{\downarrow}$	AUC$^{\uparrow}_{10}$	FR$^{\downarrow}_{10}$	GTE$^{\downarrow}$
0	3.99	61.10	1.11	8.79	3.92	61.45	0.89	8.72
1	4.41	56.81	1.11	9.70	4.17	58.70	0.89	8.94
2	4.13	58.63	0.44	8.98	4.08	59.06	0.44	8.50
3	3.60	63.70	0.44	8.34	3.77	62.11	0.44	8.44
4	3.45	65.22	0.44	8.03	3.55	64.12	**0.22**	7.87
5	3.43	65.39	0.44	7.93	3.53	64.47	0.44	8.10
6	3.22	67.43	0.44	7.99	**3.18**	**67.77**	**0.22**	**7.44**

data from ParFace. To evaluate the influence of labeled palsy data in our approach, we split the dataset into a training and test set and fine-tune the previously trained models with a portion of the data. The results are shown in Table 4.6. We use 6 sequences for training and 3 for testing. We split the training set into 6 parts, each containing N number of sequences, where N is in the range [1,6]. The number of sequences used are added as numerals in the Table. MSE-0 and Awing-0 use the models trained without palsy data, to evaluate the test set of ParFace (450 images in total).

The results show an overall improvement when more data is added to fine-tune the models. By adding $6 * 150 = 900$ training images with palsy data to a model trained with more than 61K labeled images (representing less than 2%), we obtained performance gains of around 20% in the NME and more than 10% in the GTE.

These experiments validate the use of ParFace to fine-tune semi- or fully supervised DNN for 3DA-2D and 3D face alignment on asymmetrical facial expressions.

4.3.12 Results on 2D Face Alignment

We report the performance on 2D alignment on the test sets of 300-W and WFLW in Table 4.7. We trained the 2D landmark detectors with the respective training sets of each dataset, using the same architecture as in 3DA-2D face alignment. In this sense, the 3D landmark detector was omitted. Note that our goal is 3D face alignment and we do not aim to improve the current SOTA on 2D face alignment, but rather to achieve a similar performance to the 2D alignment architecture 3FabRec. Also note that 3FabRec used VGGFace2 [253] and AffectNet during their reconstruction stage, with $\sim$2.1M training images. Since VGGFace2 was withdrawn, we used smaller datasets, which account for $\sim$590K images for training and 70K in the validation set.

Following related work, we use the normalized mean error (NME) for both datasets and the failure rate (FR) and area under the curve (AUC) at 10% of the Cumulative Error Distribution on

Table 4.7: Comparison with 3FabRec, the baseline architecture for 2D face alignment. Note that ParFace-Net was trained with $\sim 31\%$ of the data used for 3FabRec. In addition, our pipeline is targeted to 3D face alignment on symmetric and asymmetric facial expressions.

Method	300W - NME$^{\downarrow}$			WFLW		
	Common	Challenging	Full	NME$^{\downarrow}$	AUC$_{10}^{\uparrow}$	FR$_{10}^{\downarrow}$
3FabRec [171]	**3.36**	**5.74**	**3.82**	5.62	48.4	8.28
ParFace-Net$_{MSE}$	3.44	<u>5.83</u>	3.91	<u>5.33</u>	<u>49.76</u>	<u>7.68</u>
ParFace-Net$_{Awing}$	<u>3.38</u>	5.85	<u>3.86</u>	**5.30**	**49.78**	**7.56**

Table 4.8: Ablation study on 2D face alignment. The first two best results per category are highlighted.

Self-Superv.	IDI	Loss	EFT	300W - NME $\pm \sigma^{\downarrow}$			WFLW		
				Common	Challenging	Full	NME $\pm \sigma^{\downarrow}$	AUC$_{10}^{\uparrow}$	FR$_{10}^{\downarrow}$
-	-	MSE	-	10.15 ± 5.96	20.58 ± 10.83	12.19 ± 7.18	17.21 ± 11.28	8.42	67.12
-	-	MSE	✓	4.90 ± 3.31	10.80 ± 6.83	6.06 ± 4.24	7.68 ± 5.92	34.84	19.68
✓	-	MSE	-	3.90 ± 3.10	7.50 ± 6.07	4.60 ± 3.86	7.37 ± 6.04	40.48	17.88
✓	✓	MSE	-	3.70 ± 2.84	6.61 ± 4.74	4.27 ± 3.30	6.60 ± 5.65	44.28	14.32
✓	-	MSE	✓	3.58 ± 2.75	6.02 ± 4.31	4.05 ± 3.12	5.52 ± 4.47	48.03	8.48
✓	✓	MSE	✓	<u>3.44</u> $\pm$ <u>2.65</u>	**5.83** $\pm$ <u>4.16</u>	<u>3.91</u> $\pm$ <u>3.00</u>	<u>5.33</u> $\pm$ <u>4.40</u>	<u>49.76</u>	<u>7.68</u>
-	-	AWing	-	11.38 ± 6.73	21.32 ± 11.28	13.33 ± 7.83	17.40 ± 11.25	6.15	71.12
-	-	AWing	✓	4.96 ± 3.33	10.68 ± 6.66	6.08 ± 4.20	7.61 ± 5.75	35.24	18.88
✓	-	AWing	-	4.30 ± 3.62	8.73 ± 7.98	5.17 ± 4.80	7.87 ± 6.47	37.83	20.84
✓	✓	AWing	-	3.99 ± 3.04	7.25 ± 5.66	4.63 ± 3.70	7.12 ± 6.11	41.35	16.56
✓	-	AWing	✓	3.62 ± 2.79	6.13 ± 4.27	4.11 ± 3.13	5.44 ± 4.41	48.63	8.36
✓	✓	AWing	✓	**3.38** $\pm$ **2.59**	<u>5.85</u> $\pm$ **4.12**	**3.86** $\pm$ **2.95**	**5.30** $\pm$ **4.35**	**49.78**	**7.56**

WFLW. The metrics are normalized with the inter-ocular (IO) normalization.

Our results show that ParFace-Net with in-domain inversion surpasses 3FabRec on WFLW. For 300W, the performance of ParFace-Net is comparable in the Common and Full set, but slightly worse in the Challenging set. We hypothesize this might be caused due to the relatively small amount of challenging images with large head poses, strong illumination changes and large occlusion, used to train the AE.

Ablation Study

Similarly to 3DA-2D and 3D face alignment, we analyzed the impact of the different modules in the detection of 2D facial landmarks. The results for 300W and WFLW are reported in Table 4.8.

Training the AE with Portions of the Data. The influence of the amount of data used for training the self-supervised stage in the alignment of 2D landmarks is investigated as well. The

Table 4.9: Ablation study on 2D face alignment after training the self-supervised stage with portions of the data. The AE in * was trained only with palsy data.

%	300W - NME $\pm \sigma^{\downarrow}$		
	Common	Challenging	Full
0	4.96 ± 3.33	10.68 ± 6.66	6.08 ± 4.20
1	4.11 ± 2.97	8.15 ± 5.60	4.90 ± 3.64
3*	4.18 ± 2.97	8.35 ± 5.80	4.99 ± 3.70
13	3.78 ± 2.78	6.85 ± 4.88	4.38 ± 3.30
34	3.56 ± 2.71	6.18 ± 4.31	4.07 ± 3.09
97	3.42 ± 2.69	$\mathbf{5.83} \pm 4.18$	3.89 ± 3.04
100	$\mathbf{3.38} \pm \mathbf{2.59}$	$\underline{5.85} \pm \mathbf{4.12}$	$\mathbf{3.86} \pm \mathbf{2.95}$

results on 300W are shown in Table 4.9. The landmark detectors were trained with the Awing loss and encoder fine-tuning.

Equivalently to 3DA-2D and 3D alignment, the amount and variety of the training data affect the performance of the landmark detector. Therefore, we hypothesize that more unlabeled training data in the self-supervised stage would help close the gap for 2D face alignment on 300W w.r.t 3FabRec.

Training the Self-Supervised Stage. For this experiment, the 2D landmark detector is trained before and after the AE has been trained in the self-supervised stage. In the first case, only the encoder is pre-trained on ImageNet. The results are shown in Table 4.8. We additionally compare the performance of adding encoder fine-tuning after the landmark detector training.

The results show that without the self-supervised stage and encoder fine-tuning, the landmarks detector fails to exploit the latent code and predict the landmark correctly. After fine-tuning the encoder, the latent code is adjusted and the results dramatically improve.

Training with In-Domain Inversion. We examined the impact of the in-domain inversion module and report the results in Table 4.8. For every metric, we observed an improved performance when using this module, similarly to 3DA-2D and 3D face alignment. This can be explained as the latent code integrates an enhanced representation of the facial attributes, after the encoder has been fine-tuned with in-domain inversion.

Effect of AWing Loss. Finally, we assessed the performance of the AWing loss in contrast to the MSE for 2D landmark detection. The results are shown in Table 4.8. We observed that the AWing loss had a positive impact on face alignment, after being combined along with in-domain inversion and encoder fine-tuning. In other cases, the MSE loss showed a better performance. This could be attributed to the fact that MSE converges faster than AWing for the same number of epochs. After the encoder is fine-tuned, the improvement on the AWing loss in more noticeable than for the MSE.

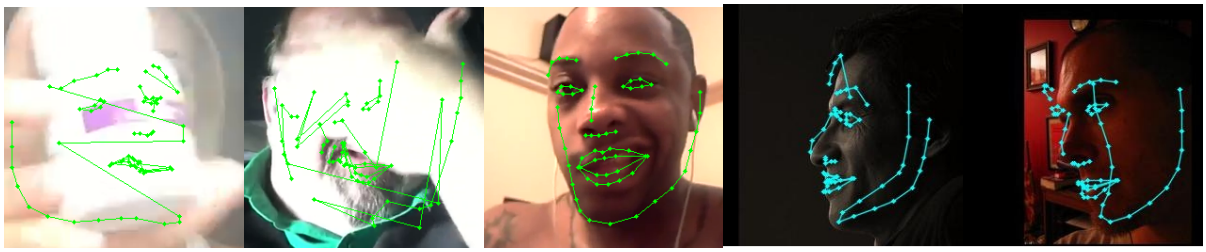


Figure 4.10: Failure cases.

4.3.13 Discussion and Limitations

As most of the heatmap-based methods for face alignment, our approach fails under extreme occlusions, as shown in Figure 4.10. Other failure cases occur when the face synthesis fails due to unusual facial expressions, large head poses, lighting and low contrast. 3DMMs-based methods are more robust in such cases, by keeping the spatial structure of the landmarks, even if the face is not properly aligned. However, they are not able to align the landmarks correctly for unseen faces, such as in asymmetrical expressions (see DECA in Figure 4.7).

We noticed that our method heavily depends on the training data in the self-supervised stage. By using datasets with less diversity in terms of pose, expression, occlusion and illumination, the performance of the landmark detectors drops. The dependency on unlabeled data is not to be seen as a drawback, since the generation of such datasets is much less expensive than labeled data. We also observed that using a small set of unlabeled palsy faces ($\sim$3% of the total amount, see Table 4.1) to train the AE enabled the in-domain inversion module to encode asymmetrical features in the latent vector, improving the landmark detection.

Dedicated architectures for HQ face reconstruction such as StyleGAN2 [208] could replace the inverted ResNet-18, as in [207], to improve the reconstruction. This comes at a cost of increased complexity and trainable parameters in the pipeline. While StyleGAN2 has $\sim$28M parameters and a computational complexity of 143.15 Giga Multiply-Accumulate Operations (GMACs), ResNet-18 has $\sim$11M parameters and a complexity of 1.82GMAC.

4.4 Facial Landmark Tracking

The detection of facial landmarks in 2D images has received a great attention in the last decade, as it is a fundamental step for several applications in computer vision. Most of the approaches focus on still images, and are extended to videos by using a tracking-by-detection scheme.

In this Section, we introduce a novel tracking pipeline based on grouped-landmark Kalman Filters, that can be integrated into existing deep-learning-based 3D face alignment approaches. The proposed method improves the landmark accuracy in cases with large occlusion, extreme head poses and blurriness that affect existing approaches. Our experiments on the Menpo 3DA-2D and ParFace benchmark show improvements on model-free and 3D-model-based face alignment approaches.

4.4.1 Overview of the Proposed Landmark Tracking Pipeline

Our hybrid 3D landmark tracking pipeline combines the landmarks retrieved from a DNN-based face alignment method with a Kalman Filter. Based on a motion model, the filter predicts the landmark positions, which are later corrected by the measurements from the DNN. Additionally, our pipeline leverages uncertainty information from heatmap or random-variable-based face alignment, by adapting the Kalman Filter measurement noise based on the confidence of the face alignment detection.

4.4.2 Noise Adaptive Clustered Landmark Kalman Filters

We perform frame-to-frame tracking of facial landmarks using linear Kalman Filters [45]. As described in Chapter 2.2, Kalman Filter represents the belief for the state $\mathbf{x}_k$ at time-step k by its mean $\hat{\mathbf{x}}_k$ and covariance $\boldsymbol{P}_k$. In the prediction or time-update step of a Kalman Filter between frames $k-1$ and k, the state $\hat{\mathbf{x}}_k$ is updated based on an underlying motion model, described by a transition matrix $\mathbf{A}$ as

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} \tag{4.6}$$

and the covariance is computed following Eq. (2.25), where $\mathbf{Q}_k$ denotes the process noise covariance. Subsequently, in the Kalman Filter correction step, the measurement $\mathbf{z}_k$ for frame k is incorporated. In our case, this measurement is given by the landmark location predictions from the DNN. First, the Kalman gain $\mathbf{K}_k$ and the innovation $\mathbf{y}_k$ are computed using Eq. (2.27) and (4.7), where $\mathbf{H}$ is the observation or correction function and $\mathbf{R}_k$ the measurement covariance.

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^- \tag{4.7}$$

Finally, the filter state and covariance are updated following Eq. (2.28) and (2.29), respectively. Eq. (2.28) can be rewritten as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{y}_k. \tag{4.8}$$

Existing works on landmark tracking with Kalman Filters consider each landmark separately, by introducing one Kalman Filter per landmark [237, 238]. However, taking into account the strong correlations in the motion between face landmarks due to the connectivity of face parts, we propose to use clustered Kalman Filters that contain information of more than one landmark in their state. We propose two landmark grouping options shown in Figure 4.11.

For our Kalman Filters, we implemented two motion models: constant velocity and constant acceleration. For the constant velocity model, the Kalman Filter state is given by

$$\mathbf{x}_{\text{cv}} = \left[\ell_1^\top\ \ell_2^\top\ \cdots\ \ell_M^\top\ \dot{\ell}^\top\right]^\top, \in \mathbb{R}^{(3M+3)\times(1)} \tag{4.9}$$

where $\ell_i = [\ell_{i,x}, \ell_{i,y}, \ell_{i,z}]^\top$ is a 3D landmark position, M is the number of landmarks in the Kalman Filter and $\dot{\ell} = \left[\dot{\ell}_x, \dot{\ell}_y, \dot{\ell}_z\right]^\top \in \mathbb{R}^3$ is a common landmark velocity vector for all landmarks.

The transition matrix is then defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3M} & \mathbf{T} \\ \mathbf{0}_{3\times 3M} & \mathbf{I}_3 \end{bmatrix}, \in \mathbb{R}^{(3M+3)\times(3M+3)} \tag{4.10}$$

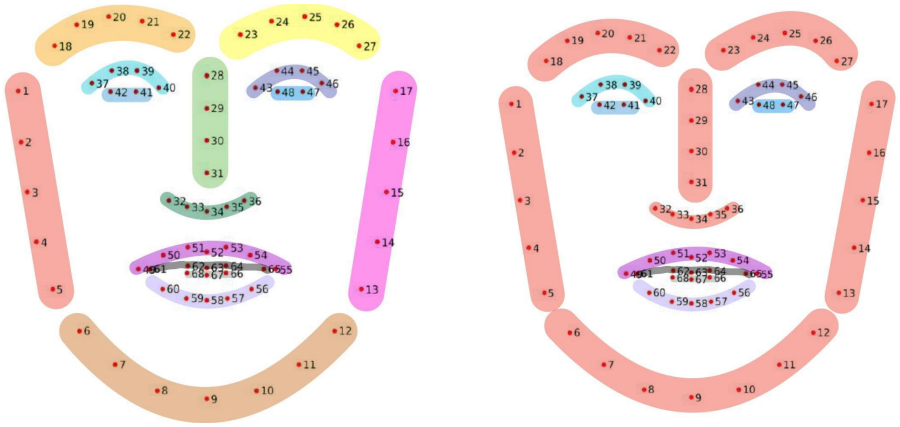


Figure 4.11: Landmark grouping in Kalman Filters. The right group (G2) separates highly expressive regions (eyes, mouth) from the rest of the face, while the left group (G1) is even more granular.

with $\mathbf{T} = [\mathbf{I}_3\Delta t \ \cdots \ \mathbf{I}_3\Delta t]^\top, \in \mathbb{R}^{3M\times 3}$, and Δt is the time between two consecutive frames, calculated from the respective dataset frame rate.

The measurement vector $\mathbf{z}_k^\top = \begin{bmatrix} \overline{\ell}_1^{k\top} & \cdots & \overline{\ell}_M^{k\top} \end{bmatrix}, \in \mathbb{R}^{3M\times 1}$ contains the landmark estimates $\overline{\ell}_i$ produced by the face alignment network. Thereby, the correction function is given by

$$\mathbf{H} = [\mathbf{I}_{3M} \quad \mathbf{0}_{3M\times 3}], \in \mathbb{R}^{(3M)\times(3M+3)}. \tag{4.11}$$

For the constant acceleration model, the Kalman Filter definition follows the same concept, with the addition of a common acceleration vector for all landmarks, making the state vector

$$\mathbf{x}_{\text{ca}} = \begin{bmatrix} \ell_1^\top & \ell_2^\top & \cdots & \ell_M^\top & \dot{\ell}^\top & \ddot{\ell}^\top \end{bmatrix}^\top, \in \mathbb{R}^{(3M+6)\times(1)}. \tag{4.12}$$

The transition and observation matrices $\mathbf{A}_{\text{ca}}$ and $\mathbf{H}_{\text{ca}}$ for the constant acceleration model are given by

$$\mathbf{A}_{\text{ca}} = \begin{bmatrix} \mathbf{I}_{3M} & \mathbf{T} & \mathbf{T_2} \\ \mathbf{0}_{3\times 3M} & \mathbf{I}_3 & \mathbf{I}_3\Delta t \\ \mathbf{0}_{3\times 3M} & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{4.13}$$

and

$$\mathbf{H}_{\text{ca}} = [\mathbf{I}_{3M} \quad \mathbf{0}_{3M\times 6}], \in \mathbb{R}^{(3M)\times(3M+6)}, \tag{4.14}$$

where $\mathbf{T_2} = \left[\mathbf{I}_3 0.5\Delta t^2 \ \cdots \ \mathbf{I}_3 0.5\Delta t^2\right]^\top, \in \mathbb{R}^{3M\times 6}$.

4.4.3 Confidence Adaptive Kalman Filter

Some face alignment networks such as FAN provide additional information on their confidence for each landmark detection. We incorporate this information into the Kalman Filter and make its measurement noise covariance dynamically-adaptive based on this confidence. For a landmark measurement $\overline{\ell}_i$, the noise covariance is computed as:

$$\sigma^2_{\ell_i} = \begin{bmatrix} \sigma^2_{\ell_{i,x}} \\ \sigma^2_{\ell_{i,y}} \\ \sigma^2_{\ell_{i,z}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\bar{\ell}_{i,x}(1-c_{\bar{\ell}_i})}{|\bar{\ell}_i|} \right)^2 \\ \left(\frac{\bar{\ell}_{i,y}(1-c_{\bar{\ell}_i})}{|\bar{\ell}_i|} \right)^2 \\ \left(\frac{\bar{\ell}_{i,z}(1-c_{\bar{\ell}_i})}{|\bar{\ell}_i|} \right)^2 \end{bmatrix} \tag{4.15}$$

where $c_{\bar{\ell}_i}$ is the confidence value between $[0, 1]$ for landmark i from the face alignment network. Using Eq. (4.15), the measurement covariance matrix of our Kalman Filter is updated for every frame k as

$$\mathbf{R}_k = diag\left(\begin{bmatrix} \sigma^{2\top}_{\ell_1} & \sigma^{2\top}_{\ell_2} & \cdots & \sigma^{2\top}_{\ell_M} \end{bmatrix} \right). \tag{4.16}$$

4.4.4 Activation of Face Detector

Face alignment is performed on images cropped around the face. To that end, a face detector is employed to retrieve the bounding box on every frame in tracking-by-detection approaches. Other methods propose to use the landmarks of the previous frame to estimate the bounding box of the current frame, under the assumption that the motion between frames is not large.

In this work, we introduce a novel mechanism for reducing the times a face detector is required in video sequences, under the same assumption. This mechanism follows the behavior of the Kalman Filter state covariance $\boldsymbol{P}_k$ in consecutive frames as a cue to activate the face detector. When this covariance consistently increases over μ number of frames, as a result of a decrease on the confidence score of the landmarks, it is an indicator of divergence, and hence the face detector is employed in the current frame. Otherwise, the bounding box (BB) is continuously computed from the landmarks detected in the previous frame. That is,

$$\text{BB} = \begin{cases} \text{From face detector} & \text{if } \boldsymbol{P}_n > \boldsymbol{P}_{n-1} \ \ \forall\, n \ \in (k-\mu, k] \\ \text{From landmarks} & \text{otherwise.} \end{cases} \tag{4.17}$$

4.4.5 Datasets

Our method for landmark tracking was evaluated on the 3DA-2D Menpo tracking benchmark [83] and ParFace. Menpo is currently the most used video sequence-based dataset for 3D facial landmark tracking, while ParFace is targeted to track facial landmarks on patients with unilateral facial paralysis.

In Menpo, the training and test sets consist of 90 and 35 videos, respectively, of 1K frames each, with annotations for 84 landmarks. ParFace, on the other hand, is composed of 9 annotated sequences of 150 frames each, with 68 landmarks. While Menpo is a large-scale dataset with sequences under varying conditions and large head rotations, ParFace has challenging asymmetrical facial expressions and total occlusion in some frames.

In our work, ParFace and the training set of Menpo were used for parameter fine-tuning and evaluation of different Kalman Filter architectures, while the test set of Menpo was used in the

face detection activation mechanism. ParFace and the test set of Menpo were used to evaluate the performance of the Kalman Filter with different face alignment methods as well.

4.4.6 Evaluation Metrics

Following [83], the performance is evaluated using the normalized point-to-point root mean square error (RMSE) in Eq. (2.41), Cumulative Error Distribution (CED) curve, failure rate (FR) based on an error threshold of 5% and area under the curve (AUC) up to an error of 0.08. In ParFace, we use the ground truth error (GTE) to evaluate the full 3D coordinates. The RMSE is normalized by the face diagonal of the bounding box that tightly encloses the ground truth landmarks in each frame, while the GTE is normalized by the inter-ocular distance, *i.e.*, the outer corners of the eyes. These metrics are defined in Section 2.5.

4.4.7 Ablation Study

We investigated different Kalman Filter configurations, in addition to the performance of the proposed face detector activation mechanism. In the experiments where a face detector is needed, we used FaceBoxes [71] to crop the face. FaceBoxes is described in Section 2.4.1.

Kalman Filter design

We first evaluated the performance of several Kalman Filter configurations, as described in Section 4.4.2. In this experiment, we used FAN [73] for landmark detection and provided the ground truth landmarks to crop the face. The results are shown in Table 4.10. Note that FAN provides a confidence score for every landmark, that is exploited by our Kalman Filter with adaptive covariance.

Table 4.10: Performance of different Kalman Filter configurations on the training set of Menpo 3DA-2D benchmark and ParFace.

Method	RMSE$^{\downarrow}$	
	Menpo	ParFace
FAN [73]	0.02193	0.04343
FAN + Kalman Filter: Constant acceleration (CA)	0.0218	0.03908
FAN + Kalman Filter: Constant velocity (CV)	0.02172	0.03359
FAN + Kalman Filter: G1 + CV	0.02154	0.03464
FAN + Kalman Filter: G1 + CV + adapt. covariance	<u>0.02149</u>	**0.03271**
FAN + Kalman Filter: G2 + CV + adapt. covariance	**0.02148**	<u>0.03285</u>

Since the Kalman Filter with constant velocity (CV) has a better performance than the constant acceleration model in both datasets, we used CV in the rest of the experiments. From these results, we observed that the Kalman Filter has a positive effect on the face tracking, particularly when using landmark grouping and after integrating the face alignment confidence as

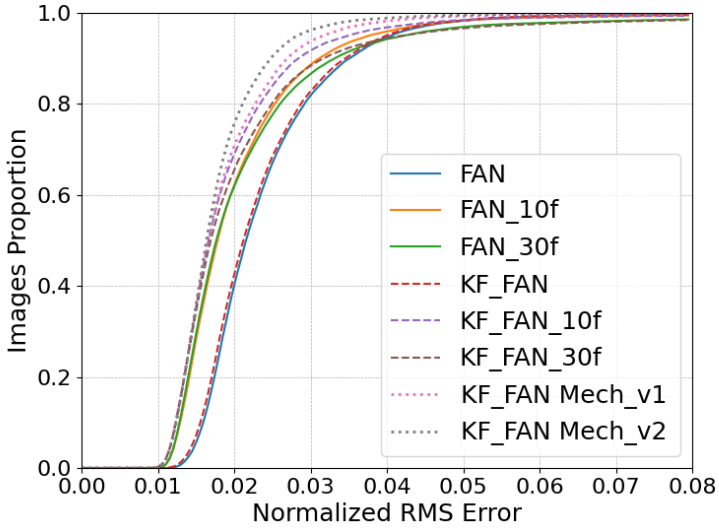


Figure 4.12: CED curve for FAN with and without Kalman Filter, with face detector or activation mechanism. v1 corresponds to Activation Mechanism of Kalman Filter with ratio 1/3 and v2 to Kalman Filter with ratio 1/5.

adaptive covariance. The improvements are more evident in ParFace, which includes sequences with partial and total occlusion, fast motion and large head rotations.

In the following, we selected the best performing configuration on the Menpo benchmark with grouping G2. For face alignment methods without confidence scores, we excluded the adaptive covariance.

Activation of face detector

We additionally tested the proposed face detector activation mechanism, detailed in Section 4.4.4. We evaluated it in the Menpo benchmark and used FAN as the face alignment method. We compared the mechanism to using face detection in every frame, every 10 frames (10f) and every 30 frames (30f), and with and without Kalman Filter. In the cases without the mechanism and where the face detector is used periodically, we use the landmarks from the previous frame to compute the bounding box that crop the faces. The results on the 3DA2D Menpo test set are shown in Table 4.11 and the CED curve in Figure 4.12.

We investigated using $\mu = 3$, and varied the ratio of Kalman Filters for which the covariance constantly increases, between 1/3 and 1/5 of all Kalman Filters. We observed a significant improved performance in every metric when using the activation mechanism, in contrast to using the face detector in every frame or periodically every 10 or 30 frames, with and without Kalman Filter. We also observed a lower RMSE and a higher AUC when using the Kalman Filter for every corresponding pair with the face detector, in contrast to without Kalman Filter. Note that with the activation mechanism with a Kalman Filter ratio of 1/3, the face detector was used in average 18.67% of the total number of frames, while with 1/5, 52.4%. These results demonstrate the robustness of the mechanism to enhance the Kalman Filter results and reduce the usage of the face detector, particularly for implementations with limited resources.

Table 4.11: Performance of face detection activation mechanism vs using the face detector with FAN on Menpo 3DA-2D benchmark. Activation Mechanism v1 corresponds to Kalman Filter with ratio 1/3 and v2 to Kalman Filter with ratio 1/5.

Method	RMSE$^{\downarrow}$	AUC$^{\uparrow}$	FR$_5^{\downarrow}$(%)
Face detector	0.0245	0.696	1.72
Face detector $\times$10f	0.0216	0.737	1.67
Face detector $\times$30f	0.0294	0.727	3.07
Kalman Filter + Face detector	0.0240	0.701	1.63
Kalman Filter + Face detector $\times$10f	0.0205	0.753	1.57
Kalman Filter + Face detector $\times$30f	0.0238	0.735	3.38
Kalman Filter + Activation Mech. v1	<u>0.0189</u>	<u>0.761</u>	<u>0.73</u>
Kalman Filter + Activation Mech. v2	**0.0181**	**0.771**	**0.46**

4.4.8 Results on 3D Face Tracking

Face alignment network

The influence of the proposed tracking approach was evaluated on four different DNN-based alignment methods: (a) a heatmap-based method, FAN [73]; (b) a volumetric approach, JVCR [74]; (c) a 3DMM-based pipeline, 3DDFA [169]; and (d) a hybrid approach, based on 3DMM and landmark regression, SynergyNet [201]. Note that these approaches were trained on a smaller dataset, 300W-LP [13], which consists of $\sim$61K static images with annotations for 68 landmarks. As it does not contain videos, these approaches were not trained for face tracking.

We evaluated on the test set of Menpo and in ParFace. Note that mapping from the 84 landmarks from Menpo to 68 and back is a straightforward task, since the extra landmarks correspond to the midpoint in each pair of consecutive landmarks in the jaw. The results of this experiment in Menpo are shown in Figure 4.13 and Table 4.12. For ParFace, the results are shown in Figure 4.14 and Table 4.13.

In Table 4.12, we observe that there is a consistent improvement of the RMSE score when our Kalman Filter is added. The same holds for the AUC metric and the Failure Rate (FR), except in SynergyNet. This could be a result of the sensitivity of SynergyNet to the choice of face detector.

In Figure 4.13 we observed that 3DDFA-V2, a 3DMM-based method, is more robust to video sequences than model-free methods such as FAN and JVCR. Nonetheless, the improvement gained by our KF method is more noticeable when we combine it with FAN, where the confidence score is provided.

Menpo tracking benchmark

We additionally compared our results with the ones from the Menpo benchmark, which comprises methods specifically designed for the tracking of facial landmarks. The results are shown in Table 4.14 and Figure 4.15.

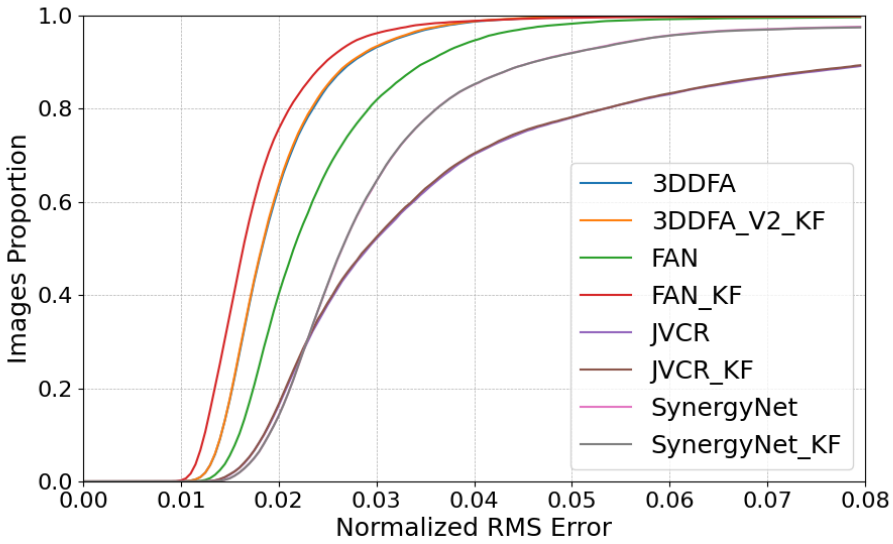


Figure 4.13: CED curve for face alignment alignment methods with and without Kalman Filter on the Menpo 3DA2D test set.

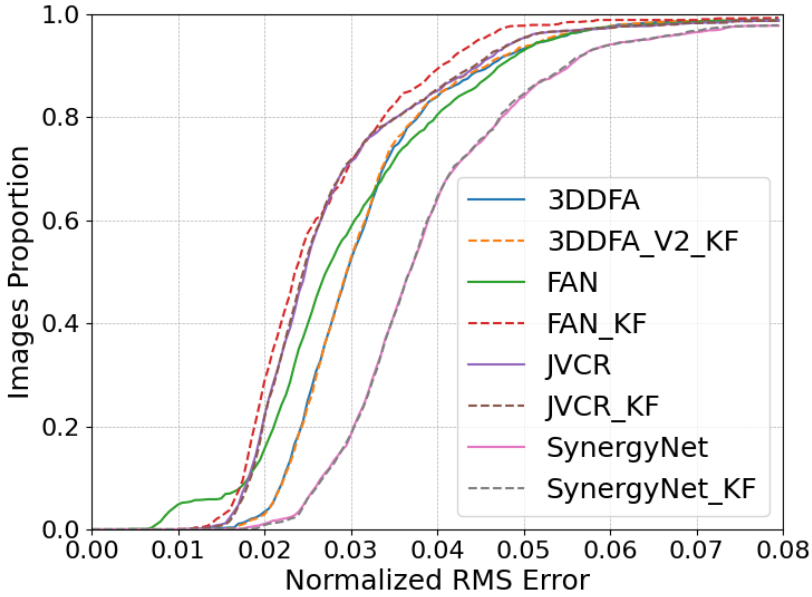


Figure 4.14: CED curve for face alignment alignment methods with and without Kalman Filter on ParFace.

Table 4.12: RMSE, Area under the curve and failure rate (%) on Menpo 3DA2D tracking test set.

Method	Without Kalman Filter			With Kalman Filter		
	RMSE$^{\downarrow}$	AUC$^{\uparrow}$	FR$_5^{\downarrow}$	RMSE$^{\downarrow}$	AUC$^{\uparrow}$	FR$_5^{\downarrow}$
FAN [73]	0.0245	0.696	1.72	0.0181	0.771	0.46
JVCR [74]	0.0416	0.538	21.7	0.0410	0.540	21.5
3DDFA-V2 [169]	0.0199	0.746	0.21	0.0198	0.747	0.19
SynergyNet [201]	0.0342	0.620	7.89	0.0340	0.620	7.93

Table 4.13: RMSE, Area under the curve, failure rate (%) and GTE on ParFace.

Method	Without Kalman Filter				With Kalman Filter			
	RMSE$^{\downarrow}$	AUC$^{\uparrow}$	FR$_5^{\downarrow}$	GTE$^{\downarrow}$	RMSE$^{\downarrow}$	AUC$^{\uparrow}$	FR$_5^{\downarrow}$	GTE$^{\downarrow}$
FAN [73]	0.0439	0.619	7.04	0.103	0.0275	0.662	2.22	0.083
JVCR [74]	0.0418	0.643	4.59	0.122	0.0314	0.644	4.52	0.100
3DDFA-V2 [169]	0.0454	0.597	6.74	0.141	0.0343	0.598	6.44	0.121
SynergyNet [201]	0.0533	0.508	16.1	0.165	0.0437	0.508	15.5	0.146

4.4.9 Discussion

In this Section, we evaluated multiple configurations of the proposed tracking method with Kalman Filters along with FAN. Our results showed that grouping the landmarks and adding the adaptive covariance improved the performance on the Menpo training set and in ParFace. The integration of the activation mechanism to the previous setup visibly contributed to decrease the RMSE, FR and the need of the face detector, while boosting the AUC.

In the Menpo benchmark, the proposed method boosted the performance of FAN to have competitive results in comparison to the SOTA. as shown in Table 4.14.

Table 4.14: Area under the curve and Failure Rate on Menpo 3DA2D tracking benchmark. *Taken from [83].

Method	RMSE$^{\downarrow}$	AUC$^{\uparrow}$	FR$_5^{\downarrow}$
Deng et al. [83]*	-	**0.7977**	1.68
Xiong et al. [233]	0.0694	<u>0.7935</u>	3.38
Zadeh et al. [205]	0.0515	0.7187	1.83
Crispell and Bazik [76]	0.0193	0.7617	1.61
FAN [73]	0.0245	0.6960	1.72
Kalman Filter + FAN + AM v1	<u>0.0188</u>	0.7612	<u>0.73</u>
Kalman Filter + FAN + AM v2	**0.0180**	0.7707	**0.46**

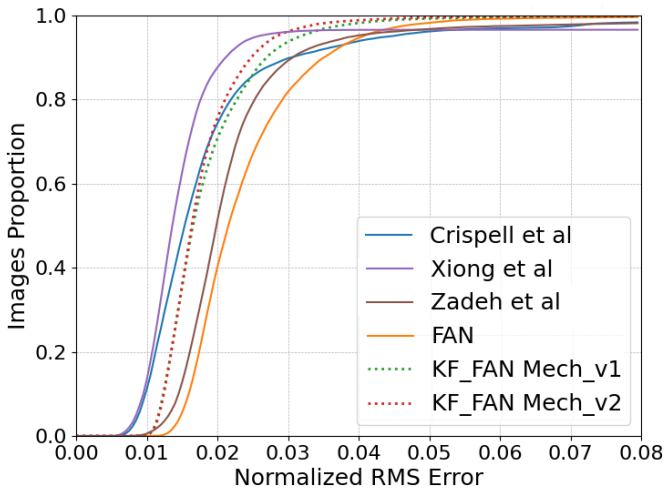


Figure 4.15: CED curve on Menpo 3DA2D tracking benchmark.

Moreover, our tracking approach proved beneficial to different types of face alignment networks. Methods based on 3DMM such as 3DDFA-V2 are more robust in video sequences without many challenges as in Menpo (see Table 4.12), by keeping the spatial structure of the face. Hence, the improvement might not be evident in such cases. However, in challenging sequences with large head rotations, occlusion and varying illumination, as in ParFace, the enhancement is more noticeable. This improvement was observed in both 3DA2D and 3D landmarks in ParFace. In the Menpo test set, FAN with the Kalman Filter stands out as the best performing method, where the uncertainty of the landmarks is exploited by our tracking approach.

4.5 Conclusions

This Chapter introduced a novel pipeline for 3D face alignment, targeted to faces with symmetrical and asymmetrical expressions. We propose a semi-supervised architecture which exploits large unlabeled datasets and integrates face alignment with smaller labeled datasets. We explore the latent space in the self-supervised stage, and optimize the encoder to produce a disentangled latent space with in-domain inversion. Our landmark detector uses the AWing loss to regress 3DA-2D landmark heatmaps and a newly introduced separate branch computes the depth of the 3D landmarks. A future direction would be to exploit additional 3DMMs parameters, to enable the autoencoder to learn the pose, expression, and shape from 2D images under large head poses and extreme occlusion.

Furthermore, we propose a novel mechanism to integrate tracking to single-frame-based face alignment approaches. Our method is based on Kalman Filter and exploits the uncertainty from the detected facial landmarks and their connectivity. Our results show how this mechanism benefits the alignment in videos, particularly in cases with large poses and occlusion. Our experiments were evaluated on sparse face alignment approaches, but our pipeline can be extended to dense face alignment methods with uncertainty as in [87], where landmarks can be grouped semantically.

5 Performance-Driven Facial Animation

Contents

5.1 Introduction 106
5.2 Related work 107
5.3 Overview of the Proposed Pipeline 108
5.4 Target Energy Functional 110
5.5 Experiments and Results 112
5.6 Conclusions 117

Publication

This Chapter is based on the work presented in:

- Jilliam M. Díaz Barros, V. Golyanik, K. Varanasi and D. Stricker, "Face It!: A Pipeline for Real-Time Performance-Driven Facial Animation", in IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 2209-2213. 2019.

5.1 Introduction

Real-time performance-driven facial animation refers to the problem of capturing a live video stream of a person and animating a virtual avatar upon the observed facial expressions. Although this problem was first investigated in the context of virtual avatar generation for films and computer games, such a system can also help in developing affective user interfaces in real world contexts. For example, facial movements of users can be used to assess their psychological state, intent in reaching for a specific tool or response to an interactive computer system. Some applications of such interfaces could be: driver monitoring in automobiles, service kiosks for patients in hospitals or installations in theme parks.

Depending on the target application, there is always a trade-off between the complexity of the acquisition setup and the quality of the animation output [254]. On one side, there are high-end systems used in the movie and gaming industries, such as active 3D scanners or marker-based motion capture systems. Even though they provide realistic animations, they are intrusive and require substantial manual intervention. On the other side, there are simple, inexpensive and non-intrusive passive-scanning devices such as conventional monocular RGB cameras. Even though RGB or intensity-based facial-tracking methods have limited operational performance (*e.g.*, under varying illumination), monocular cameras are ubiquitous and flexible in installation and usage. More recently, several approaches based on commodity RGB-D sensors have been proposed [255, 254, 256]. Nevertheless, the most common visual data acquisition technology in every-day life constitutes RGB cameras as those embedded in mobile devices.

In this work, we aim at real-time 3D facial character animation from monocular RGB or intensity images, which can be used in consumer-centric applications. In order to facilitate real-time interaction, the system has to have low hardware and data requirements, while being robust to a diverse range of human users. Hence, the 2D facial tracking has to be robust, accurate and lightweight. Moreover, the setup should not rely on specialized hardware or markers.

This Chapter introduces a new lightweight approach for real-time performance-driven facial animation from monocular videos. We transfer facial expressions from 2D images to a 3D virtual character by estimating the rigid head pose and non-rigid face deformations from detected and tracked 2D facial landmarks. We map the input face into the facial expression space of the 3D head model using blendshape model and formulate a lightweight energy-based optimization problem which is solved by non-linear least squares.

Our performance-driven animation pipeline runs at 18 frames per second on a single CPU. Additionally, it robustly handles varying head poses and different facial expressions, including moderately asymmetric ones. Compared to related methods, our approach does not require training data, specialized camera setups or graphics cards, and is suitable for embedded systems. We support our claims with several experiments in Section 5.5.

The primary contributions of this Chapter are:

- A new real-time framework for performance-driven facial animation from a monocular setup. We formulate 3D character animation as a lightweight energy-based optimization problem solved with non-linear least-squares (Section 5.4).

- To fulfill real-time constraints, our energy functional relies only on a sparse set of 2D facial landmarks, which are used to update the head pose and facial expressions (Section 5.4).
- A novel differentiable energy term for limiting the range of the blendshape target weights, described in Section. 5.4.3.
- A set of experiments for the validation of different aspects of the proposed method (Section. 5.5).

5.2 Related work

In this Section, we summarize SOTA methods in monocular facial performance capture. For an extensive overview on this topic, including multi-view setups, we refer the reader to [257].

Several works propose approaches for non-rigid tracking and character animation which require either specialized setups, physical markers, RGB-D cameras or manual intervention [258, 254, 255, 259, 260, 261, 262]. In [258], Weise *et al*. propose a real-time approach based on non-rigid registration and tracking using a PCA model for expression retargeting. A structured-light scanner is used to recover high-resolution depth maps. For animating different characters, deformation transfer was required on an additional step. In a later work [254], they used a Microsoft Kinect for character animation using facial tracking. In this sense, RGB-D data is used to recovered a user-specific model by reconstructing offline dynamic 3D expression model (DEM), where manual intervention was required. Facial tracking is formulated as a non-rigid registration problem and an animation prior is used for regularization. In [255], Bouaziz *et al*. presented a real-time face tracking method to animate 3D models from RGB-D data. An online adaptive DEM, along with an identity PCA model and a parameterized deformation model were used to create user-specific corrective shapes, without the need of manual correction or user-specific calibration.

Cao *et al*. [85] introduced a real-time facial animation approach from 2D data. This method requires a user-specific shape regressor trained in a preprocessing step with manual adjustments, where the user is asked to perform predefined facial expressions. In the follow-up [263], they use public image datasets to train the regressor, instead of the initial calibration step. In addition, aligned 2D facial landmarks are used to adjust the camera calibration and the user identity. [264] describes a bilinear face model for identity and facial expression representation, based on 2D or RGB-D data, which can be used to generate a blendshape model of an actor or animate a 3D face.

Garrido *et al*. [265] introduced an offline approach for automatic reconstruction and animation of user-specific 3D face rigs from monocular videos. Their pipeline consists of three layers, where a parametric shape model is defined to encompass the subspace of facial identity, facial expression and fine-scale details such as wrinkles. The three levels are combined using sparse linear regression. In [260], Thies *et al*. proposed a real-time facial expression transfer approach using RGB-D data for both source and target. A joint parametric model is used to fit identity, expression and skin reflectance. They achieved photo-realistic facial reenactment with impressive results; however, a GPU-based Gauss-Newton optimizer was required to achieve real time performance. In a later work [266], they introduced a real-time photo-realistic facial monocular

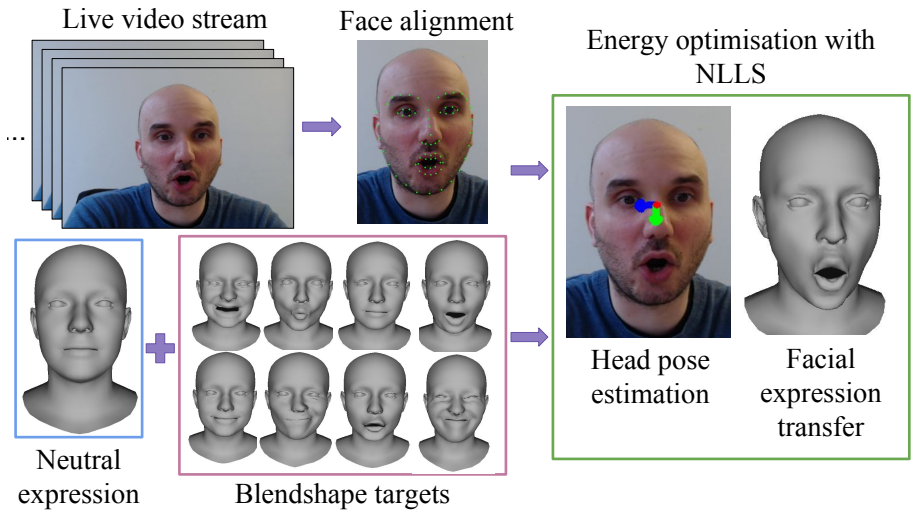


Figure 5.1: Overview of the proposed pipeline for 2D-3D facial expression transfer: facial landmarks are detected in every incoming image and used to find the head pose and an optimal blendshape combination resembling the observed appearance.

reenactment approach. They track facial landmarks relying on a dense photometric consistency measure and use GPU-based iteratively reweighted least squares solver to achieve real-time frame rates.

Blanco *et al*. [261] proposed a marker-based method to animate virtual characters. The results are very impressive, but the actors need to wear the markers to create a calibration sequence in order to transfer the expressions to the characters. Liu *et al*. [267] introduced a real-time expression-transfer approach from 2D data which is adaptable to user-specific data. Their setup requires a preprocessing step for the acquisition of target-specific training images. The approach of Saito *et al*. [268] for real-time 3D facial performance capture from RGB data relies on accurate DNN-based facial region segmentation and is robust to occlusions and significant head rotations. Recently, some commercial facial performance capture software has been released (*e.g.,* Apple's iPhone X app to animate a virtual character with its depth camera [269]).

In this work, we use a monocular setup and a lightweight energy-based minimization which can be used in affective user interfaces. Our approach runs on a single CPU at real-time rates while relying on robust facial landmark extraction. We do not require specialized hardware, preprocessing steps, manual intervention, large collections of training data or pretrained target-specific regressors. Thus, our method addresses several limitations of existing 2D-to-3D facial expression transfer approaches.

5.3 Overview of the Proposed Pipeline

Figure 5.1 provides an overview of the proposed pipeline. A set of blendshape targets that comprises a wide range of facial expressions is selected in advance. Then, in every incoming frame, we detect a sparse set of 2D facial landmarks for the recovery of rigid and non-rigid facial motion. We define a linear parametric model based on blendshapes and retrieve parameters mod-

eling the head pose and facial expressions by solving an energy-based optimization problem. Finally, we map the 2D facial expressions to a virtual 3D character which can be an animatable avatar or a person-specific 3D reconstruction obtained in a preprocessing step. Our method assumes perspective projection and known intrinsic camera parameters.

5.3.1 Blendshape Representation

Blendshape models provide a simple yet robust technique for facial animation. They allow to parameterize facial expressions by building a linear weighted sum of basis elements [270]. The set of D blendshape targets defines the valid range of expressions and limits face movements to a subspace of dimension D. Unlike PCA-based models, each basis shape encodes a semantically meaningful expression.

The face model is given by a column vector $\mathbf{f} \in \mathbb{R}^{3p}$ composed of p vertices with the coordinates vectorized as $[x_0, y_0, z_0, x_1, y_1, z_1, ..., x_p, y_p, z_p]^T$. Similarly, each blendshape target is denoted by a vector $\mathbf{b}_k \in \mathbb{R}^{3p}$. The absolute blendshape model is then defined as:

$$\mathbf{f} = \sum_{k=0}^{n} w_k \mathbf{b}_k, \tag{5.1}$$

where $0 \leq w_k \leq 1$ are the blendshape weights [270]. We arrange n blendshape targets into a matrix $\mathbf{B} = [\mathbf{b}_0, ..., \mathbf{b}_n] \in \mathbb{R}^{3p \times n}$ defining the expression semantics transferable to the avatar. $\mathbf{b}_0$ denotes a face with neutral expression and $\mathbf{b}_i \; \forall i \neq 0$ corresponds to different base expressions. After concatenating w_k into a vector $\mathbf{w} \in \mathbb{R}^n$, Eq. (5.1) can be rewritten as:

$$\mathbf{f} = \mathbf{B}\mathbf{w}. \tag{5.2}$$

Similarly to commercial animation software such as Maya [271] and SOTA methods [255, 265, 266], we use the delta form of the blendshape model, *i.e.,* each column of $\mathbf{B}$ is composed of offsets w.r.t $\mathbf{b}_0$: $\mathbf{B} = [\mathbf{b}_1 - \mathbf{b}_0, ..., \mathbf{b}_n - \mathbf{b}_0]$. As a result, multiple rows of $\mathbf{B}$ are composed of zero or near zero values. Then, Eqs. (5.1) and (5.2) read as follows:

$$\mathbf{f} = \mathbf{b}_0 + \sum_{k=1}^{n} w_k (\mathbf{b}_k - \mathbf{b}_0) = \mathbf{b}_0 + \mathbf{B}\mathbf{w}. \tag{5.3}$$

5.3.2 Registration of Blendshape Targets

The proposed pipeline leverages a set of blendshape targets collected from real face scans and registered to the FLAME model topology [272], provided by [273]. Although the variety of facial expressions in the dataset is not as high as in FaceWarehouse [264], the low number of vertices (~ 5023) makes them attractive for real-time applications on a single CPU. The scans in [272], although consistently vertex-aligned, are shifted for different expressions. To compensate for the slight misalignment, we register the scans by solving the following constrained orthogonal Procrustes problem [274]:

$$\mathbf{R} = \arg\min_{\mathbf{\Omega}} \|\mathbf{\Omega}\mathbf{A} - \mathbf{B}\|_{\mathcal{F}}, \text{ subject to } \mathbf{\Omega}^{\mathsf{T}}\mathbf{\Omega} = \mathbf{I}, \tag{5.4}$$

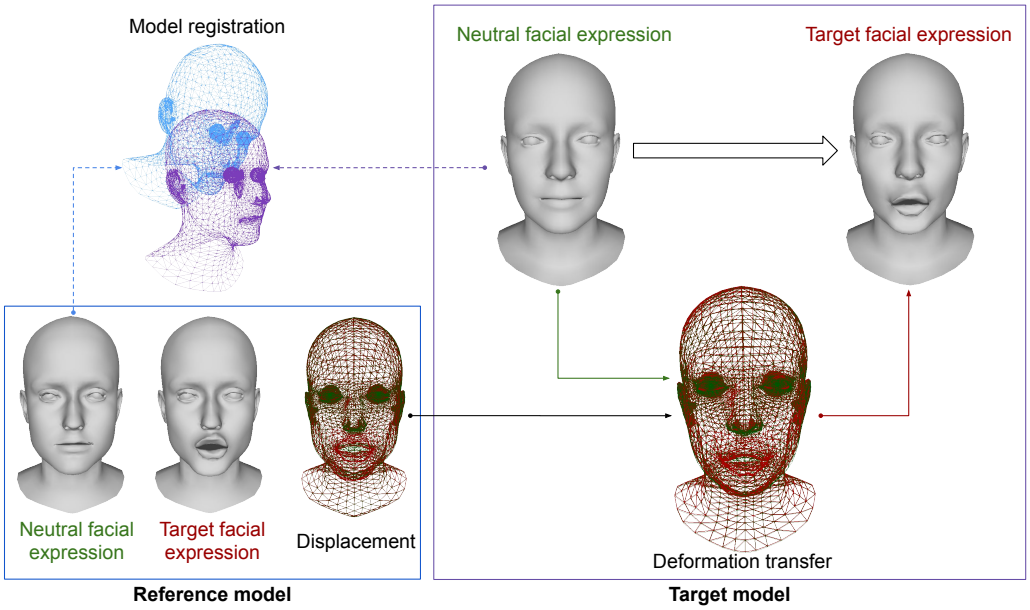


Figure 5.2: Registration of blendshape targets from a reference to a target model.

where **A** and **B** are the two blendshape targets to be registered, **R** is the orthogonal matrix that maps **A** to **B** and $\|\cdot\|_{\mathcal{F}}$ denotes Frobenius norm.

Since **R** is a rotation matrix, $\det(\mathbf{R}) = 1$. Hence, we can apply the Kabsch algorithm [275, 276] to decompose **R** with SVD. That is, we extract

$$\mathbf{R} = \mathbf{U}\Sigma'\mathbf{V}^{\mathsf{T}}, \tag{5.5}$$

where

$$\Sigma = \mathrm{diag}(1\,1\,\det(\mathbf{V}\mathbf{U}^{\mathsf{T}})). \tag{5.6}$$

Note that only a subset of points on the back side of the head is used for the alignment.

After all the scans are registered to a base model with neutral facial expression, we augment the set of facial expressions available in our base model. To that end, we select the desired set of facial expressions from models with different identities and map them to our target model, as shown in Figure 5.2. The facial expressions are transferred by finding first the vertices displacement between the reference models with neutral facial expression and with the target facial expressions. The displacement for every expression is then added to our base model with neutral facial expression, to generate the set of blendshape targets used in our experiments.

5.4 Target Energy Functional

We propose to minimize a multi-objective energy function $\mathbf{E}(\gamma)$ for $\gamma = (\mathbf{R}, \mathbf{t}, \mathbf{w})$, where **R** and **t** are the rotation and translation, *i.e.,* the head pose, and **w** is the vector of blendshape weights for the facial expression recovery:

$$\mathbf{E}(\gamma) = \omega_{\text{sparse}}\,\mathbf{E}_{\text{sparse}}(\gamma) + \omega_{\text{prior}}\,\mathbf{E}_{\text{prior}}(\gamma). \tag{5.7}$$

$\mathbf{E}_{\text{sparse}}$ is the data term that measures the model's head pose and facial expression from the input 2D facial landmarks. It consists of $\mathbf{E}_{\text{pose}}$ and $\mathbf{E}_{\text{fit}}$, as follows:

$$\mathbf{E}_{\text{sparse}}(\boldsymbol{\gamma}) = \omega_{\text{pose}}\,\mathbf{E}_{\text{pose}}(\mathbf{R},\mathbf{t}) + \omega_{\text{fit}}\,\mathbf{E}_{\text{fit}}(\mathbf{w}). \tag{5.8}$$

$\mathbf{E}_{\text{prior}}$ comprises a regularization term for the head pose, $\mathbf{E}_{\tau}$, as well as constraints on the blendshape weights, $\mathbf{E}_{\beta}$ and $\mathbf{E}_{\sigma}$:

$$\mathbf{E}_{\text{prior}}(\boldsymbol{\gamma}) = \omega_{\tau}\,\mathbf{E}_{\tau}(\mathbf{R},\mathbf{t}) + \omega_{\beta}\,\mathbf{E}_{\beta}(\mathbf{w}) + \omega_{\sigma}\,\mathbf{E}_{\sigma}(\mathbf{w}). \tag{5.9}$$

The weights $\omega_{\{\cdot\}}$ in Eq. (5.7)-(5.9) define the contribution of each energy term to $\mathbf{E}(\boldsymbol{\gamma})$.

5.4.1 Non-Rigid Tracking

We detect 2D facial landmarks using the off-the-shelf face alignment approach from Kazemi *et al*. [148] which aligns an ensemble of regression trees. We retrieve 68 facial landmarks around the jawline, lips, nose, eyes and eyebrows. Optical flow is then used to track the landmarks in consecutive frames, as described in Chapter 2.1.4. The correspondences between the 2D facial landmarks and points on 3D blendshape targets are known in advance per design.

5.4.2 Rigid Head Pose Estimation

An initial estimate of the rigid head pose is computed based on Chapter 3. A set of robust facial landmarks including eyes' canthi, both lateral and medial, and points around the nose, are used to minimize the reprojection error of the 3D-2D correspondences. For the other frames, we minimize the reprojection error of the $\eta = 68$ facial landmarks using Eq. (2.22), where $[\mathbf{R}|\mathbf{t}]$ are the extrinsic camera parameters (camera pose), $\mathbf{P}$ and $\mathbf{p}$ are the 3D and 2D corresponding facial landmarks, respectively, and i is the feature point index. As the calibration of the camera is known, Eq. (2.22) is minimized in the least squares sense using Levenberg-Marquardt iteration.

Inspired by [255], we include an additional term $\mathbf{E}_{\tau}$ to enforce temporal smoothness on the head pose:

$$\mathbf{E}_{\tau}(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{\eta} \left\| [\mathbf{r}|\mathbf{t}]_{t-2} - 2[\mathbf{r}|\mathbf{t}]_{t-1} + [\mathbf{r}|\mathbf{t}]_{t} \right\|_2^2, \tag{5.10}$$

with the angle-axis representation of the rotation $\mathbf{r} = [r_x, r_y, r_z]$ around the x, y and z-axes and t being the time frame.

5.4.3 2D-3D Transfer of Facial Expressions

To recover the facial expression, we minimize the reprojection error of the facial landmarks using the blendshape model in Eq. (5.3), for n blendshape targets:

$$\mathbf{E}_{\text{fit}}(\mathbf{w}) = \sum_{k=1}^{n} \left\| \pi(\mathbf{b}_0 + \mathbf{B}_i\mathbf{w}) - \mathbf{p}_i \right\|_2^2. \tag{5.11}$$

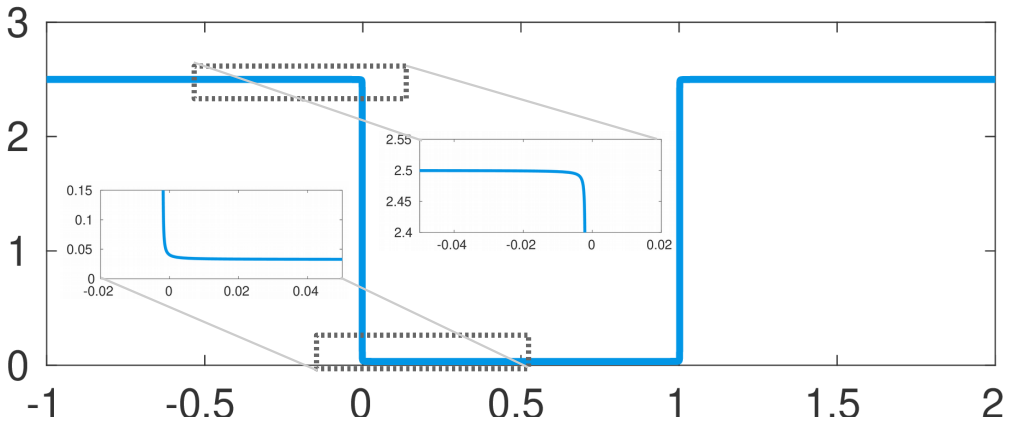


Figure 5.3: Our function – a sum of two Heaviside approximations – to limit the blendshape target weights **w** in the range [0,1].

Since the elements of the blendshape basis are not orthogonal, *i.e.,* not linearly independent, the same facial expression can be recovered using different target combinations. Thus, we include a sparsity prior based on [255] defined as a ℓ_1-norm:

$$\mathbf{E}_\sigma(\mathbf{w}) = \sum_{k=1}^{n} ||\mathbf{w}||_1. \tag{5.12}$$

To avoid compensation artifacts, the weights are usually set in the range [0, 1]. This implies that we need a differentiable function so that in the range [0,1] it generates a zero penalty, and a large penalty otherwise. We define such function by adding two smooth Heaviside function approximations [277]:

$$\mathbf{E}_\beta(\mathbf{w}) = \frac{\pi}{4}\left(\tan^{-1}\left(\frac{\mathbf{w}-a}{b}\right) - \tan^{-1}\left(\frac{\mathbf{w}+a-1}{b}\right)\right) + c, \tag{5.13}$$

with $a = 1.002, b = 2 \cdot 10^{-5}$ and $c = 2.5$. The function is shown in Figure 5.3.

In contrast to [254, 255], we do not use any temporal coherence constraints on the blendshape weights.

5.4.4 Energy Minimization

We solve an energy-based optimization problem for 50 parameters: 6 DoF for head pose and 44 parameters (the number of blendshape targets) for the facial expression, with a total of 68×2 residuals for $\mathbf{E}_{\text{sparse}}$, six for $\mathbf{E}_\tau$ and one for each $\mathbf{E}_\beta$ and $\mathbf{E}_\sigma$.

5.5 Experiments and Results

The pipeline was implemented in C++ using DLib [153] and the *ceres* solver [278]. The experiments were performed on a commodity computer with an Intel Xeon(R) W3520 processor and 8GB of RAM.



Figure 5.4: Identity set from the COMA dataset [273]. The original scans are part of the D3DFACS dataset [279].

Table 5.1: Averaged runtime of the proposed performance-driven animation pipeline.

Process	Time (ms)
Face alignment	23.7
Energy minimization	31.63
Runtime	55.33

5.5.1 Datasets

Our approach uses a pre-defined set of 44 blendshape targets with different facial expressions from the dataset in [273]. These blendshape targets were taken from the scans in the D3DFACS dataset [279] and registered to the FLAME model [272]. The dataset has models from 10 identities, shown in Figure 5.4. A representative set of the blendshape targets used in this work are shown in Figure 5.5. These targets have consistent topology and vertex-wise correspondences across different facial expressions, with 5023 vertices and 9976 faces.

We additionally collected a set of RGB video sequences, captured with a Logitech C920 HD Pro webcam at the resolution of 640 $\times$ 480 pixels. Representative images from our dataset are shown in Figure 5.8.

5.5.2 Runtime Analysis

For the aforementioned setup, the average throughput for $\sim$1000 frames amounts to 18 frames per second. The time each process requires is shown in Table 5.1.

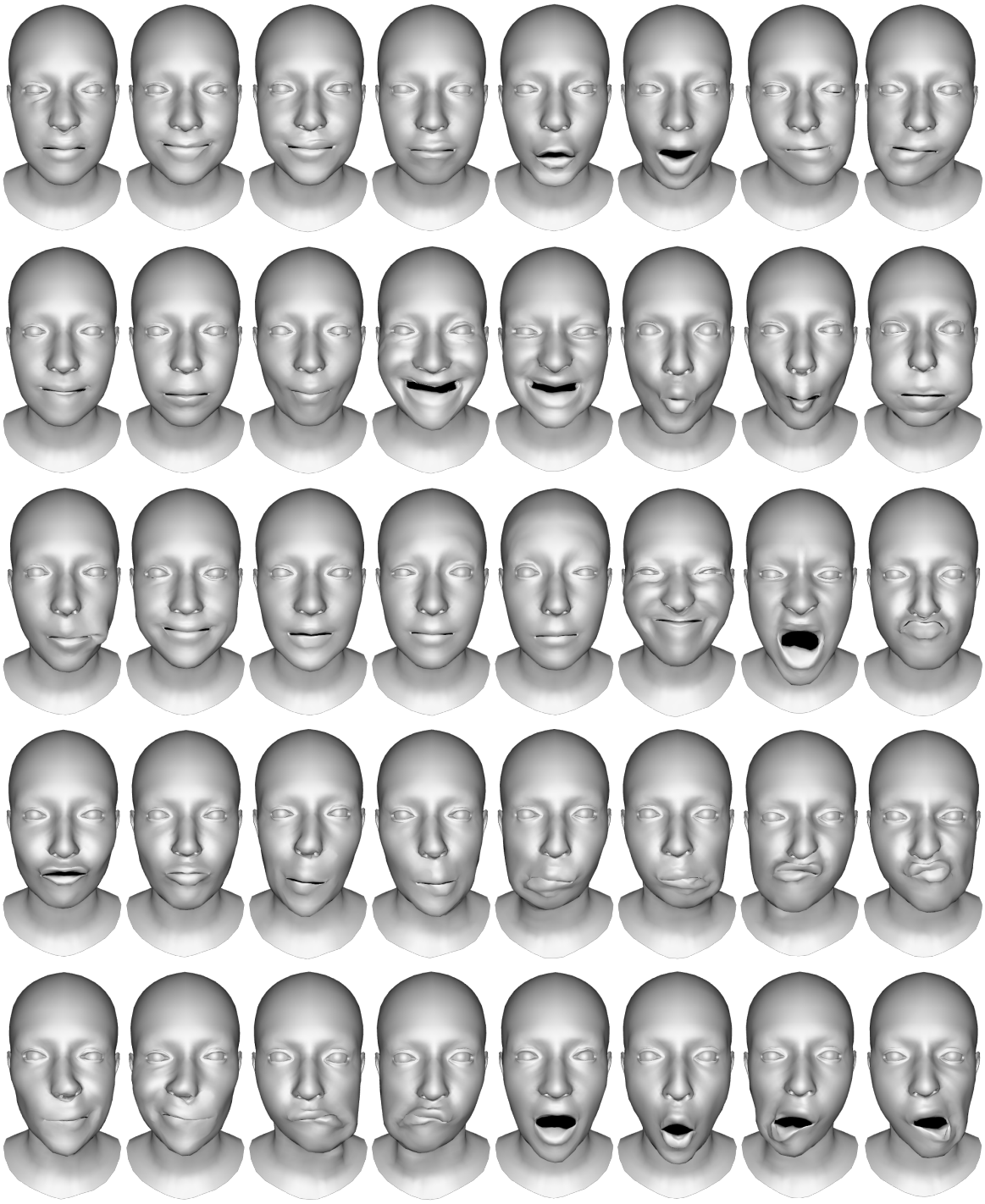


Figure 5.5: Blendshape targets.

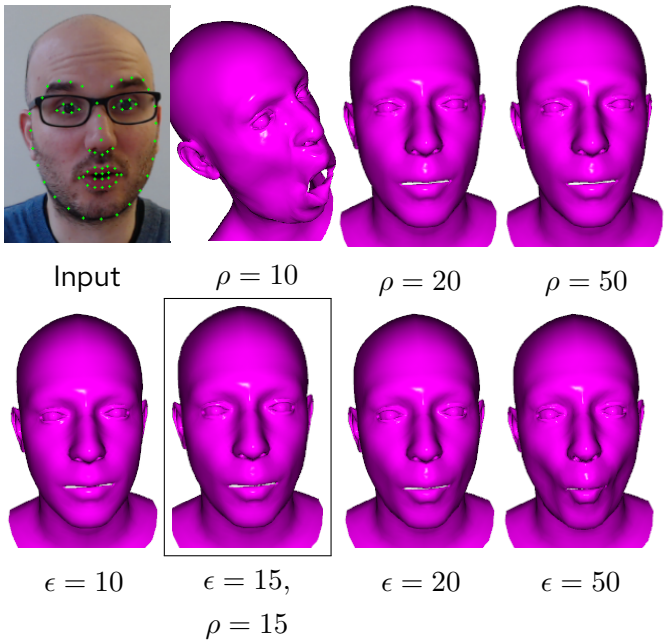


Figure 5.6: Energy minimization results under different parameters. ρ and ϵ are the number of iterations used to estimate the head pose and facial expression, respectively. Top: Head pose for a fixed $\epsilon = 15$. Bottom: Facial expression for a fixed $\rho = 15$. Selected parameters: $\epsilon = 15$ and $\rho = 15$.

5.5.3 Parameter Tuning

We investigated how the internal number of iterations in the energy function affects the output and runtime. Figure 5.6 shows the resulting head poses and facial expressions for one frame. To select a fixed set of parameters for all experiments, we consider the trade-off between accuracy and runtime. In Figure 5.7, the computation of the head pose requires around 15 iterations to converge, while the estimation of the blendshape target weights does not entirely converge during the first 50 iterations. Still, 15 iterations are sufficient to transfer similar facial expressions to the target, as shown in Figure 5.8.

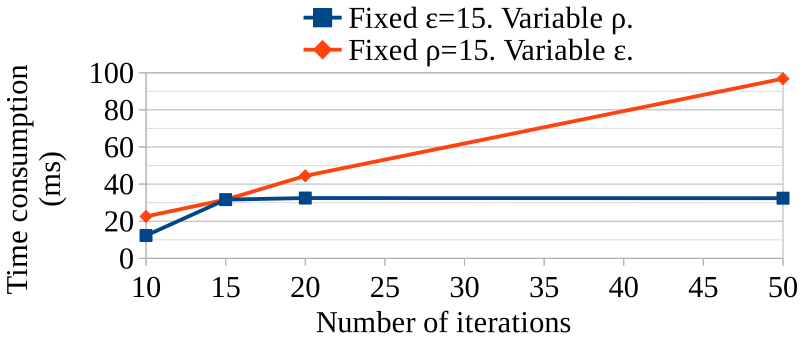


Figure 5.7: Runtime of the energy minimization for varying ρ and ϵ, averaged over ~1000 frames.

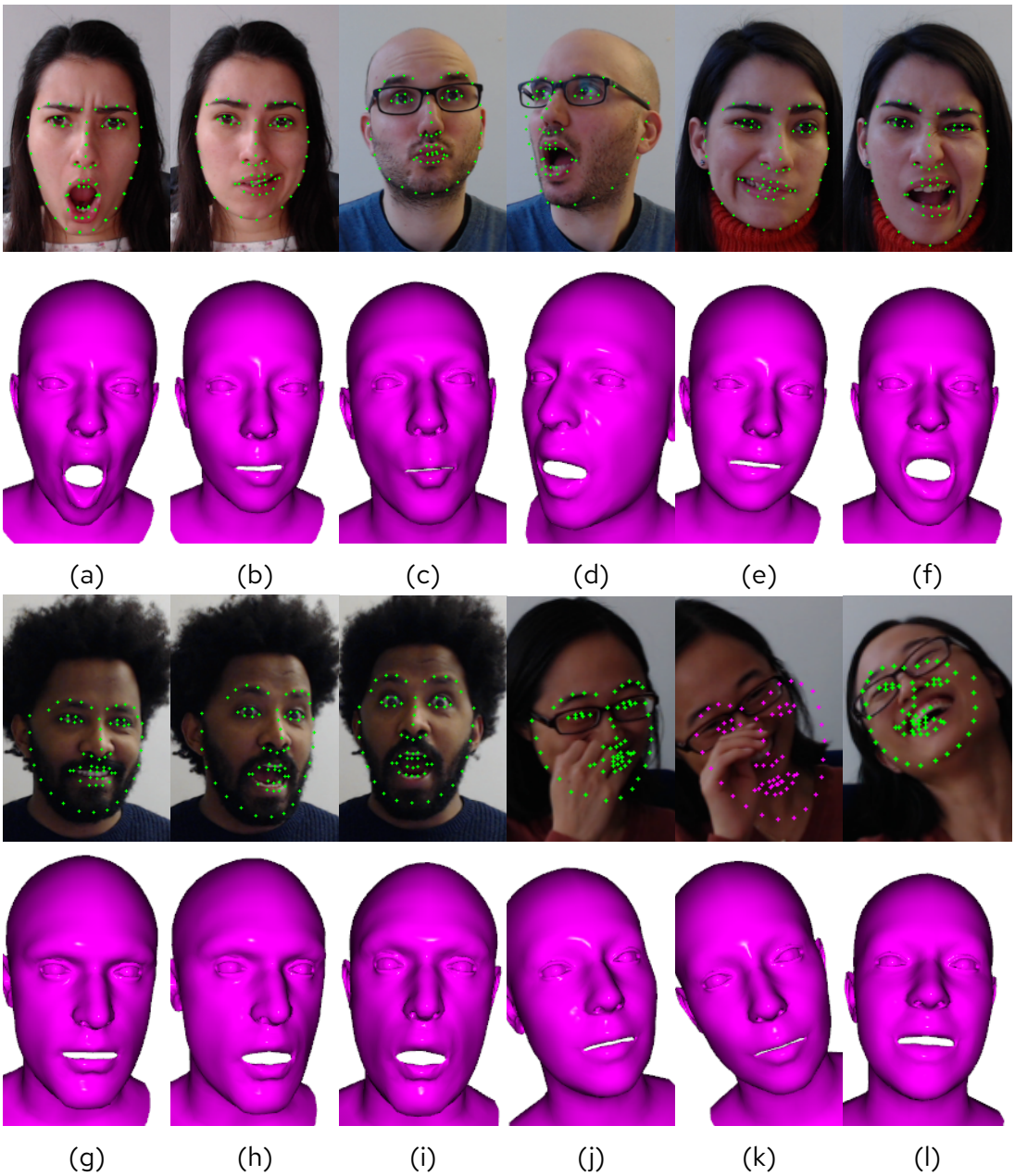


Figure 5.8: Results of our performance-driven facial animation approach. Top: Input images with detected facial landmarks. Bottom: Animated 3D characters. (a)-(i) demonstrate the variety of supported facial expressions; (j)-(l) show results under occlusion and extreme poses.

5.5.4 Qualitative Results

Qualitative results of our performance-drive facial animation pipeline are shown in Figure 5.8. Our method is able to detect and transfer multiple facial expressions, including slightly asymmetrical expressions.

5.5.5 Head Pose Evaluation

We evaluated head pose using the Boston University head tracking database [37], which contains 45 sequences of individuals performing different head movements. We use the mean absolute error (MAE) to compare the rotation to related approaches in the SOTA (see Table 5.2). We report translation errors (in inches) of 2.27, 0.90 and 2.04 for the x, y and z-axes, respectively. Our results are close to related work, although the compared methods are intended for face alignment and head pose estimation only, without any facial performance capture.

Table 5.2: Comparison of rotational MAE on the Boston University dataset.

Method	MAE$^{\downarrow}$			
	Roll	Pitch	Yaw	Average
Jeni *et al*. [75]	2.41	**2.66**	3.93	**3.0**
Wu *et al*. [131]	3.1	5.3	4.9	4.43
Gou *et al*. [132]	3.3	4.8	5.1	4.4
Diaz Barros *et al*. [17]	**2.32**	3.41	**3.90**	3.21
Ours	2.35	3.62	4.38	3.45

5.5.6 Discussion

As shown in Figure 5.8: (a)-(i), our pipeline can handle occlusions caused by glasses, long hair and beard. Although face alignment has limited performance for facial expressions with strong asymmetry, our method can transfer such expressions adequately (see Figure 5.8: (b), (e) and (f)). The performance of our approach is clearly affected by the accuracy of facial landmark detection and tracking, large head rotations and occlusions (see Figure 5.8: (j)-(l)). Similarly to other methods using RGB data, the method is sensitive to low illumination (Figure 5.8 (i)).

5.6 Conclusions

In this Chapter, we introduce a real-time pipeline for performance-driven facial animation for monocular systems. Our framework leverages a sparse set of 2D facial landmarks and a dataset of blendshape targets, to animate the face model. The facial landmarks are detected in every frame of an RGB stream input that is used for the animation. The head pose and facial expression recovery are formulated as a lightweight optimization problem with blendshapes. Our pipeline runs at $\sim$18FPS on a single CPU and does not require training data nor special hardware which makes it suitable for embedded systems, with potential for affective user interfaces.

6 Sign Language Production

Contents

6.1 Introduction 120
6.2 Related work 122
6.3 SynthSL and Rendering Pipeline 125
6.4 Sign Language Image Synthesis 128
6.5 Experiments and Results 130
6.6 Conclusions 137

Publication

This Chapter is based on the following work:

- Jilliam M. Díaz Barros, C-Y. Wang, J. Malik, A. Arafa and D. Stricker, "SynthSL: Expressive Humans for Sign Language Image Synthesis", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024.

6.1 Introduction

Sign language (SL) is one of the main communication channels for people with severe or profound deafness. It is considered a natural language, where signs result from the combination of body postures, hand gestures and facial expressions, giving a representation in the spatio-temporal domain. It differs from the spoken language in both grammar and word order, without a one-to-one mapping from sign to word. Instead, the spoken sentences can be translated to glosses, which represent the written form of the signs [14].

Since SL perception is visual, it is actively investigated in computer graphics and vision, particularly for three main tasks: production, recognition and translation. SL production aims at generating sign videos or human pose sequences from written or spoken sentences, while recognition and translation refer to the interpretation of sign videos. Sign recognition targets the identification of glosses, whereas in translation, signs are mapped to written or spoken language words, in some cases using glosses in an intermediate step.

Communication with the Deaf community is a bidirectional process, and both recognition and production are essential to increase accessibility. Production, in particular, is a crucial step for them to understand live speeches, news, cultural activities, among others, specially when no interpreters are available. Signs are preferred over closed captions or subtitles, as SL users are accustomed to their own natural language, while written text is not always easily comprehensible for them [280, 281].

Although SL is used worldwide, it is not universal, since signs can vary widely in different regions, and several types and dialects might co-exist in the same country [280, 282, 283]. Each variation has a defined structure and its own grammar. This poses a challenge in sign recognition and translation, since datasets cannot be used cross-linguistically.

In the era of deep learning, large-scale datasets are of vital importance for training and testing. However, collecting such datasets is usually expensive and time consuming. Additionally, SL datasets require expert signers and annotators to label the data, in order to avoid interpretation errors due to similarity in the signs [283]. To address this limitation, some datasets have been collected from multiple sources, such as learning channels for SL [283] or from news or weather broadcasts [284], usually providing only RGB videos. Acquisition of additional ground truth data such as 2D/3D skeletons, keypoints and body pose require specialized systems such as 3D motion capture (MoCap) data or custom setups with hundreds of cameras and sensors [285]. Unfortunately, these systems are not easily accessible.

In this Chapter, we introduce a large-scale synthetic dataset on SL with expressive humans, SynthSL, based on the SMPL-X [286] model. Inspired by realistic synthetic datasets in human analysis such as SURREAL [287], we propose a pipeline for augmenting real RGB-based sign language datasets to include other modalities and add rich ground truth data such as depth information, rendered models, pose, body-part segmentation, normal maps and 2D/3D body joints. With the proposed data generation framework, we aim to extend existing datasets and advance research on SL production and interpretation. Furthermore, our rendering pipeline can contribute to close the gap between animated avatars used for SL production and synthetic sign

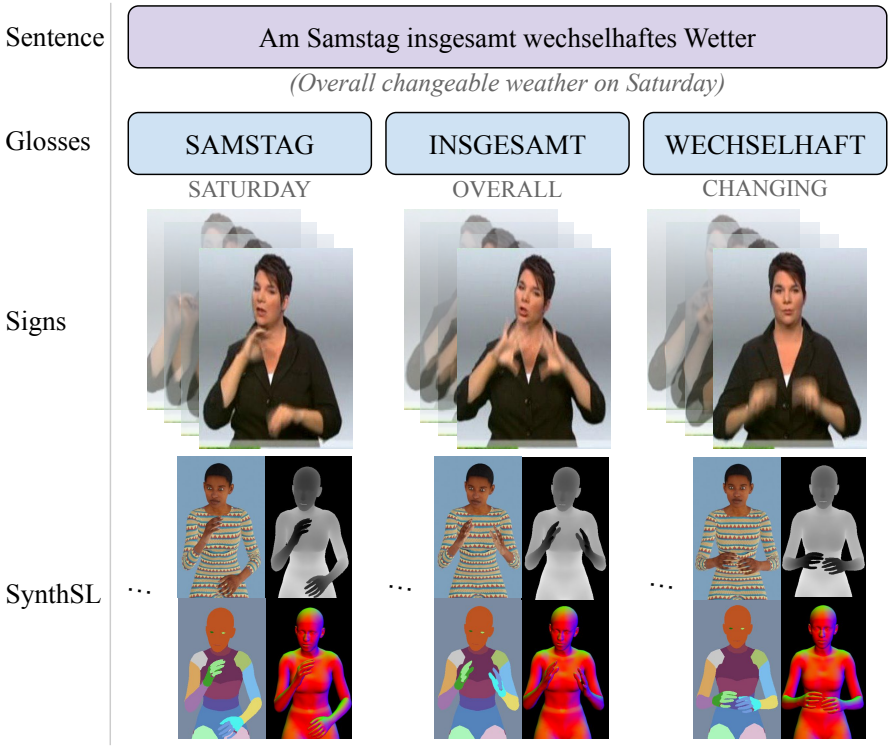


Figure 6.1: In sign language, the representation of sentences is encapsulated by glosses, which map sentences to morphemes. Existing datasets usually provide RGB or RGB+D sign sequences and the respective translation to sentences and glosses, which can be exploited for sign language recognition, translation and production. We introduce a rendering pipeline and our dataset SynthSL, which extends the *PHOENIX 2014T* dataset [14] to include additional ground truth such as rendered RGB images, depth and normal maps, segmentation masks and SMPL-X [286] model parameters.

videos.

The use of synthetic data offers multiple advantages on SL research: *(i)* overcome the limitations of monocular RGB-based videos with extended ground truth data, especially for learning models; *(ii)* enable inter-signer variation. Body shapes and textures (with identity and clothing) are randomized, to increase the diversity in our dataset; *(iii)* ensure enough number of instances per word/sentence. Each sentence can be generated with different signers, adding random noise to the signs to increase variability. *(iv)* eliminate motion blurry, intrinsic to fast motions in video sequences; *(v)* control over the camera position and the acquisition of multiple views; and *(vi)* control of external factors such as illumination and background.

Finally, we propose a new pipeline for synthesizing sign language images, targeted to sign language production (SLP). This pipeline is based on Generative Adversarial Networks (GANs), where a base signer (or *style*) image, the current pose and a target pose are used as input. The model then generates an output image of the base signer in the target pose. We leverage the improved performance of transformers on vision tasks and introduce SwinGenerator, a new generator based on Swin Transformer V2 [56].

The main contributions of this work are:

- A new rendering pipeline based on the SMPL-X model, which integrates the shape and pose of the body, hands and face, in addition to facial expressions.
- SynthSL, a large-scale synthetic dataset for research on sign language. Our dataset expands the *PHOENIX 2014T* dataset [14] with ground truth on rendered models, segmentation masks, depth and normal maps, 2D/3D body joints, body pose and shape, which are provided on a frame basis.
- A new deep-learning based model for sign image synthesis, with a novel SwinGenerator. Our model takes as input a source and target pose, plus a style image, and generates an image of the signer in the given style and target pose.

6.2 Related work

In this Section, we review the state of the art in SLP and analyze current SL large-scale datasets. We additionally describe related work on human analysis with synthetic datasets.

6.2.1 Sign Language Recognition and Translation

The task of interpreting signs, either to extract glosses, or written or spoken sentences has been widely studied for several years. Related works are based on monochrome [288], RGB [283, 289, 14], RGB-D cameras [290, 291], and other systems, such as gloves with embedded sensors [292, 291]. The interpretation can be performed on letter (or finger-spelled words) [290], word [283, 289] or sentence-level [293]. The recognition of individual words is also known as isolated sign language recognition, while for sentences it is denominated continuous sign language recognition [283].

Classical approaches targeted recognition of individual glosses or finger-spelled words, usually by modeling on separate steps the spatial and temporal components of the signs. For the former, handcrafted features are designed and for the temporal component, statistical models such as Hidden Markov Model (HMM) and hidden conditional random fields are used. Deep-learning based models follow a similar approach, using CNNs for extracting features and architectures such as HMM, Connectionist Temporal Classification (CTC) or Dynamic Time Warping (DTW) for temporal modeling.

More recently, [294] introduced a SL recognition method which employs SMPL-X to get a 3D body model. [295] presented an encoder-decoder architecture which exploits optical flow videos for SL translation. For a comprehensive study on methods for sign recognition, we refer the reader to [296].

6.2.2 Sign Language Production

Traditionally, SLP has been focused on the animation of 3D avatars, using MoCap data [280, 297, 298] or pre-designed synthetic sequences [299, 300, 301]. Recent works have moved towards the generation of skeleton poses or realistic video sequences, using deep neural networks

(DNN). Some approaches rely in an intermediate step to translate spoken sentences to skeletal poses, using glosses. [302] produces sign videos from spoken language, by combining Neural Machine Translation (NMT) and a generative model. The sentences are translated to glosses using an encoder-decoder, which in turn are mapped to skeletal pose sequences using a dictionary. [303] overrides the look-up table and generates skeletal pose sequences via Motion Graph, which is conditioned by a sequence of gloss probabilities extracted from the written or spoken sentences. [304] maps glosses to signs using a tailored DNN.

Other works explore the direct transformation from text to skeletal poses, without an intermediate gloss representation. [305] used an architecture based on neural networks to generate fixed-length sign pose sequences. [306] introduced progressive transformers for SLP, from discrete spoken sentences to continuous sign pose sequences. They also propose a counter decoding to enable the generation of sequences of variable length. [307] extended the framework to integrate a Mixture Density Network (MDN) to the transformer. [308] proposed an adversarial approach to exploit the multi-channel features of SL, to include facial expressions and mouthing in the sign pose sequences.

More relevant to our work is the pose-conditioned human synthesis for SLP. PSGN [302] proposed a hybrid architecture composed of a Variational Autoencoder (VAE) [60] generator and DCGAN [309], to synthesize sign sequences conditioned on the skeletal information and appearance. [310] produces realistic sign videos using the GAN architecture for human motion transfer in *Everybody Dance Now* [311], while the synthesis pipelines in [303, 304] are based on pix2pixHD [312], with an encoder-decoder-based generator. Similarly, SignGAN [307, 313] generates realistic sign videos using a conditional GAN based on [311] and [312], given a style image and a sign pose sequence. It additionally introduces a keypoint-based loss to improve the synthesis of hands. SignDIFF in [314] is based on dual-condition diffusion models for SLP on American Sign Language (ASL). These methods use high-quality datasets to synthesize sign images with high resolution.

6.2.3 Sign Language Datasets

Datasets can be classified according to the language, if they were captured in a controlled or unconstrained (in the wild) scenario and if they are targeted to finger-spelling [315], word (isolated) or sentence level (continuous) sign language. Table 6.1 lists a set of large-scale datasets for isolated and continuous sign language, from different regions around the world.

Some datasets were collected from weather forecasts [323, 284, 319], TV broadcast [324, 323], videos on YouTube [289], or different online channels of the Deaf community [283]. Others datasets comprise videos recorded in controlled environments, with multiview RGB [321], or multimodal systems [322, 317, 318], and provide a large variety of ground truth data, e.g. depth maps, 2D/3D keypoints, facial landmarks, body and hand pose, usually extracted from Kinect v1/v2 and OpenPose [325]. *How2Sign* [285] includes data captured in a custom setup with hundreds of cameras and sensors placed in a geodesic dome.

Other datasets introduce complementary annotations, e.g. [326] extends *RWTH PHOENIX Weather 2014* [319] with hand shapes annotation, while [327] provides annotations to analyze

Table 6.1: Large-Scale Word and Sentence-level Sign Language Datasets. **Not publicly available*. **This information was inferred by the data provided in [303], where all the signers perform each sign 3 times.

Name	Language	Videos	Sentences	Signers	Vocab. size	Data type	Level	In the wild
DEVISIGN [316]	Chinese	24000	-	8	2000	RGB-D	Word	-
SMILE* [317]	Swiss German	12600**	-	42	100	RGB-D	Word	-
MS-ASL [289]	American English	25513	-	222	1000	RGB	Word	✓
WLASL [283]	American English	21083	-	119	2000	RGB	Word	✓
BosphorusSign22k [318]	Turkish	22542	-	6	744	RGB-D	Word	-
PHOENIX 2012 [284]	German	190	1980	7	911	RGB	Sentence	-
PHOENIX 2014 [319]	German	645	6861	9	1558	RGB	Sentence	-
PHOENIX 2014T [14]	German	8257	8257	9	1066	RGB	Sentence	-
CSL [320]	Chinese	25000	100	50	178	RGB-D	Sentence	-
KETI* [321]	Korean	14672	105	14	524	RGB	Sentence	-
How2Sign [322]	American English	2500	38611	10	4000	RGB-D	Sentence	-
GSL [296]	Greek	10290	10290	7	310	RGB-D	Sentence	-
SWISSTXT-WEATHER C4A [323]	Swiss German	183	811	-	1248	RGB	Sentence	-
SWISSTXT-NEWS C4A [323]	Swiss German	181	6031	-	10561	RGB	Sentence	-
VRT-NEWS - C4A [323]	Flemish	120	7174	-	6875	RGB	Sentence	-
BOBSL [324]	British English	1962	1.2M	39	2281	RGB	Sentence	-

facial expressions and emotions on the same dataset. [14] reformatted the dataset with new segmentation boundaries, to ease SL recognition and translation.

6.2.4 Synthetic Datasets for Human Analysis

Synthetic datasets have been widely used in recent years to compensate the scarcity of large-scale data for training deep learning models [328, 329]. Many human-based datasets synthesize images either from virtual characters generated with computer graphic software [330, 331, 332], or by using parametric body models such as SCAPE [333] or SMPL [334]. The latter has gained special attention for synthetic human datasets [287, 335, 336, 337, 338], due to its realistic appearance and its compatibility to existing rendering engines. SMPL is a statistical body model learned from thousand of 3D body scans and is based on blend skinning.

[287] introduced SURREAL (Synthetic hUmans foR REAL tasks), the fist synthetic dataset with realistic rendered humans performing different actions. Pose and motion were capture using MoCap data and additionally to the 2D/3D body pose, the dataset provides depth map, segmentation masks, surface normals and optical flow. They demonstrated that CNNs trained on synthetic data can be used for human body segmentation and depth estimation on real RGB images. [335] presented SURREACT (Synthetic hUmans foR REal ACTions), a dataset intended to investigate the scope of temporal CNNs trained on synthetic data for human action recognition. They exploited the synthetic data to include augmentation, such as multiple viewpoints

and variations in appearance and motion. [336] explored synthetic data for hand-object manipulation. They introduced a synthetic dataset, ObMan, which rendered the hands and body using the SMPL+H model [339], to produce synthetic RGB images, depth maps and segmentation masks. SMPL+H integrates a parametric hand model, MANO [339] to the SMPL model. The images were rendered using Blender and its python API. [337] and [340] explore synthetic datasets with optical flow fields for single and multi-human motion analysis (MHOF). The output was rendered using a similar approach to [336]. For MHOF, additional ground truth such as depth maps, segmentation masks, normal maps, and SMPL+H shape and pose parameters are provided. This ground truth was created using the SMPL+H model and MoCap data.

[341] investigated the performance of multi-person pose estimation networks trained on a purely synthetic dataset and in an augmented dataset based on real images. Based on [342], they additionally presented a style-transfer method to map synthetic to real humans. BEDLAM [343] introduced a dataset of synthetic humans based on the SMPL-X model, with high level of realism. The bodies are animated using datasets collected with MoCap. Their experiments show that simple models trained on this synthetic data can improve more elaborated SOTA approaches in human pose and shape (HPS) estimation.

Previous work demonstrates that human-based synthetic datasets help improve DNN performance in vision-related tasks. With our dataset and rendering pipeline, we aim to help advance the research on SL, by providing additional types of ground-truth information to existing datasets.

6.3 SynthSL and Rendering Pipeline

In this Section, we introduce a pipeline to render synthetic sign images, used to generate our large-scale dataset, SynthSL. Our rendering framework, depicted in Figure 6.2, leverages the parametric model SMPL-X [286], to extend RGB-based sign videos with additional ground-truth annotations. To that end, we exploit a state-of-the-art model fitting pipeline that regresses the SMPL-X parameters from RGB images. These parameters are then used to animate the human model and generate synthetic data. Our pipeline can be applied to videos from different sources, such as existing SL datasets, broadcasts and YouTube videos. Nonetheless, its use is not limited to SL-related tasks and can be employed in the generation of other human-related datasets. We note that our pipeline is compatible with other rendering pipelines and animation software.

SMPL-X Model

The SMPL-X model, taking its name from *SMPL eXpressive*, improves the SMPL model [334] by integrating detailed face and hand models to the given body template. The model was trained on multiple datasets of 3D human scans, incorporating the additional parametric models to increase the resolution of hands and face. Given a pose θ, shape β and facial expression ψ, the SMPL-X model is formulated as $M(\theta, \beta, \psi) : \mathbb{R}^{|\theta| \times |\beta| \times |\psi|} \rightarrow \mathbb{R}^{3N}$, with $N = 10475$ being the number of vertices.

The pose parameters θ embody the pose for K joints in the whole body, including jaw and

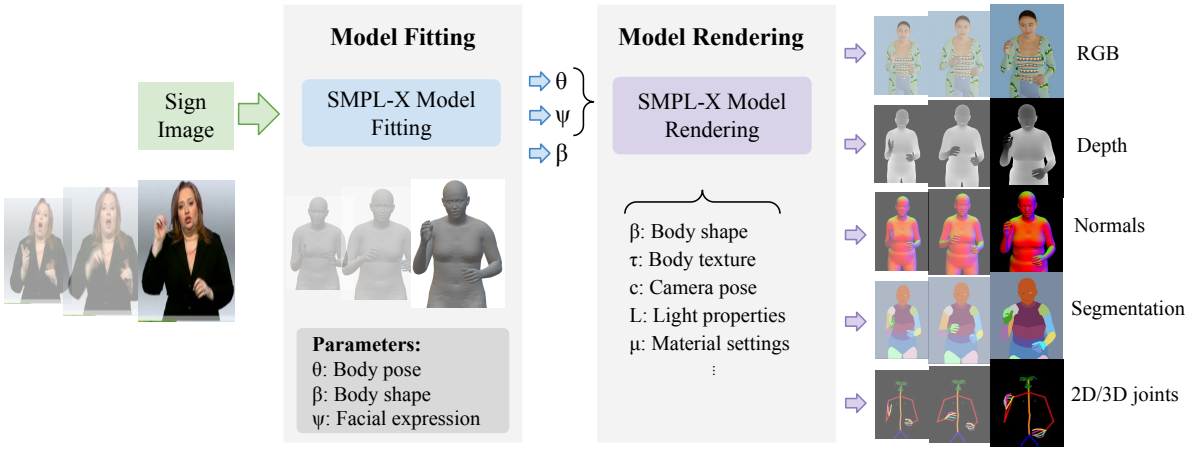


Figure 6.2: Rendering pipeline for generating synthetic data. Given an RGB sequence, a SMPL-X model fitting method estimates the body pose, shape and facial expression parameters per frame. The body pose and facial expression are passed to the animation framework, where multiple types of ground-truth data are rendered. Additional parameters such as body shape and texture are randomized per sequence. Input frames taken from *PHOENIX 2014T* [14].

finger joints. In total, SMPL-X has $K = 55$ joints, with one joint corresponding to the global orientation and the 54 left to body joints. 30 of these joints belong to the fingers, which are modeled based on MANO [339]. Such model uses a PCA pose space to reduce the dimension of the axis-angle rotations from 90 (3 angles per joint) to 24 parameters. Similarly, the facial expression is represented using a lower dimension PCA space with 10 parameters based on FLAME [272].

SMPL-X has in total 119 parameters: 99 for the pose (75 for { body, eyes and jaw } and 24 for the hands), 10 parameters for the shape and 10 for the facial expression. In addition, it has the option to use a female, male or neutral body model.

Augmented dataset

SynthSL was generated from the training set of the publicly available dataset *PHOENIX 2014T* [14]. This dataset is targeted to continuous recognition and translation on German SL, and comprises sign videos describing the weather forecast. Even though this dataset has a large vocabulary size (1000+), the number of signers is relatively low (9) and is only RGB-based.

Shape, pose and expressions

Our dataset is based on the SMPL-X model, which comprises body, face and hands. It differs from related human body rendering pipelines such as SURREAL [287] and Obman [336], which use the SMPL (body) and SMPL+H (body and hands), respectively.

The model parameters are estimated using SMPLer-X [344], a state-of-the-art fitting approach based on vision transformers (ViT) [54]. This method has currently the best performance for body modeling in the AGORA benchmark [345], which evaluates SMPL-X model fitting on a synthetic dataset with high realism. In SynthSL, we estimate the pose θ and facial expression ψ from SMPLer-X with ViT-Small. These parameters are then passed to the ren-



Figure 6.3: Selection of signers from SynthSL.

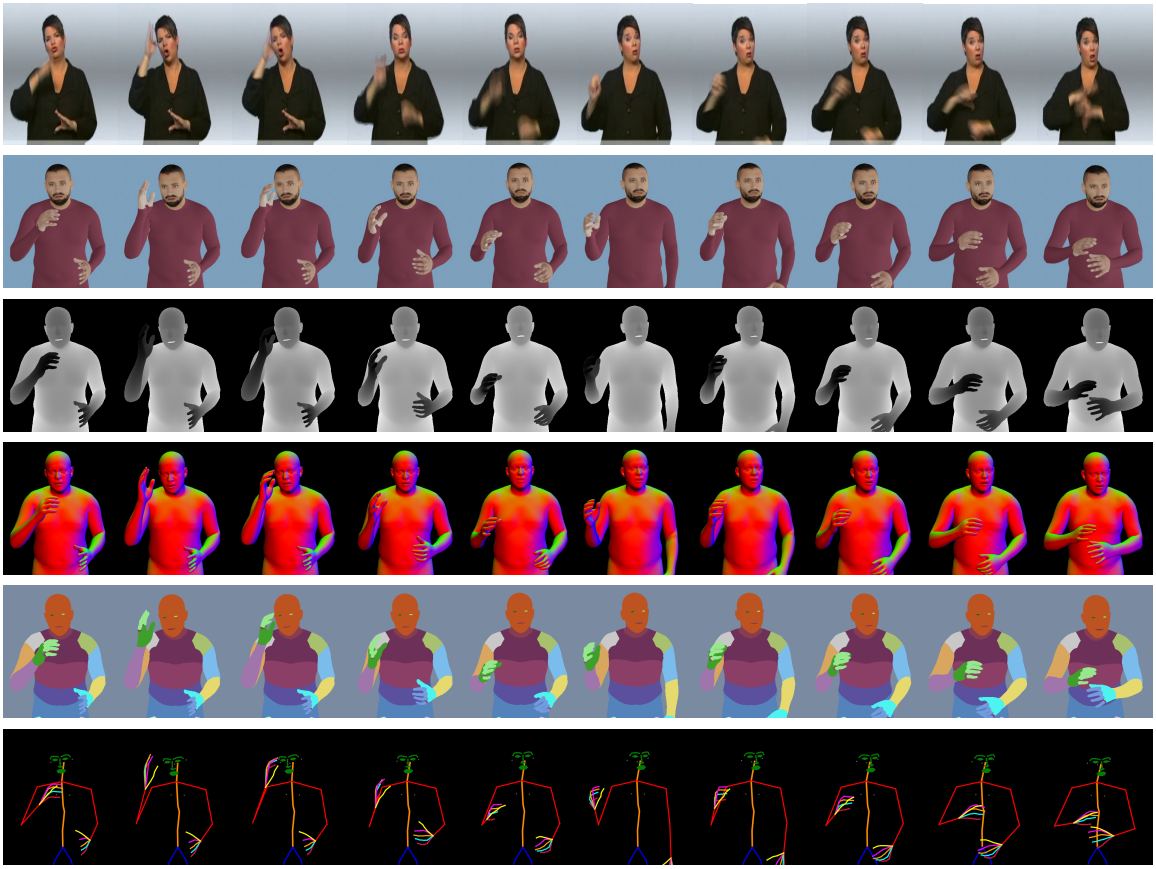


Figure 6.4: Selected frames from SynthSL.

dering step. The body shape, on the other hand, is randomly generated for every sequence to increase the diversity of the signers.

Texture and background

The body textures are randomly sampled from the BEDLAM dataset [343]. It includes 50 skin textures for male and 50 for female models, with large diversity in terms of identity and skin tones. In addition, the dataset provides 1738 clothing textures which are overlaid on the given skins.

Our rendering pipeline offers the option to randomly select backgrounds from ImageNet [251] or LSUN [346]. In our dataset, we fixed the background to only one specific color, to simulate the controlled recording scenario in sign videos. Some samples images of the signers are shown in Figure 6.3.

Rendering

The RGB images are rendered using the Python API from Blender [347], similarly to [336]. However, our pipeline generates multiple types of ground truth data, such as segmentation masks, depth and normal maps, following SURREAL and MHOF, as shown in Figure 6.4. In contrast to these pipelines, the facial expression and jaw pose are also integrated, to render expressive models. The RGB images, segmentation masks, depth and normal maps are rendered with a default resolution of $512 \times 512 \times 3$. SMPL-X model parameters such as body pose, shape, facial expression and 2D/3D body joints are provided in SynthSL as well. SMPL-X is available in Autodesk's Filmbox (FBX) file format, which enables the animation in software such as Blender and Maya [348].

Our rendering pipeline provides multiple benefits to SL research: (1) to augment existing RGB-based datasets; (2) to assist in the design of DNN architectures, where various types of data could be integrated into SL-based models; and (3) to guide in the collection of new datasets, where different technologies and algorithms could be evaluated beforehand.

6.4 Sign Language Image Synthesis

In addition to the rendering pipeline and our synthetic dataset SynthSL, we investigate the synthesis of sign images for SLP. This task can be divided into three main stages: *(i)* transcription of spoken to written words, *(ii)* translation from written words to glosses and *(iii)* mapping from glosses to signs for production. In this work, we focus on the third stage, the generation of sign images and videos from glosses. More specifically, we investigate the generation of sign images from skeleton poses representing the glosses.

Inspired by [302], we designed a GAN architecture [349] to generate sign images from an input base signer and a target pose. This problem is often termed as image to image translation, where the generator G maps the input data to a given visual domain, conditioned on a given input data. The discriminator D decides whether the generated image is fake or not, and alternates gradient updates with the generator during training. Our pipeline introduces the pose of the base signer as an additional input to G, as shown in Figure 6.5. We additionally propose a new generator architecture for image synthesis, based on Swin Transformers [56, 55].

6.4.1 Generator and Discriminator Networks

The generator G and discriminator D are trained in intervals with an adversarial training, where G tries to maximize the number of images that deceives D, and D penalizes G for generating unrealistic images [61].

D is trained using the adversarial loss [61], given by:

$$\mathcal{L}_{adv} = \mathbb{E}_y[\log(D(y))] + \mathbb{E}_x[\log(1 - D(G(x)))], \tag{6.1}$$

where $\mathbb{E}[\cdot]$ is the expected value and $D(\cdot)$ is the probability estimated by the discriminator that the given input corresponds to real or generated data. In the first term in (6.1), D takes as input the ground-truth sign image, y. This term enhances the ability of D to correctly identify real

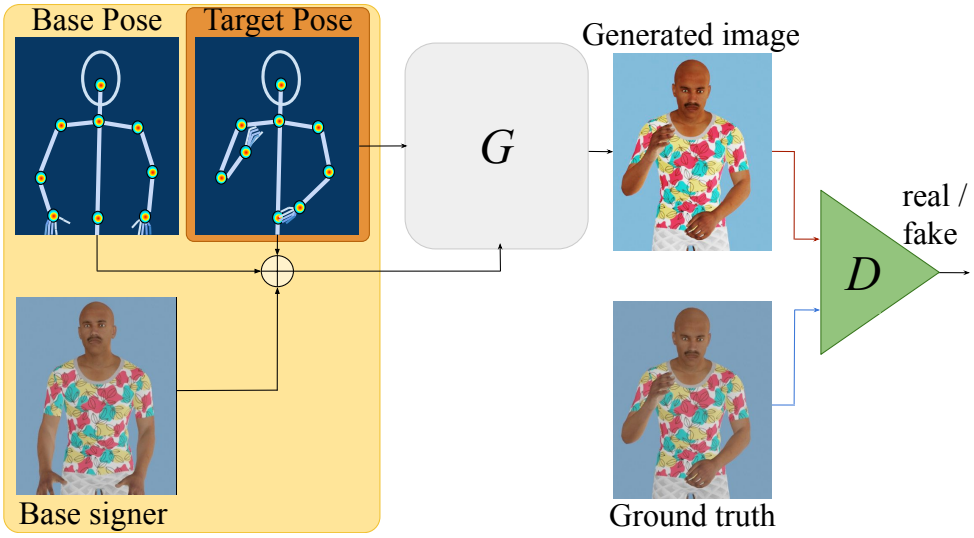


Figure 6.5: Proposed framework for the synthesis of sign language images.

images. In the second term, D assesses the image reconstructed by the generator, $G(x)$, and is used to improve the capability of D to distinguish $G(x)$ as fake or real.

In our architecture, the input of the generator G corresponds to the concatenation of the base signer, base pose and target pose, x, while the output $G(x)$ is an RGB image depicting the base signer in the given target pose. We propose to train G with the loss function:

$$\mathcal{L}_G = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{perc}\mathcal{L}_{perc}, \tag{6.2}$$

with $\mathcal{L}_{rec}$ being the reconstruction loss given by the $L1$ pixel-wise distance between the ground truth and reconstructed image at $G(x)$; $\mathcal{L}_{adv}$ is the adversarial image loss in (6.1); $\mathcal{L}_{perc}$ the perceptual loss [239]; and $\lambda_{(\cdot)}$ is the weight of each loss.

The final objective is then formulated as:

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv} \arg\min_G \max_D \mathcal{L}_{adv} + \lambda_{perc}\mathcal{L}_{perc}. \tag{6.3}$$

6.4.2 Generator Network

We introduce a new generator model, SwinGenerator, as depicted in Figure 6.6. The backbone of our generator G is based on Swin Transformer V.2 [56], which improves previous models such as ViT, by integrating shifting windows [55]. Since Swin Transformer applies the attention mechanism to patches of the window instead of the full image, the model complexity is reduced, with improved computational performance. This enhancement allows the use of this model on higher-resolution sign images.

In our SwinGenerator, the Swin Transformer V.2 serves as the encoder, with each block containing paired W-MSA (Windows Multi-Head Attention) and SW-MSA (Shifted Windows Multi-Head Attention) for feature extraction. Patch merging is used for down-sampling, reducing the size of the output feature map by half and doubling the number of channels.

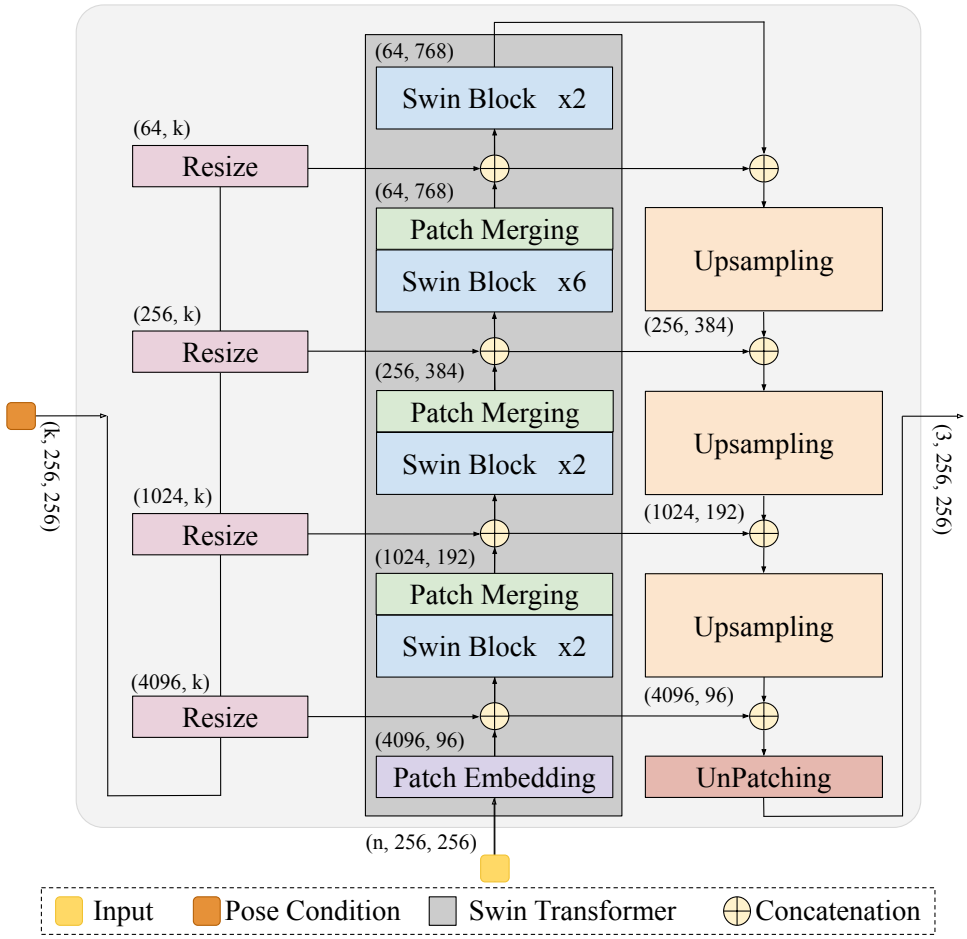


Figure 6.6: Architecture of the proposed SwinGenerator.

To enable the generation of images conditioned on a given pose and appearance, we propose to add upsampling and un-patching layers, which act as a decoder and map the embedded code to the synthesized sign image. The upsampling layers are based on transposed convolutions, followed by two convolution layers, and the un-patching layer recovers the original image structure from the patches.

We additionally integrate skip connections between the corresponding down-sampling and up-sampling layers, to improve the convergence of the model. Furthermore, the target pose is resized and concatenated to the output of every Swin block and to the corresponding skip connection required for the decoder.

6.5 Experiments and Results

In this section, we provide the details of the proposed synthesis architecture and describe the experiments conducted for synthesizing SL images. The proposed synthesis pipeline was developed in PyTorch and trained on a Nvidia RTX A6000 GPU. Inference time was computed on an Nvidia RTX2070.

6.5.1 Datasets

We trained and evaluated our synthesis pipeline using SynthSL, *PHOENIX 2014T* and *BosphorusSign22k*. Images in *PHOENIX 2014T* have resolution of 210×260×3, while *BosphorusSign22k* has full HD images of 1920×1080×3 pixels. Since *BosphorusSign22k* is very large, we used 4% of the training data, which we found was sufficient for generating realistic sign language images. Nonetheless, we used the complete test set for evaluation. In SynthSL, we selected 1000 sequences, with 600 for training, 100 for validation and 300 for testing. The training set alone has 79362 frames in total, with an average of ∼132 frames per sequence. The test set has 35848 frames, with ∼119 frames per sequence..

6.5.2 Data pre-processing and model architecture

During pre-processing, input images were resized to 256×256×3. For *BosphorusSign22k*, we trimmed 1/4 from each side, to have square images. The face and body keypoints were resized accordingly.

The backbone of our SwinGenerator is a SwinV2-T, with number of channels C = 96 and block = $\{2, 2, 6, 2\}$. Its input corresponds to the base signer RGB image, plus the base and target poses. The output is an RGB image of the signer in the new pose, with the same size of the base signer image, $256 \times 256 \times 3$. The base and target poses have each $k = 96$ 2D keypoints extracted with MediaPipe [350], where 42 belong to the hands, 46 to the face and 8 to the body. These keypoints are mapped to Gaussian heatmaps, with one channel per keypoint.

The discriminator is composed of 2 convolutional layers and 3 residual blocks. It takes as input the ground truth image and the generated image, as depicted in Figure 6.5. The discriminator is updated one time for every generator update.

We used the Adam optimizer [252] with learning rate α_G = 2e-3 for G and α_D = 2e-4 for D. In (6.2), we set λ_{rec} = 5, λ_{adv} = 2 and λ_{perc} = 0.5. All the models are trained for 30 epochs, with batch size of 32. For the perceptual loss, we used layers $\{1, 3, 5, 9, 13\}$ from VGG19 [240].

6.5.3 Baseline architectures

Since existing sign language synthesis models are not publicly available, we implemented a baseline architecture inspired by [302]. We introduced five main changes to the architecture in [302]: (i) the generator is based on U-Net [50] instead of VAE, similarly to pix2pix [349]; (ii) the keypoints are mapped to Gaussian heatmaps, instead of binary heatmaps; (iii) we used the same keypoints $k = 96$ as in our architecture, and not 10; (iv) we added convolutional blocks in the input of the encoder and output of the decoder, to generate higher quality images of size 256×256×3, instead of 128×128×3; and (v) we introduced the perceptual loss to train the generator. We refer to this model as "U-Net".

Additionally, we evaluated a second baseline generator model, named as "ResNet". This models is composed of deep residual blocks [51] in the encoder, following related works on image synthesis [239, 312, 351] and SLP [313, 303], while the decoder is given by transposed convolutions.

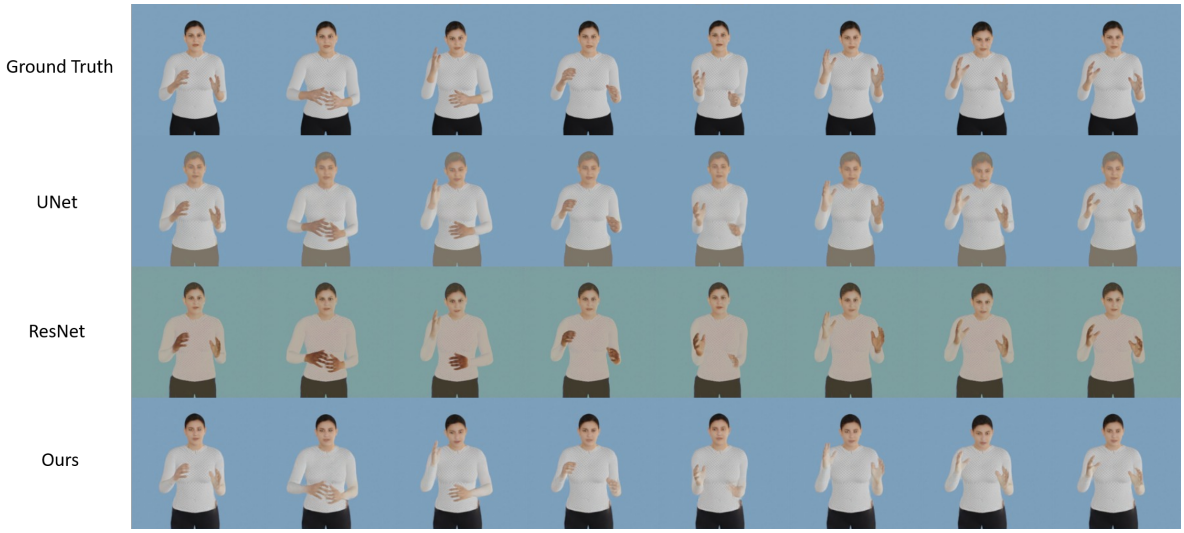


Figure 6.7: Sign images synthesized with the proposed pipeline and baseline architectures on SynthSL.

In both models, the input is given by the concatenation of the base signer and target pose, following [313, 302, 303]. We use the same discriminator and loss function in (6.3) as in our model for training and evaluation, for a fair comparison.

6.5.4 Metrics

The performance of the sign image synthesis pipeline is evaluated using the structural similarity index metric (SSIM), the peak-signal-to-noise ratio (PSNR), the position errors (PE) and the Fréchet Inception Distance (FID), described in Chapter 2.5. In the SSIM, PSNR and FID the generated RGB image is compared to the respective ground truth RGB image in the given pose. For the PE, we extracted the keypoints from the ground truth and generated images using MediaPipe, and summed the Euclidean distance between the respective keypoints.

6.5.5 Evaluation

The quantitative results of the proposed synthesis approach on *PHOENIX 2014T*, SynthSL and *BosphorusSign22k* are reported in Tables 6.2, 6.3 and 6.4, respectively, under SwinGenerator.

Qualitative results for the three datasets are shown in Figure 6.7, 6.8 and 6.9. In contrast to *PHOENIX 2014T* and *BosphorusSign22k*, SynthSL has large diversity in terms of signer and clothes. Furthermore, the images do not suffer from blurriness, enabling the synthesis of sharp hands without additional datasets.

Generator architecture

We evaluated the influence of the proposed SwinGenerator and compared against the baseline generator architectures based in U-Net and ResNet, described in Section 6.5.3. The results for *PHOENIX 2014T*, SynthSL and *BosphorusSign22k* are shown in Tables 6.2, 6.3 and 6.4, respectively. Qualitative results are shown in Figure 6.8, 6.7 and 6.9.

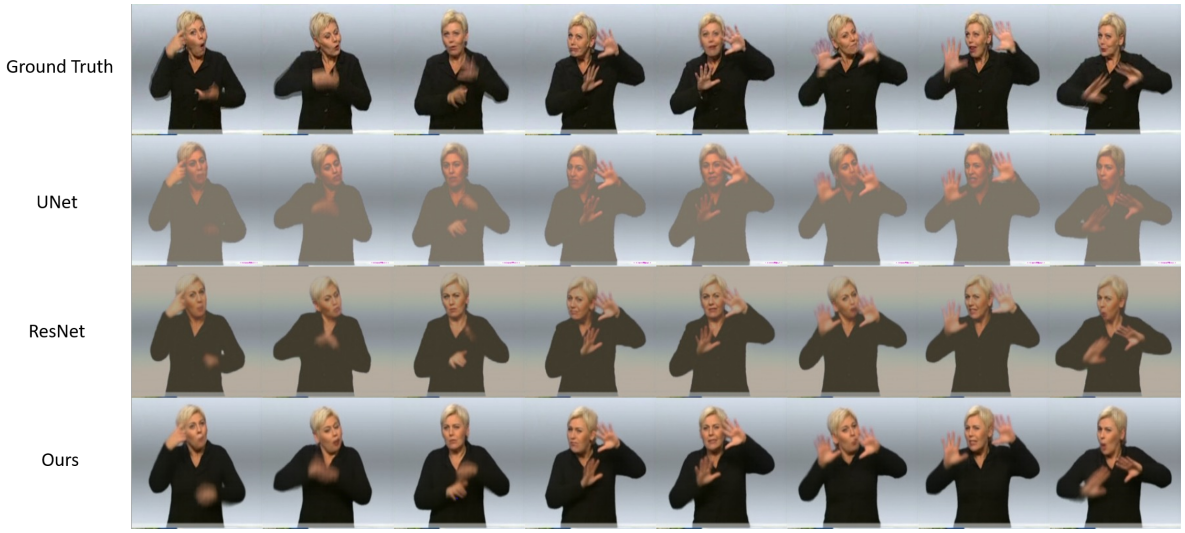


Figure 6.8: Sign images synthesized with the proposed pipeline and baseline architectures on Phoenix 2014T.



Figure 6.9: Sign images synthesized with the proposed pipeline and baseline architectures on BosphorusSign22k.

Table 6.2: Quantitative evaluation of different generator architectures on Phoenix 2014T.

Method	SSIM↑	PSNR↑	PE↓	FID↓
U-Net	0.684	30.999	0.884	42.624
ResNet	0.745	28.692	0.832	47.624
SwinGenerator	**0.835**	**33.971**	**0.675**	**31.678**

Table 6.3: Quantitative evaluation of different generator architectures on SynthSL.

Generator Architecture	SSIM↑	PSNR↑	PE↓	FID↓
U-Net	0.883	35.071	0.489	16.301
ResNet	0.897	33.050	0.469	14.160
SwinGenerator	**0.911**	**36.382**	**0.391**	**7.544**

In all datasets, the ResNet-based architecture showed an improved performance w.r.t. U-Net in terms of SSIM and PE. For the PSNR and FID, the results vary in each dataset. Our SwinGenerator outperforms both architectures in all datasets in every metric. Furthermore, the image quality is degraded for both U-Net and ResNet using the same GAN architecture. In many cases, the SwinGenerator provides steadier hands reconstructions (see first column of *PHOENIX 2014T* in Figure 6.8), specially if the hands suffer from blurriness. However, in some cases the ResNet-based architecture generates slightly better hands as in the last column of *BosphorusSign22k* in Figure 6.9.

6.5.6 Ablation Study

Transformer model

In this ablation study, we investigated the influence of different modules within SwinGenerator for synthesizing images using SynthSL.

We first evaluated the SwinGenerator without concatenating the resized target pose in the residual connections. This architecture is shown in Table 6.5 as Swin-T$_{noTP}$. The results show that adding the target pose in the residual connections is advantageous for our architecture, improving every metric, especially the PE and FID.

We additionally implemented a pipeline with only the source image as input, and the residual connection incorporating the difference between target pose and base pose. This model is

Table 6.4: Quantitative evaluation of different generator architectures on BosphorusSign22k.

Generator Architecture	SSIM↑	PSNR↑	PE↓	FID↓
U-Net	0.769	29.673	0.420	34.161
ResNet	0.865	32.150	0.302	21.356
SwinGenerator	**0.930**	**37.589**	**0.211**	**15.824**

Table 6.5: Quantitative evaluation of different SwinGenerator architectures on SynthSL.

Generator Architecture	SSIM↑	PSNR↑	PE↓	FID↓
Swin-T$_{noTP}$	0.892	36.170	0.511	16.569
Swin-T$_{TPres}$	0.895	36.362	0.468	12.633
SwinGenerator	0.911	36.382	0.391	7.544

Table 6.6: Quantitative evaluation with different sizes of Swin V2 on SynthSL.

Swin Architecture	SSIM↑	PSNR↑	PE↓	FID↓
Swin-T	0.911	36.382	0.391	7.544
Swin-S	0.912	36.810	0.387	7.346
Swin-B	0.918	37.114	0.374	6.304

shown in Table 6.5 as Swin-T$_{Res}$. This architecture reduces the model size, but sacrifices some level of quality compared to the proposed SwinGenerator.

Transformer backbone

Our architecture was also evaluated with different backbone sizes of Swin V2 on SynthSL. More specifically, we compared the tiny Swin-T with the small (Swin-S) and base (Swin-B) versions of Swin Transformers. The quantitative results are shown in Table 6.6.

As expected, an improved performance is observed when using larger Swin Transformer models. However, this comes at a computational cost, as the model size and complexity increase 2 and 4 times, respectively, when using Swin-S and Swin-B, in comparison to Swin-T. In Swin-S and Swin-B the number of layers are $\{2, 2, 18, 2\}$, while the channel number of the hidden layers in the first stage C is 96 for Swin-S and 128 for Swin-B.

Loss function

We additionally compared different adversarial loss functions to train the generator and discriminator on SynthSL. The results in Table 6.7 show that the choice of the adversarial loss does not have a significant impact on the performance of our pipeline.

Pre-trained model

Finally, we evaluated the effect on image synthesis when pre-training the networks in other datasets. Results are shown in Table 6.8. Overall, pre-training on other datasets result on an

Table 6.7: Quantitative evaluation of different adversarial loss functions on SynthSL.

Loss function	SSIM↑	PSNR↑	PE↓	FID↓
Vanilla	0.911	36.382	0.391	7.544
LS-GAN [352]	0.910	36.716	0.391	7.845
Hinge [353]	0.909	36.628	0.386	8.856
WGAN-GP [241]	0.913	36.803	0.394	7.927

Table 6.8: Quantitative evaluation on Phoenix 2014T and BosphorusSign22k with pre-trained models.

(Pre-)Trained			Fine-Tuned & Evaluated		SSIM↑	PSNR↑	PE↓	FID↓
SynthSL	Phoenix	Bosphorus	Phoenix	Bosphorus				
-	✓	-	✓	-	0.835	33.971	0.675	31.678
✓	-	-	✓	-	0.842	34.221	0.659	26.921
-	-	✓	✓	-	0.839	34.143	0.659	27.928
-	-	✓	-	✓	0.930	37.589	0.211	15.824
✓	-	-	-	✓	0.931	37.676	0.205	11.792

Table 6.9: Runtime analysis for image synthesis.

Generator Architecture	Model size (MB)↓	Inference time (ms)↓
U-Net	1108.45	43.59
ResNet	45.761	112.60
SwinGenerator	184.04	46.34

improved performance in every metric in *PHOENIX 2014T* and *BosphorusSign22k*.

6.5.7 Runtime Analysis

We estimated the average inference time for the different generator architectures after 300 iterations. The results are listed in Table 6.9.

We additionally provide the sizes of the generator models in Table 6.9. These results show that our SwinGenerator is able to perform in real time on GPU, with a frame rate of $\sim$ 22 FPS. Furthermore, the model size is considerably smaller than U-Net and with an inference time $\sim$2.5 times faster than ResNet.

6.5.8 Limitations and failure cases

In SynthSL, we observed some artifacts when the clothes of the signer had similar texture to the skin, as shown in Figure 6.10. This problem did not occur in *PHOENIX 2014T* and *BosphorusSign22k*, as the signers always wear dark clothes. These artifacts were also present in images generated with U-Net and ResNet. In *PHOENIX 2014T* and *BosphorusSign22k*, some images had noisy regions in the face or clothes of the signers, as shown in Figure 6.10. These artifacts were due to an overflow in the normalization layer and can be removed by including a post-processing step to limit the pixel intensities within the [0, 255] range.

6.5.9 Discussion

Our evaluation shows that the proposed SwinGenerator introduces a significant improvement w.r.t. the U-Net and ResNet-based generators, widely used for image synthesis, including SLP.

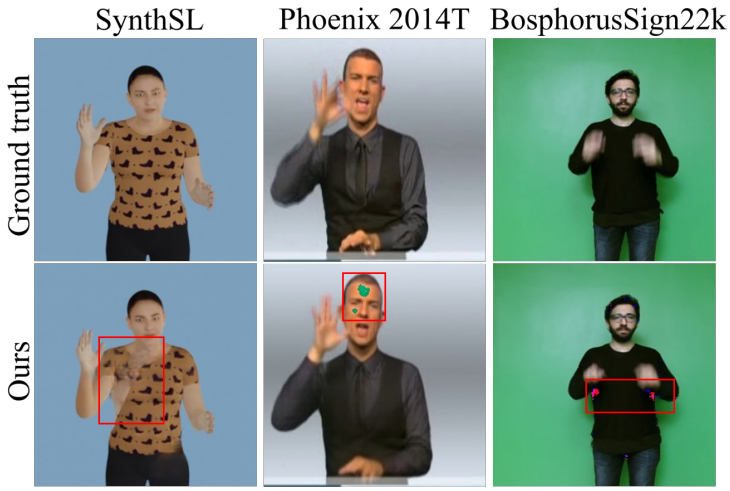


Figure 6.10: Failure cases.

These improvements were observed in all the evaluated datasets.

We additionally found that the ResNet-based generator tends to suffer from mode collapse with the current pipeline. In the literature, ResNet-based architectures for SLP [313, 303] are usually trained with the multi-scale discriminator from pix2pixHD [312]. In our experiments, we used a simple discriminator model to focus mainly on the performance of the evaluated generators. The multi-scale discriminator could be used as an alternative for an improved performance on the synthesis of high-resolution images.

Finally, we observed that on real datasets for SLP hand synthesis suffers from blurriness, due to the inherent fast motion on sign gestures. In that regard, [313] provides a framework and keypoint-based loss with outstanding hand reconstruction. However, a private dataset with high quality hands and manually selected frames without blurriness were used. Since this dataset is not publicly available, SynthSL could be used as an alternative. With our dataset SynthSL and the proposed rendering pipeline, we aim to provide additional resources for research on SLP.

6.6 Conclusions

In this Chapter, we introduced SynthSL, a large-scale synthetic dataset for research on sign language. We also proposed a rendering pipeline, to extend RGB-based sign videos to include rich ground truth data such as body, hands and head poses, segmentation masks, depth and normal maps. In addition, we presented a SL synthesis architecture conditioned on a given body pose and appearance. To that end, we leverage a GAN to generate realistic sign sequences with a newly introduced SwinGenerator for image synthesis. For future work, we would like to investigate the effect of additional input data such as depth map and segmentation masks for image synthesis. Furthermore, we aim to integrate additional ground truth data in our rendering pipeline such as optical flow, which can be exploited in SL-related tasks as in [295].

7 Summary and Conclusion

7.1 Summary

The analysis and modeling of the human body are of great interest in several scientific fields besides computer science, including disciplines in natural and social sciences. In computer vision, face analysis has been leveraged for several tasks, including the development of assistance tools and assistive technologies. Information such as the motion of the head under tasks that require total concentration could be exploited in driver-assistance systems, to detect the level of drowsiness of vehicle drivers. Facial asymmetry can also be analyzed to track the evolution of patients with unilateral facial paralysis. The generation of realistic sign images could improve accessibility to hearing-impaired individuals when signers are not available.

In this thesis, four separate focal points are covered: (i) the estimation of the rigid head motion, encompassing large head rotations; (ii) the detection and tracking of fiducial face landmarks in the wild, including cases with asymmetrical expressions; (iii) pose and expression transfer to animate an avatar in real time via a 3D morphable model; and (iv) the synthesis of the face and upper body for sign language production.

The modelling of the rigid and non-rigid motions of the head are targeted to improve the applicability on devices with limited resources. To that end, we introduced three methods for rigid head pose estimation in Chapter 3 and a pipeline for performance-driven facial animation in Chapter 5. These frameworks are based on the estimation of the head pose and expression through optimization methods that minimize the reprojection error using a sparse set of facial features.

For head pose estimation, we explore the use of predefined landmarks detected in every frame along with keypoints tracked in consecutive frames. Our best-performing pipeline integrates the pose estimated from keypoints into the prediction stage of a Kalman filter, while the pose computed from landmarks is added into the correction step of the filter. Our results show an improved performance w.r.t. related works and its suitability to execute in real time. In addition, we proposed a method that does not have any constraint for initialization, as facing the camera for the first frame, making it useful for head pose estimation under realistic scenarios.

For modeling the non-rigid face motion, we leverage a linear 3D morphable model based on blendshapes. Our approach transfers the pose and expression to animate a 3D avatar, where a newly introduced loss function enables real-time capabilities on CPU.

Furthermore, in Chapter 4 we presented an architecture for landmark detection, which refines the learned latent representation to encode more representative face features. This approach is targeted to face alignment on a wide range of facial expressions, including asymmetrical ones. We additionally introduced an approach to extend single-frame-based face alignment networks to video sequences. Our approach exploits the uncertainty of each predicted landmark, along landmark connectivity.

Finally, we proposed a framework for sign language images synthesis in Chapter 6, using a novel generator based on transformers. Given a style image and a pose representation, our

model generates the respective sign with the given style. We additionally introduced a rendering pipeline to extend existing sign language datasets, and generate synthetic data with different types of annotation data. Our new dataset SynthSL was created using the rendering pipeline, and used to evaluate the performance of the proposed sign image synthesis framework.

7.2 Concluding Remarks

Several methods of the state of the art rely on facial landmarks for head pose estimation. Even though they might be a reliable source for frontal and near-frontal faces, facial landmarks are sensitive to extreme head rotations and (self-) occlusions, where important reference regions of the face such as the eyes or nose are partially or totally occluded. In order to tackle this problem, we propose to integrate the head pose computed from a set of keypoints that can be tracked continuously, even when the facial landmarks are not visible. Although a keypoint-based head pose estimation approach could be used alone, it might suffer from drifting in long sequences [121, 122]. Accordingly, a mechanism to reinforce and correct the head pose from keypoint tracking using facial landmarks provides an improved performance. The combination of poses given by these two set of features introduces a synergy that enables the estimation of the pose in challenging scenarios.

The estimation and transfer of head pose combined with facial expression for avatar animation are time-consuming tasks, leading to the usage of additional resources to operate in real time. One of the main reason is that 3D head models are usually very detailed, to include a wide range of facial features and expressions. In this thesis, we introduced a lightweight pipeline that exploits a sparse set of facial landmarks and compact 3D head scans. The scans encompass a wide level of expressiveness, despite their low number of vertices, enabling the animation in real time on CPU on settings with low resources.

In contrast to 2D landmarks, the detection of 3D facial landmarks offers advantages for non-frontal head poses. Since these methods are usually trained on large-scale datasets with images in the wild, they are robust under illumination changes, partial occlusion and large head poses. However, extreme facial expressions are usually not covered in these datasets. Herein, we introduce a new dataset with patients of unilateral facial paralysis, ParFace, and propose a method for the detection of landmarks on a wide range of expressions, using a semi-supervised approach. Our dataset and proposed pipeline aim to bridge the gap for the development of assistive tools to track the recovery of patients at home.

Another limitation in many 3D face alignment approaches is that they are targeted to single-frame detections. This constrains their operation in video sequences with challenging frames, *e.g.*, with blurriness or large occlusion. We propose a pipeline that exploits the motion across frames to improve the estimation under these challenges. Our pipeline showed a higher performance when landmark uncertainty is provided by deep learning pipelines, *e.g.*, when landmarks are represented with heatmaps or random variables.

An extension of face analysis methods is in applications for enhancing communication often in combination with hand gesture analysis, such as sign language recognition and translation systems. Since sign language is not universal, pipelines for production and interpretation

are language-specific, deterring their application when large public datasets are not available. Moreover, the synthesis of sign images has multiple challenges such as low resolution, blurriness, and scarce datasets with multi-modal data and large signer variability. Our rendering pipeline enables the extension of existing datasets, by providing multiple types of data on a frame basis. Furthermore, the datasets can be rendered with different signers, adding noise to the pose, to simulate increased sign variability.

7.3 Future Work

The proposed approaches for head pose estimation and for pose and expression transfer exploit 2D facial landmarks detected in every frame. These pipelines could leverage the use of lightweight 3DA-2D and 3D face alignment approaches, instead of 2D landmarks, to improve the detection in frames with large head rotations. However, most of the 3DA-2D and 3D face alignment pipelines are based on deep neural networks, which require the use of dedicated hardware such as GPUs. The development of lightweight approaches under limited resources is a research area that has been scarcely investigated. These methods could exploit datasets such as [354], which provides 3D landmark annotations and the respective head pose on video sequences.

In a following work, the proposed head pose estimation methods could be integrated with eye gaze estimation to assess the level of engagement or attentiveness in a given task. To that end, datasets that provide both types of annotations, *e.g.*, AutoPose [355], could be explored. The eye motion could also be incorporated in our performance-driven facial animation pipeline in a future work, as the used model includes eye balls.

The modelling of other head parts such as tongue and hair for avatar animation have gained attention in the last years. The use of these models in settings with limited resources are also an interesting area of research for future works, where the model parameters can be reduced using techniques such as PCA.

As mentioned in Chapter 4, a limitation in existing face analysis pipelines is the robust detection of facial landmarks in uncommon expressions. This problem is a consequence of the limited annotated data in extreme cases, *e.g.*, in patients of unilateral facial paralysis. Since the collection of this data is costly, future work should investigate more semi-supervised or unsupervised models where a latent representation is learned, and it can be exploited for modelling asymmetry.

Furthermore, the proposed face alignment pipeline could be used as an starting point for evaluating the level of paralysis based on the detected landmarks. Current assessing pipelines could exploit the detected landmarks or a new classification method that provides a score based on the latent code or the provided landmarks are areas to explore.

In sign language production, an interesting future line of research is the mapping from written or spoken text to SMPL-X parameters, instead of body poses. This would remove the need of the extra step for keypoints estimation. The use of multi-modal data such as depth or normal maps should also be explored. SynthSL could be exploited for these purposes, as it provides SMPL-X parameters and multiple types of annotations.

Bibliography

[1] D. A. Leopold and G. Rhodes, "A comparative view of face perception." *Journal of Comparative Psychology*, vol. 124, no. 3, p. 233, 2010.

[2] R. Palermo and G. Rhodes, "Are you always on my mind? a review of how face perception and attention interact," *Neuropsychologia*, vol. 45, no. 1, pp. 75–92, 2007.

[3] M. Dewan, M. Murshed, and F. Lin, "Engagement detection in online learning: a review," *Smart Learning Environments*, vol. 6, no. 1, pp. 1–20, 2019.

[4] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," in *ACM Siggraph 2006 Courses*, ser. SIGGRAPH '06. Association for Computing Machinery, 2006, pp. 17–es.

[5] M. Leo, P. Carcagnì, P. L. Mazzeo, P. Spagnolo, D. Cazzato, and C. Distante, "Analysis of facial information for healthcare applications: A survey on computer vision-based approaches," *Information*, vol. 11, no. 3, p. 128, 2020.

[6] I. Spivak, S. Krepych, V. Faifura, and S. Spivak, "Methods and tools of face recognition for the marketing decision making," in *Int. Scientific-Practical Conf. Problems of Infocommunications, Science and Technology (PIC S&T)*. IEEE, 2019, pp. 212–216.

[7] B. Meden, P. Rot, P. Terhörst, N. Damer, A. Kuijper, W. J. Scheirer, A. Ross, P. Peer, and V. Štruc, "Privacy–enhancing face biometrics: A comprehensive survey," *Trans. on Information Forensics and Security*, vol. 16, pp. 4147–4183, 2021.

[8] M. Haghighat and M. Abdel-Mottaleb, "Low resolution face recognition in surveillance systems using discriminant correlation analysis," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2017, pp. 912–917.

[9] J.-A. Kang, M. H. Chun, S. J. Choi, M. C. Chang, and Y. G. Yi, "Effects of mirror therapy using a tablet pc on central facial paresis in stroke patients," *Ann. Rehabil. Med.*, vol. 41, no. 3, pp. 347–353, 2017.

[10] M. Leo, M. Del Coco, P. Carcagni, C. Distante, M. Bernava, G. Pioggia, and G. Palestra, "Automatic emotion recognition in robot-children interaction for asd treatment," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*, December 2015.

[11] F. M. Talaat, "Real-time facial emotion recognition system among children with autism based on deep learning and iot," *Neural Computing and Applications*, vol. 35, no. 17, pp. 12 717–12 728, 2023.

[12] L. A. Kumar, D. K. Renuka, S. L. Rose, I. M. Wartana, and M. Shunmuga Priya, "Deep learning based assistive technology on audio visual speech recognition for hearing impaired," *Int. Journal of Cognitive Computing in Engineering*, vol. 3, pp. 24–30, 2022.

[13] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3D total solution," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 41, no. 1, pp. 78–92, 2017.

[14] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2018, pp. 7784–7793.

[15] J. M. Díaz Barros, F. Garcia, B. Mirbach, and D. Stricker, "Real-time monocular 6-DoF head pose estimation from salient 2D points," in *Int. Conf. on Image Processing (ICIP)*. IEEE, Sept 2017, pp. 121–125.

[16] J. M. Díaz Barros, F. Garcia, B. Mirbach, K. Varanasi, and D. Stricker, "Combined framework for real-time head pose estimation using facial landmark detection and salient feature tracking," in *Proc. of the 13th Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, vol. 5, INSTICC. SciTePress, 2018, pp. 123–133.

[17] J. M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker, "Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation," in *Winter Conf. on Applications of Computer Vision (WACV)*. IEEE, March 2018.

[18] ——, "Real-time head pose estimation by tracking and detection of keypoints and facial landmarks," in *Computer Vision, Imaging and Computer Graphics Theory and Applications: Int. Joint Conf., VISIGRAPP, Revised Selected Papers*. Springer, 2019, pp. 326–349.

[19] J. M. Díaz Barros, V. Golyanik, K. Varanasi, and D. Stricker, "Face it!: A pipeline for real-time performance-driven facial animation," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2019, pp. 2209–2213.

[20] J. M. Díaz Barros, J. Rambach, P. Murthy, and D. Stricker, "In-domain inversion for improved 3D face alignment on asymmetrical expressions," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2024, pp. 1–8.

[21] J. M. Díaz Barros, C.-Y. Wang, D. Stricker, and J. Rambach, "Confidence-aware clustered landmark filtering for hybrid 3D face tracking," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2023, pp. 2945–2949.

[22] J. M. Díaz Barros, C.-Y. Wang, J. Malik, A. Arafa, and D. Stricker, "SynthSL: Expressive humans for sign language image synthesis," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2024, pp. 1–8.

[23] B. Mirbach, F. G. Becerro, and J. M. Díaz Barros, "Method and system for head pose estimation," Jun. 3 2021, uS Patent App. 16/632,689.

[24] Z. Zhang, "A flexible new technique for camera calibration," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, 2000.

[25] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *European Conf. on Computer Vision (ECCV)*. Santa Margherita Ligure, Italy: Springer, May 1992, pp. 321–334.

[26] H. Ackermann and B. Rosenhahn, "Non-rigid self-calibration of a projective camera," in *Asian Conf. on Computer Vision (ACCV)*. Daejeon, Korea: Springer, November 2013, pp. 177–190.

[27] I. E. Sutherland, "Three-dimensional data input by tablet," *Proceedings of the IEEE*, vol. 62, no. 4, pp. 453–461, 1974.

[28] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[29] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[30] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[31] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[32] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[33] D. G. Lowe, "Object recognition from local scale-invariant features," in *Int. Conf. on Computer Vision (ICCV)*, vol. 2. IEEE, 1999, pp. 1150–1157.

[34] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conf. on Computer Vision (ECCV)*. Springer, 2006, pp. 430–443.

[35] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, pp. 105–119, 2010.

[36] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, vol. 81, 1981, pp. 674–679.

[37] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 4, 2000.

[38] B. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int. Journal of Computer Vision (IJCV)*, vol. 13, pp. 331–356, 1994.

[39] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no. 8, pp. 930–943, 2003.

[40] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 2969–2976.

[41] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *Int. Journal of Computer Vision (IJCV)*, vol. 81, no. 2, pp. 155–166, 2009.

[42] G. Terzakis and M. Lourakis, "A consistently fast and globally optimal solution to the perspective-n-point problem," in *European Conf. on Computer Vision (ECCV)*. Springer, 2020, pp. 478–494.

[43] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of Applied Mathematics*, vol. 16, pp. 588–A604, 1944.

[44] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[45] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, no. 82, pp. 35–45, 1960.

[46] F. Rosenblatt *et al.*, *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan books Washington, DC, 1962, vol. 55.

[47] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Transactions on Electronic Computers*, no. 3, pp. 299–307, 1967.

[48] K. Fukushima, "A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, pp. 193–202, 1980.

[49] H. J. Kelley, "Gradient theory of optimal flight paths," *Ars Journal*, vol. 30, no. 10, pp. 947–954, 1960.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Med. Image Comput. Comput. Assist. Interv. (MICCAI)*. Springer, 2015, pp. 234–241.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[53] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *Int. Conf. on Machine Learning (ICML)*. PMLR, 2020, pp. 10 524–10 533.

[54] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[55] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 012–10 022.

[56] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 009–12 019.

[57] H. Bao, L. Dong, F. Wei, W. Wang, N. Yang, X. Liu, Y. Wang, J. Gao, S. Piao, M. Zhou *et al.*, "Unilmv2: Pseudo-masked language models for unified language model pre-training," in *Int. Conf. on Machine Learning (ICML)*. PMLR, 2020, pp. 642–652.

[58] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2021.

[59] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion model," *arXiv preprint arXiv:2209.02646*, 2022.

[60] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *Int. Conf. Learn. Represent. (ICLR)*, 2014.

[61] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 27, 2014.

[62] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Int. Conf. on Machine Learning (ICML)*. PMLR, 2015, pp. 2256–2265.

[63] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[64] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2001, pp. I–511.

[65] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2497–2504.

[66] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[67] D. E. King, "Max-margin object detection," *arXiv preprint arXiv:1502.00046*, 2015.

[68] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[69] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Robust face detection by simple means," in *DAGM 2012 CVAW workshop*, 2012.

[70] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, pp. 1499–1503, 2016.

[71] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Li, "Faceboxes: A cpu real-time face detector with high accuracy," in *International Joint Conference on Biometrics (IJCB)*. IEEE, 2017, pp. 1–9.

[72] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[73] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks)," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2017, pp. 1021–1030.

[74] H. Zhang, Q. Li, and Z. Sun, "Joint voxel and coordinate regression for accurate 3D facial landmark localization," in *Int. Conf. on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2202–2208.

[75] L. Jeni, J. Cohn, and T. Kanade, "Dense 3D face alignment from 2D video for real-time use," *Image and Vision Computing (IMAVIS)*, vol. 58, pp. 13–24, 2017.

[76] D. Crispell and M. Bazik, "Pix2face: Direct 3D face model estimation," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*, 2017, pp. 2512–2518.

[77] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *European Conf. on Computer Vision (ECCV)*, 2018, pp. 534–551.

[78] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3D face reconstruction from a single image via direct volumetric cnn regression," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2017, pp. 1031–1039.

[79] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPR-W)*. IEEE, 2013, pp. 896–903.

[80] ——, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*. IEEE, 2013, pp. 397–403.

[81] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image and Vision Computing (IMAVIS)*, vol. 47, pp. 3–18, 2016.

[82] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen, "The Menpo facial landmark localisation challenge: A step towards the solution," in *Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2017.

[83] J. Deng, A. Roussos, G. Chrysos, E. Ververas, I. Kotsia, J. Shen, and S. Zafeiriou, "The Menpo benchmark for multi-pose 2D and 3D facial landmark localisation and tracking," *Int. Journal of Computer Vision (IJCV)*, vol. 127, no. 6, pp. 599–624, 2019.

[84] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing (IMAVIS)*, vol. 28, no. 5, pp. 807–813, 2010.

[85] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D shape regression for real-time facial animation," *ACM Trans. on Graphics (TOG)*, vol. 32, no. 4, 2013.

[86] A. Jourabloo and X. Liu, "Large-pose face alignment via CNN-based dense 3D model fitting," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 4188–4196.

[87] E. Wood, T. Baltrušaitis, C. Hewitt, M. Johnson, J. Shen, N. Milosavljević, D. Wilde, S. Garbin, T. Sharp, I. Stojiljković *et al.*, "3D face reconstruction with dense landmarks," in *European Conf. on Computer Vision (ECCV)*. Springer, 2022, pp. 160–177.

[88] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Annual conference on Computer graphics and interactive techniques*, 1999, pp. 187–194.

[89] S. Zafeiriou, G. Chrysos, A. Roussos, E. Ververas, J. Deng, and G. Trigeorgis, "The 3D Menpo facial landmark tracking challenge," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*, 2017, pp. 2503–2511.

[90] J. Booth, E. Antonakos, S. Ploumpis, G. Trigeorgis, Y. Panagakis, and S. Zafeiriou, "3D face morphable models "in-the-wild"," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5464–5473.

[91] J. Shen, S. Zafeiriou, G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*. IEEE, 2015, pp. 50–58.

[92] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[93] K. H. An and M. J. Chung, "3D head tracking and pose-robust 2D texture map-based face recognition using a simple ellipsoid model," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2008, pp. 307–312.

[94] S. Choi and D. Kim, "Robust head tracking using 3D ellipsoidal head model in particle filter," *Pattern Recognition*, vol. 41, no. 9, pp. 2901–2915, 2008.

[95] S. Kumano, K. Otsuka, J. Yamato, E. Maeda, and Y. Sato, "Pose-invariant facial expression recognition using variable-intensity templates," *Int. Journal of Computer Vision (IJCV)*, vol. 83, no. 2, pp. 178–194, Jun 2009.

[96] X. Xu and I. A. Kakadiaris, "Joint head pose estimation and face alignment framework using global and local cnn features," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*, vol. 2. IEEE, May 2017, pp. 642–649.

[97] R. Valenti, Z. Yucel, and T. Gevers, "Robustifying eye center localization by head pose cues," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 612–618.

[98] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *Trans. on Image Processing*, vol. 21, no. 2, pp. 802–815, 2012.

[99] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *Trans. on Intelligent Transportation Systems (T-ITS)*, vol. 16, no. 4, pp. 2014–2027, 2015.

[100] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[101] P. Mohr, M. Tatzgern, J. Grubert, D. Schmalstieg, and D. Kalkofen, "Adaptive user perspective rendering for handheld augmented reality," in *Symposium on 3D User Interfaces (3DUI)*. IEEE, 2017, pp. 176–181.

[102] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness," *Trans. on Intelligent Transportation Systems (T-ITS)*, vol. 11, no. 2, pp. 300–311, 2010.

[103] J. Kun, S. Bok-Suk, and K. Reinhard, "Novel backprojection method for monocular head pose estimation," *Int. Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, no. 1, pp. 50–58, 2013.

[104] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 4, pp. 607–626, 2009.

[105] S. Baker, I. , Matthews, J. Xiao, R. Gross, T. Kanade, and T. Ishikawa, "Real-time non-rigid driver head tracking for driver mental state estimation," in *11th World Congress on Intelligent Transportation Systems*, 2004.

[106] S. Y. Cheng, S. Park, and M. M. Trivedi, "Multi-spectral and multi-perspective video arrays for driver body tracking and activity analysis," *Computer Vision and Image Understanding*, vol. 106, no. 2, pp. 245–257, 2007.

[107] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3D face analysis," *Int. Journal of Computer Vision (IJCV)*, vol. 101, no. 3, pp. 437–458, February 2013.

[108] Z. Guo, H. Liu, Q. Wang, and J. Yang, "A fast algorithm face detection and head pose estimation for driver assistant system," in *Int. Conf. on Signal Processing*, vol. 3. IEEE, 2006.

[109] D. J. Tan, F. Tombari, and N. Navab, "Real-time accurate 3D head tracking and pose estimation with consumer RGB-D cameras," *Int. Journal of Computer Vision (IJCV)*, pp. 1–26, 2017.

[110] Y. Zhu and K. Fujimura, "Head pose estimation for driver monitoring," in *Intelligent Vehicles Symposium*. IEEE, 2004, pp. 501–506.

[111] A. Gudi, X. Li, and J. van Gemert, "Efficiency in real-time webcam gaze tracking," in *European Conf. on Computer Vision Workshops (ECCV-W)*. Springer, 2020, pp. 529–543.

[112] A. Larumbe-Bergera, G. Garde, S. Porta, R. Cabeza, and A. Villanueva, "Accurate pupil center detection in off-the-shelf eye tracking systems using convolutional neural networks," *Sensors*, vol. 21, no. 20, p. 6847, 2021.

[113] Y. Feng, N. Goulding-Hotta, A. Khan, H. Reyserhove, and Y. Zhu, "Real-time gaze tracking with event-driven eye segmentation," in *Conf. on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022, pp. 399–408.

[114] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 617–624.

[115] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz, "Robust model-based 3D head pose estimation," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2015, pp. 3649–3657.

[116] C. Papazov, T. K. Marks, and M. Jones, "Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[117] D. Derkach, A. Ruiz, and F. M. Sukno, "Head pose estimation based on 3-D facial landmarks localization and regression," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, May 2017, pp. 820–827.

[118] T. Baltrušaitis, P. Robinson, and L. Morency, "3D constrained local model for rigid and non-rigid facial tracking," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[119] A. Saeed and A. Al-Hamadi, "Boosted human head pose estimation using kinect camera," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2015, pp. 1752–1756.

[120] A. Schwarz, M. Haurilet, M. Martinez, and R. Stiefelhagen, "Driveahead - a large-scale driver head pose dataset," in *Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPR-W)*. IEEE, 2017.

[121] J. S. Jang and T. Kanade, "Robust 3D head tracking by online feature registration," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2008.

[122] J.-S. Jang and T. Kanade, "Robust 3D head tracking by view-based feature point registration," in *People Image Analysis (PIA) Consortium*. Carnegie Mellon University, 2010.

[123] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, and R. Horaud, "Head pose estimation via probabilistic high-dimensional regression," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2015, pp. 4624–4628.

[124] S. Asteriadis, K. Karpouzis, and S. Kollias, "Head pose estimation with one camera, in uncalibrated environments," in *Workshop on Eye Gaze in Intelligent Human Machine Interaction*. ACM, 2010, pp. 55–62.

[125] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. Journal of Computer Vision (IJCV)*, vol. 15, pp. 123–141, 1995.

[126] H. Wang, F. Davoine, V. Lepetit, C. Chaillou, and C. Pan, "3D head tracking via invariant keypoint learning," *Trans. on Circuits and Systems for Video Technology*, vol. 22, no. 8, pp. 1113–1126, 2012.

[127] B. H. Prasad and R. Aravind, "A robust head pose estimation system for uncalibrated monocular videos," in *Indian Conf. on Computer Vision, Graphics and Image Processing*. ACM, 2010, pp. 162–169.

[128] A. Larumbe, M. Ariz, J. J. Bengoechea, R. Segura, R. Cabeza, and A. Villanueva, "Improved strategies for HPE employing learning-by-synthesis approaches," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*, 2017, pp. 1545–1554.

[129] Z.-H. Feng, P. Huber, J. Kittler, W. Christmas, and X.-J. Wu, "Random cascaded-regression copse for robust facial landmark detection," *Signal Processing Letters*, vol. 22, no. 1, pp. 76–80, 2014.

[130] C. Yin and X. Yang, "Real-time head pose estimation for driver assistance system using low-cost on-board computer," in *SIGGRAPH Int. Conf. on Virtual-Reality Continuum and Its Applications in Industry*, vol. 1. ACM, 2016, pp. 43–46.

[131] Y. Wu, C. Gou, and Q. Ji, "Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion," *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[132] C. Gou, Y. Wu, F.-Y. Wang, and Q. Ji, "Coupled cascade regression for simultaneous facial landmark detection and head pose estimation," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2017.

[133] B. Ahn, J. Park, and I. S. Kweon, "Real-time head orientation from a monocular camera using deep neural network," in *Asian Conf. on Computer Vision (ACCV)*. Springer, 2014, pp. 82–96.

[134] L. Zamuner, K. Bailly, and E. Bigorgne, "A pose-adaptive constrained local model for accurate head pose tracking," in *Int. Conf. on Pattern Recognition (ICPR)*. IEEE, 2014, pp. 2525–2530.

[135] S. Tulyakov, R.-L. Vieriu, S. Semeniuta, and N. Sebe, "Robust real-time extreme head pose estimation," in *Int. Conf. on Pattern Recognition (ICPR)*. IEEE, 2014, pp. 2263–2268.

[136] X. Liu, W. Liang, Y. Wang, S. Li, and M. Pei, "3D head pose estimation with convolutional neural network trained on synthetic images," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2016, pp. 1289–1293.

[137] Y. Zhang and C. Kambhamettu, "3D head tracking under partial occlusion," *Pattern Recognition*, vol. 35, no. 7, pp. 1545–1557, 2002.

[138] J. Sung, T. Kanade, and D. Kim, "Pose robust face tracking by combining active appearance models and cylinder head models," *Int. Journal of Computer Vision (IJCV)*, vol. 80, no. 2, pp. 260–274, 2008.

[139] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Int. Journal of Imaging Systems and Technology*, vol. 13, no. 1, pp. 85–94, 2003.

[140] S. Basu, I. Essa, and A. Pentland, "Motion regularization for model-based head tracking," in *Int. Conf. on Pattern Recognition (ICPR)*, vol. 3. IEEE, 1996, pp. 611–616.

[141] Y. Yu, K. A. Funes Mora, and J.-M. Odobez, "Robust and accurate 3D head pose estimation through 3DMM and online head model reconstruction," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, May 2017, pp. 711–718.

[142] R. S. Ghiass, O. Arandjelović, and D. Laurendeau, "Highly accurate and fully automatic head pose estimation from a low quality consumer-level RGB-D sensor," in *Workshop on*

Computational Models of Social Interactions: Human-Computer-Media Communication. ACM, 2015, pp. 25–34.

[143] L. Morency, J. Whitehill, and J. Movellan, "Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2008, pp. 1–8.

[144] J. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting by regularized landmark mean-shift," *Int. Journal of Computer Vision (IJCV)*, vol. 91, no. 2, pp. 200–215, 2011.

[145] N. A. Dodgson, "Variation and extrema of human interpupillary distance," in *Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291. SPIE, 2004, pp. 36–46.

[146] C. C. Gordon, B. Bradtmiller, C. E. Clauser, T. Churchill, J. T. McConville, I. Tebbetts, and R. A. Walker, "Anthropometric survey of u.s. army personnel: Methods and summary statistics," in *Technical report 89-044. Natick MA: U.S. Army Natick Research, Development and Engineering Center.*, 1989.

[147] J. Y. Bouguet, "Pyramidal implementation of the affine Lucas-Kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, pp. 1–10, 2001.

[148] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[149] "Makehuman: Open source tool for making 3D characters," http://www.makehumancommunity.org/, 2017, [Online; accessed 31-May-2018].

[150] K. E. Atkinson, *An introduction to numerical analysis*. John Wiley & Sons, 2008.

[151] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in C: The art of scientific computing," 1992.

[152] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[153] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, 2009.

[154] G. R. C. for Artificial Intelligence (DFKI)., "Head pose estimation dataset," http://av.dfki.de/publications/real-time-head-pose-estimation-by-tracking-and-detection-of-keypoints-and-facial-landmarks/, 2018.

[155] S. Lefevre and J. M. Odobez, "Structure and appearance features for robust 3D facial actions tracking," in *Int. Conf. on Multimedia and Expo*. IEEE, June 2009, pp. 298–301.

[156] N.-T. Tran, F.-E. Ababsa, M. Charbit, J. Feldmar, D. Petrovska-Delacrétaz, and G. Chollet, "3D face pose and animation tracking via eigen-decomposition based bayesian approach," in *Int. Symposium on Visual Computing*. Springer, 2013, pp. 562–571.

[157] C. Huang, Y. Li, C. Chen, and X. Tang, "Deep imbalanced learning for face recognition and attribute prediction," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2019.

[158] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019, pp. 7763–7772.

[159] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," *ACM Trans. on Graphics (TOG)*, vol. 40, no. 4, pp. 1–13, 2021.

[160] E. Wood, T. Baltrušaitis, C. Hewitt, S. Dziadzio, T. Cashman, and J. Shotton, "Fake it till you make it: face analysis in the wild using synthetic data alone," in *Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 3681–3691.

[161] I. Grishchenko, A. Ablavatski, Y. Kartynnik, K. Raveendran, and M. Grundmann, "Attention mesh: High-fidelity face mesh prediction in real-time," *arXiv preprint arXiv:2006.10962*, 2020.

[162] G. J. Hsu, W. Huang, and J. Kang, "Hierarchical network for facial palsy detection." in *Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2018, pp. 580–586.

[163] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2018, pp. 2129–2138.

[164] A. Bandini, S. Rezaei, D. L. Guarin, M. Kulkarni, D. Lim, M. I. Boulos, L. Zinman, Y. Yunusova, and B. Taati, "A new dataset for facial motion analysis in individuals with neurological disorders," *Journal of Biomedical and Health Informatics (JBHI)*, vol. 25, no. 4, pp. 1111–1119, 2020.

[165] D. L. Guarin, Y. Yunusova, B. Taati, J. R. Dusseldorp, S. Mohan, J. Tavares, M. M. van Veen, E. Fortier, T. A. Hadlock, and N. Jowett, "Toward an automatic system for computer-aided assessment in facial palsy," *Facial Plastic Surgery & Aesthetic Medicine (FPSAM)*, vol. 22, pp. 42–49, 2020.

[166] H. S. Kim, S. Y. Kim, Y. H. Kim, and K. S. Park, "A smartphone-based automatic diagnosis system for facial nerve palsy," *Sensors*, vol. 15, no. 10, pp. 26 756–26 768, 2015.

[167] Z. Guo, G. Dan, J. Xiang, J. Wang, W. Yang, H. Ding, O. Deussen, and Y. Zhou, "An unobtrusive computerized assessment framework for unilateral peripheral facial paralysis," *Journal of Biomedical and Health Informatics (JBHI)*, vol. 22, no. 3, pp. 835–841, 2017.

[168] G. Parra-Dominguez, R. Sanchez-Yanez, and C. Garcia-Capulin, "Facial paralysis detection on images using key point analysis," *Applied Sciences*, vol. 11, no. 5, p. 2435, 2021.

[169] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3D dense face alignment," in *European Conf. on Computer Vision (ECCV)*. Springer, 2020, pp. 152–168.

[170] H. Jin, S. Liao, and L. Shao, "Pixel-in-pixel net: Towards efficient facial landmark detection in the wild," *Int. Journal of Computer Vision (IJCV)*, vol. 129, no. 12, pp. 3174–3194, 2021.

[171] B. Browatzki and C. Wallraven, "3FabRec: Fast few-shot face alignment by reconstruction," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 6110–6120.

[172] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *Int. Journal of Computer Vision (IJCV)*, vol. 127, pp. 115–142, 2019.

[173] G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou, "A comprehensive performance evaluation of deformable face tracking 'in-the-wild'," *Int. Journal of Computer Vision (IJCV)*, vol. 126, no. 2-4, pp. 198–232, 2018.

[174] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 6, pp. 681–685, 2001.

[175] F. Kahraman, M. Gokmen, S. Darkner, and R. Larsen, "An active illumination and appearance (AIA) model for face alignment," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–7.

[176] J. Saragih and R. Goecke, "A nonlinear discriminative approach to aam fitting," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2007, pp. 1–8.

[177] T. Cootes and C. Taylor, "Active shape models – 'smart snakes'," in *Brit. Machine Vision Conf. (BMVC)*. BMVA Press, 1992, pp. 266–275.

[178] ——, "Active shape model search using local grey-level models: A quantitative evaluation." in *Brit. Machine Vision Conf. (BMVC)*, vol. 93. BMVA Press, 1993, pp. 639–648.

[179] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *European Conf. on Computer Vision (ECCV)*. Springer, 2008, pp. 504–513.

[180] X. Yu, J. Huang, S. Zhang, W. Yan, and D. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2013, pp. 1944–1951.

[181] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Incremental face alignment in the wild," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1859–1866.

[182] G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3659–3667.

[183] S. Zhu, C. Li, C. Change Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 4998–5006.

[184] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPR-W)*. IEEE, 2017, pp. 88–97.

[185] A. Dapogny, K. Bailly, and M. Cord, "Decafa: Deep convolutional cascade for face alignment in the wild," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2019, pp. 6893–6901.

[186] H. Li, Z. Guo, S. Rhee, S. Han, and J. Han, "Towards accurate facial landmark detection via cascaded transformers," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 4176–4185.

[187] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Style aggregated network for facial landmark detection," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 379–388.

[188] X. Zou, S. Zhong, L. Yan, X. Zhao, J. Zhou, and Y. Wu, "Learning robust facial landmark detection via hierarchical structured ensemble," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2019, pp. 141–150.

[189] S. Qian, K. Sun, W. Wu, C. Qian, and J. Jia, "Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2019, pp. 10 153–10 163.

[190] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:1904.04514*, 2019.

[191] A. Kumar, T. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng, "Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 8236–8246.

[192] A. Bulat, E. Sanchez, and G. Tzimiropoulos, "Subpixel heatmap regression for facial landmark localization," in *Brit. Machine Vision Conf. (BMVC)*. BMVA Press, 2021.

[193] B. Yu and D. Tao, "Heatmap regression via randomized rounding," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2021.

[194] P. Micaelli, A. Vahdat, H. Yin, J. Kautz, and P. Molchanov, "Recurrence without recurrence: Stable video landmark detection with deep equilibrium models," in *CVPR*. IEEE/CVF, 2023, pp. 22 814–22 825.

[195] L. Gu and T. Kanade, "3D alignment of face in a single image," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2006, pp. 1305–1312.

[196] F. Liu, D. Zeng, Q. Zhao, and X. Liu, "Joint face alignment and 3D face reconstruction," in *European Conf. on Computer Vision (ECCV)*. Springer, 2016, pp. 545–560.

[197] F. Liu, Q. Zhao, D. Zeng *et al.*, "Joint face alignment and 3D face reconstruction with application to face recognition," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2018.

[198] J. Piao, C. Qian, and H. Li, "Semi-supervised monocular 3D face reconstruction with end-to-end shape-preserved domain transfer," in *Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 9398–9407.

[199] X. Tu, J. Zhao, M. Xie, Z. Jiang, A. Balamurugan, Y. Luo, Y. Zhao, L. He, Z. Ma, and J. Feng, "3D face reconstruction from a single image assisted by 2D face images in the wild," *Trans. on Multimedia*, vol. 23, pp. 1160–1172, 2021.

[200] Z. Ruan, C. Zou, L. Wu, G. Wu, and L. Wang, "Sadrnet: Self-aligned dual face regression networks for robust 3D dense face alignment and reconstruction," *Trans. on Image Processing*, 2021.

[201] C. Wu, Q. Xu, and U. Neumann, "Synergy between 3DMM and 3D landmarks for accurate 3D facial geometry," in *Int. Conf. on 3D Vision (3DV)*. IEEE, 2021, pp. 453–463.

[202] Y. Zhou, J. Deng, I. Kotsia, and S. Zafeiriou, "Dense 3D face decoding over 2500fps: Joint texture & shape convolutional mesh decoders," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 1097–1106.

[203] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 532–539.

[204] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides, "Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2017, pp. 3980–3989.

[205] A. Zadeh, Y. Chong Lim, T. Baltrusaitis, and L. Morency, "Convolutional experts constrained local model for 3D facial landmark detection," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*. IEEE, 2017, pp. 2519–2528.

[206] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile gpus," *arXiv preprint arXiv:1907.06724*, 2019.

[207] M. Dornier, P. Gosselin, C. Raymond, Y. Ricquebourg, and B. Coüasnon, "Stylegan-based heatmap generator for face alignment with limited training data," *hal03778322v2*, 2023.

[208] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 8110–8119.

[209] L. Tran and X. Liu, "On learning 3D face morphable model from in-the-wild images," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2019.

[210] P. Chandran, G. Zoss, P. Gotardo, and D. Bradley, "Continuous landmark detection with 3D queries," in *CVPR*. IEEE/CVF, 2023, pp. 16 858–16 867.

[211] L. Zeng, L. Chen, W. Bao, Z. Li, Y. Xu, J. Yuan, and N. K. Kalantari, "3D-aware facial landmark detection via multi-view consistent training on synthetic data," in *CVPR*. IEEE/CVF, 2023, pp. 12 747–12 758.

[212] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[213] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for MobileNetv3," in *CVPR*, 2019, pp. 1314–1324.

[214] C. Jiang, J. Wu, W. Zhong, M. Wei, J. Tong, H. Yu, and L. Wang, "Automatic facial paralysis assessment via computational image analysis," *Journal of Healthcare Engineering*, vol. 2020, 2020.

[215] X. Liu, Y. Xia, H. Yu, J. Dong, M. Jian, and T. D. Pham, "Region based parallel hierarchy convolutional neural network for automatic facial nerve paralysis evaluation," *Trans. on Neural Systems and Rehabilitation Engineering (TNSRE)*, vol. 28, no. 10, pp. 2325–2332, 2020.

[216] T. H. Ngo, M. Seo, N. Matsushiro, W. Xiong, and Y.-W. Chen, "Quantitative analysis of facial paralysis based on limited-orientation modified circular gabor filters," in *Int. Conf. on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 349–354.

[217] G. Barrios Dell'Olio and M. Sra, "Farapy: An augmented reality feedback system for facial paralysis using action unit intensity estimation," in *ACM Symposium on User Interface Software and Technology*, 2021, pp. 1027–1038.

[218] X. Ge, J. M. Jose, P. Wang, A. Iyer, X. Liu, and H. Han, "ALGRNet: Multi-relational adaptive facial action unit modelling for face representation and relevant recognitions," *Trans. on Biometrics, Behavior, and Identity Science*, vol. 5, no. 4, pp. 566–578, 2023.

[219] A. Taufique, A. Savakis, and J. Leckenby, "Automatic quantification of facial asymmetry using facial landmarks," in *Western New York Image and Signal Processing Workshop (WNYISPW)*. IEEE, 2019, pp. 1–5.

[220] G. J. Hsu, J. Kang, and W. Huang, "Deep hierarchical network with line segment learning for quantitative analysis of facial palsy," *IEEE Access*, vol. 7, pp. 4833–4842, 2018.

[221] B. Hontanilla and D. Marré, "Comparison of hemihypoglossal nerve versus masseteric nerve transpositions in the rehabilitation of short-term facial paralysis using the facial clima evaluating system," *Plastic and Reconstructive Surgery*, vol. 130, no. 5, pp. 662e–672e, 2012.

[222] Y. Zhao, G. Feng, H. Wu, S. Aodeng, X. Tian, G. F. Volk, O. Guntinas-Lichius, and Z. Gao, "Prognostic value of a three-dimensional dynamic quantitative analysis system to measure facial motion in acute facial paralysis patients," *Head & face medicine*, vol. 16, no. 1, pp. 1–10, 2020.

[223] A. Gaber, M. F. Taher, and M. A. Wahed, "Quantifying facial paralysis using the kinect v2," in *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2015, pp. 2497–2501.

[224] J. J. Greene, D. L. Guarin, J. Tavares, E. Fortier, M. Robinson, J. Dusseldorp, O. Quatela, N. Jowett, and T. Hadlock, "The spectrum of facial palsy: The MEEI facial palsy photo and video standard set," *The Laryngoscope*, vol. 130, pp. 32–37, 2020.

[225] J. Delannoy and T. Ward, "A preliminary investigation into the use of machine vision techniques for automating facial paralysis rehabilitation therapy," in *Irish Signals and Systems Conference (ISSC)*. IET, 2010.

[226] H. Yoshihara, M. Seo, T. Ngo, N. Matsushiro, and Y. Chen, "Automatic feature point detection using deep convolutional networks for quantitative evaluation of facial paralysis," in *Int. Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2016, pp. 811–814.

[227] T. Wang, J. Dong, X. Sun, S. Zhang, and S. Wang, "Automatic recognition of facial movement for paralyzed face," *Bio-medical materials and engineering*, vol. 24, no. 6, pp. 2751–2760, 2014.

[228] J. Barbosa, W. Seo, and J. Kang, "parafacetest: an ensemble of regression tree-based facial features extraction for efficient facial paralysis classification," *BMC Medical Imaging*, vol. 19, no. 1, pp. 1–14, 2019.

[229] Y. Liu, Z. Xu, L. Ding, J. Jia, and X. Wu, "Automatic assessment of facial paralysis based on facial landmarks," in *Int. Conf. on on Pattern Recognition and Machine Learning (PRML)*. IEEE, 2021, pp. 162–167.

[230] D. Guarin, J. Dusseldorp, T. Hadlock, and N. Jowett, "A machine learning approach for automated facial measurements in facial palsy," *JAMA facial plastic surgery*, vol. 20, no. 4, pp. 335–337, 2018.

[231] Z. Guo, W. Li, J. Dai, J. Xiang, and G. Dan, "Facial imaging and landmark detection technique for objective assessment of unilateral peripheral facial paralysis," *Enterprise Information Systems*, pp. 1–17, 2021.

[232] X. Xiong and F. De la Torre, "Global supervised descent method," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 2664–2673.

[233] P. Xiong, G. Li, and Y. Sun, "Combining local and global features for 3D face tracking," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*. IEEE, 2017, pp. 2529–2536.

[234] J. Deng, Y. Zhou, S. Cheng, and S. Zaferiou, "Cascade multi-view hourglass model for robust 3D face alignment," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2018, pp. 399–403.

[235] D. Gordon, A. Farhadi, and D. Fox, "Re^3: Real-time recurrent regression networks for visual tracking of generic objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 788–795, 2018.

[236] D. Aspandi, O. Martinez, F. Sukno, and X. Binefa, "Fully end-to-end composite recurrent convolution network for deformable facial tracking in the wild," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2019, pp. 1–8.

[237] U. Prabhu, K. Seshadri, and M. Savvides, "Automatic facial landmark tracking in video sequences using kalman filter assisted active shape models," in *European Conf. on Computer Vision (ECCV)*. Springer, 2010, pp. 86–99.

[238] Y. Jin, X. Guo, Y. Li, J. Xing, and H. Tian, "Towards stabilizing facial landmark detection and tracking via hierarchical filtering: A new method," *Journal of the Franklin Institute*, vol. 357, no. 5, pp. 3019–3037, 2020.

[239] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conf. on Computer Vision (ECCV)*. Springer, 2016, pp. 694–711.

[240] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Int. Conf. Learn. Represent. (ICLR)*, 2015.

[241] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[242] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *European Conf. on Computer Vision (ECCV)*. Springer, 2020, pp. 592–608.

[243] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2019, pp. 6971–6981.

[244] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Int. Conf. on Computer Vision (ICCV)*. IEEE, 2015, pp. 3730–3738.

[245] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Trans. Affect. Comput.*, pp. 18–31, 2017.

[246] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019.

[247] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2879–2886.

[248] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. Huang, "Interactive facial feature localization," in *European Conf. on Computer Vision (ECCV)*. Springer, 2012, pp. 679–692.

[249] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 12, pp. 2930–2940, 2013.

[250] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J. Jacobsen, "Invertible residual networks," in *Int. Conf. on Machine Learning (ICML)*. PMLR, 2019, pp. 573–582.

[251] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.

[252] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learn. Represent. (ICLR)*, 2014.

[253] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2018, pp. 67–74.

[254] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," in *ACM Trans. on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011.

[255] S. Bouaziz, Y. Wang, and M. Pauly, "Online modeling for realtime facial animation," *ACM Trans. on Graphics (TOG)*, vol. 32, no. 4, 2013.

[256] P.-L. Hsieh, C. Ma, J. Yu, and H. Li, "Unconstrained realtime facial performance capture," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[257] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3D face reconstruction, tracking, and applications," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018.

[258] T. Weise, H. Li, L. Van Gool, and M. Pauly, "Face/off: Live facial puppetry," in *SIGGRAPH/Eurographics Symposium on Computer animation*. ACM, 2009, pp. 7–16.

[259] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives." *ACM Trans. on Graphics (TOG)*, vol. 32, 2013.

[260] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment." *ACM Trans. on Graphics (TOG)*, vol. 34, no. 6, 2015.

[261] R. Blanco i Ribera, E. Zell, J. Lewis, J. Noh, and M. Botsch, "Facial retargeting with automatic range of motion alignment," *ACM Trans. on Graphics (TOG)*, vol. 36, no. 4, 2017.

[262] Y. Guo, J. Zhang, L. Cai, J. Cai, and J. Zheng, "Self-supervised cnn for unconstrained 3D facial performance capture from a single RGB-D camera," *arXiv preprint:1808.05323*, 2018.

[263] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. on Graphics (TOG)*, vol. 33, no. 4, 2014.

[264] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," *IEEE Trans. on Visualization and Computer Graphics*, vol. 20, no. 3, 2014.

[265] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt, "Reconstruction of personalized 3D face rigs from monocular video," *ACM Trans. on Graphics (TOG)*, vol. 35, no. 3, June 2016.

[266] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[267] S. Liu, X. Yang, Z. Wang, Z. Xiao, and J. Zhang, "Real-time facial expression transfer with single video camera," *Computer Animation and Virtual Worlds*, vol. 27, 2016.

[268] S. Saito, T. Li, and H. Li, "Real-time facial segmentation and performance capture from RGB input," in *European Conf. on Computer Vision (ECCV)*. Springer, 2016.

[269] "Use animoji on your iphone x and ipad pro," https://support.apple.com/en-gb/HT208190, 2018, [Online; accessed 01-Feb-2019].

[270] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. H. Pighin, and Z. Deng, "Practice and theory of blendshape facial models." *Eurographics (State of the Art Reports)*, vol. 1, no. 8, 2014.

[271] S. Tickoo, *Autodesk Maya 2018: A Comprehensive guide*. CADCIM Techonologies, 2017.

[272] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. on Graphics (TOG)*, vol. 36, no. 6, 2017.

[273] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *European Conf. on Computer Vision (ECCV)*. Springer, 2018.

[274] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*. OUP Oxford, 2004, vol. 30.

[275] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.

[276] ——, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, no. 5, pp. 827–828, 1978.

[277] T. F. Chan and L. Vese, "Active contours without edges," *Trans. on Image Processing*, vol. 10, 2001.

[278] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[279] D. Cosker, E. Krumhuber, and A. Hilton, "A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling," in *Int. Conf. on Computer Vision (ICCV)*, 2011.

[280] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott, "Tessa, a system to aid communication with deaf people," in *Int. ACM Conf. on Assistive Technologies*. New York, USA: ACM Press, 2002, pp. 205–212.

[281] D. Wood, A. Griffiths, H. Wood, and I. Howarth, *Teaching and talking with deaf children*. New York, USA: Wiley, 1986, vol. 10.

[282] J. G. Kyle, J. Kyle, B. Woll, G. Pullen, and F. Maddix, *Sign language: The study of deaf people and their language*. Cambridge, UK: Cambridge University Press, 1988.

[283] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Winter Conf. on Applications of Computer Vision (WACV)*. Piscataway, NJ: IEEE, 2020, pp. 1459–1469.

[284] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater, and H. Ney, "RWTH-PHOENIX-Weather: A large vocabulary sign language recognition and translation corpus." in *Int. Conf. on Language Resources and Evaluation (LREC)*, vol. 9. Istanbul, Turkey: European Language Resources Association (ELRA), 2012, pp. 3785–3789.

[285] A. Duarte, S. Palaskar, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i Nieto, "How2sign: A large-scale multimodal dataset for continuous american sign language," in *Sign Language Recognition, Translation and Production (SLRTP) Workshop-Extended Abstracts*. Cham, Switzerland: Springer, 2020, pp. 1–4.

[286] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE. Piscataway, NJ: IEEE, 2019, pp. 10 975–10 985.

[287] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2017, pp. 109–117.

[288] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney, "Speech recognition techniques for a sign language recognition system," in *Annual Conf. of the Int. Speech Communication Association*. France: ISCA Archive, 2007, pp. 2513–2516.

[289] H. R. Vaezi Joze and O. Koller, "MS-ASL: A large-scale data set and benchmark for understanding american sign language," in *Brit. Machine Vision Conf. (BMVC)*, British Machine Vision Association. Durham, UK: BMVA Press, 2019, pp. 1–16.

[290] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time sign language letter and word recognition from depth data," in *Int. Conf. on Computer Vision Workshops (ICCV-W)*, IEEE. Piscataway, NJ: IEEE, 2011, pp. 383–390.

[291] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in *Int. Conf. on Multimodal Interfaces*. New York, USA: ACM, 2011, pp. 279–286.

[292] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. b. Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," *Sensors*, vol. 18, no. 7, p. 2208, 2018.

[293] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2020, pp. 10 023–10 033.

[294] A. Kratimenos, G. Pavlakos, and P. Maragos, "3D hands, face and body extraction for sign language recognition," in *Sign Language Recognition, Translation and Production (SLRTP) Workshop-Extended Abstracts*, vol. 4. Cham, Switzerland: Springer, 2020, pp. 1–4.

[295] J. Rodriguez, J. Chacon, E. Rangel, L. Guayacan, C. Hernandez, L. Hernandez, and F. Martinez, "Understanding motion in sign language: A new structured translation dataset," in *Asian Conference on Computer Vision (ACCV)*. Cham, Switzerland: Springer, 2020, pp. 1–16.

[296] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou, and P. Daras, "A comprehensive study on sign language recognition methods," *arXiv preprint arXiv:2007.12530*, vol. 1, pp. 1–12, 2020.

[297] S. Gibet, N. Courty, K. Duarte, and T. L. Naour, "The signcom system for data-driven animation of interactive virtual signers: Methodology and evaluation," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 1, no. 1, pp. 1–23, 2011.

[298] S. Gibet, F. Lefebvre-Albaret, L. Hamon, R. Brun, and A. Turki, "Interactive editing in french sign language dedicated to virtual signers: requirements and challenges," *Universal Access in the Information Society*, vol. 15, no. 4, pp. 525–539, 2016.

[299] J. Glauert, R. Elliott, S. Cox, J. Tryggvason, and M. Sheard, "Vanessa–a system for communication between deaf and hearing people," *Technology and Disability*, vol. 18, no. 4, pp. 207–216, 2006.

[300] K. Karpouzis, G. Caridakis, S.-E. Fotinea, and E. Efthimiou, "Educational resources and implementation of a greek sign language synthesis architecture," *Computers & Education*, vol. 49, no. 1, pp. 54–74, 2007.

[301] J. McDonald, R. Wolfe, J. Schnepp, J. Hochgesang, D. G. Jamrozik, M. Stumbo, L. Berke, M. Bialek, and F. Thomas, "An automated technique for real-time production of lifelike animations of american sign language," *Universal Access in the Information Society*, vol. 15, no. 4, pp. 551–566, 2016.

[302] S. Stoll, N. C. Camgöz, S. Hadfield, and R. Bowden, "Sign language production using neural machine translation and generative adversarial networks," in *Brit. Machine Vision Conf. (BMVC)*, British Machine Vision Association. Durham, UK: BMVA Press, 2018, pp. 1–12.

[303] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, "Text2sign: towards sign language production using neural machine translation and generative adversarial networks," *Int. Journal of Computer Vision (IJCV)*, vol. 128, p. 891–908, 2020.

[304] S. Stoll, S. Hadfield, and R. Bowden, "Signsynth: Data-driven sign language video generation," in *8th International Workshop on Assistive Computer Vision and Robotics*. Cham, Switzerland: Springer, 2020, pp. 1–17.

[305] J. Zelinka and J. Kanis, "Neural sign language synthesis: Words are our glosses," in *The IEEE Winter Conference on Applications of Computer Vision*. Piscataway, NJ: IEEE, 2020, pp. 3395–3403.

[306] B. Saunders, N. C. Camgoz, and R. Bowden, "Progressive transformers for end-to-end sign language production," in *European Conf. on Computer Vision (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 687–705.

[307] ——, "Everybody sign now: Translating spoken language to photo realistic sign language video," *arXiv preprint arXiv:2011.09846*, vol. 1, pp. 1–11, 2020.

[308] B. Saunders, N. C. Camgöz, and R. Bowden, "Adversarial training for multi-channel sign language production," in *British Machine Vision Virtual Conference*, British Machine Vision Association. Durham, UK: BMVA Press, 2020, pp. 1–15.

[309] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[310] L. Ventura, A. Duarte, and X. Giro-i Nieto, "Can everybody sign now? exploring sign language video generation from 2D poses," in *Sign Language Recognition, Translation and Production (SLRTP) Workshop - Extended Abstracts*. Cham, Switzerland: Springer, 2020, pp. 1–4.

[311] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Int. Conf. on Computer Vision (ICCV)*. Piscataway, NJ: IEEE, 2019, pp. 5933–5942.

[312] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8798–8807.

[313] B. Saunders, N. C. Camgoz, and R. Bowden, "Signing at scale: Learning to co-articulate signs for large-scale photo-realistic sign language production," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5141–5151.

[314] S. Fang, C. Sui, X. Zhang, and Y. Tian, "Signdiff: Learning diffusion models for american sign language production," *arXiv preprint arXiv:2308.16082*, 2023.

[315] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney, "Modeling image variability in appearance-based gesture recognition," in *ECCV Workshop on Statistical Methods in Multi-image and Video Processing*. Cham, Switzerland: Springer, 2006, pp. 7–18.

[316] X. Chai, H. Wang, and X. Chen, "The devisign large vocabulary of chinese sign language database and baseline evaluations," Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Tech. Rep. VIPL-TR-14-SLR-001, 2014.

[317] S. Ebling, N. C. Camgöz, P. Boyes Braem, N. Calzolari *et al.*, "Smile swiss german sign language dataset," in *Int. Conf. on Language Resources and Evaluation (LREC)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018, pp. 4221–4229.

[318] O. Özdemir, A. A. Kındıroğlu, N. C. Camgöz, and L. Akarun, "Bosphorussign22k sign language recognition dataset," *9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, vol. 1, pp. 1–8, 2020.

[319] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-Weather." in *Int. Conf. on*

Language Resources and Evaluation (LREC). Reykjavik, Iceland: European Language Resources Association, 2014, pp. 1911–1916.

[320] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *AAAI Conference on Artificial Intelligence*, IAAAI. Palo Alto, California: AAAI Press, 2018, pp. 2257–2264.

[321] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," *Applied Sciences*, vol. 9, no. 13, p. 2683, 2019.

[322] A. C. Duarte, "Cross-modal neural sign language translation," in *ACM Int. Conf. on Multimedia*. New York, USA: ACM, 2019, pp. 1650–1654.

[323] N. C. Camgöz, B. Saunders, G. Rochette, M. Giovanelli, G. Inches, R. Nachtrab-Ribback, and R. Bowden, "Content4all open research sign language translation datasets," in *Int. Conf. on Automatic Face and Gesture Recognition (FG)*. IEEE, 2021, pp. 1–5.

[324] S. Albanie, G. Varol, L. Momeni, H. Bull, T. Afouras, H. Chowdhury, N. Fox, B. Woll, R. Cooper, A. McParland, and A. Zisserman, "BOBSL: BBC-Oxford British Sign Language Dataset," in *arXiv preprint arXiv:2111.03635*, 2021.

[325] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2017, pp. 7291–7299.

[326] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2016, pp. 3793–3802.

[327] M. Alaghband, N. Yousefi, and I. Garibay, "Facial expression phoenix (feph): An annotated sequenced dataset for facial and emotion-specified expressions in sign language," *Image*, vol. 20, p. 27, 2020.

[328] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Int. Conf. on Computer Vision and Pattern Recognition Workshop (CVPR-W)*. Piscataway, NJ: IEEE, 2018, pp. 969–977.

[329] Y. Chen, W. Li, X. Chen, and L. V. Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2019, pp. 1841–1850.

[330] S. Bak, P. Carr, and J.-F. Lalonde, "Domain adaptation through synthesis for unsupervised person re-identification," in *European Conf. on Computer Vision (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 189–205.

[331] M. F. Ghezelghieh, R. Kasturi, and S. Sarkar, "Learning camera viewpoint using cnn to improve 3D body pose estimation," in *Int. Conf. on 3D vision (3DV)*. Piscataway, NJ: IEEE, 2016, pp. 685–693.

[332] J. Liu, H. Rahmani, N. Akhtar, and A. Mian, "Learning human pose models from synthesized data for robust rgb-d action recognition," *Int. Journal of Computer Vision (IJCV)*, vol. 127, no. 10, pp. 1545–1564, 2019.

[333] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," *ACM Trans. on Graphics (TOG)*, vol. 24, no. 3, pp. 408–416, 2005.

[334] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. on Graphics (TOG)*, vol. 34, no. 6, pp. 248:1–248:16, Oct 2015.

[335] G. Varol, I. Laptev, C. Schmid, and A. Zisserman, "Synthetic humans for action recognition from unseen viewpoints," *Int. Journal of Computer Vision (IJCV)*, vol. 129, pp. 2264––2287, 2021.

[336] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2019, pp. 11 807–11 816.

[337] A. Ranjan, J. Romero, and M. J. Black, "Learning human optical flow," in *Brit. Machine Vision Conf. (BMVC)*, British Machine Vision Association. Durham, UK: BMVA Press, sep 2018, pp. 1–13.

[338] H. M. Clever, Z. Erickson, A. Kapusta, G. Turk, K. Liu, and C. C. Kemp, "Bodies at rest: 3D human pose and shape estimation from a pressure image using synthetic data," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2020, pp. 6215–6224.

[339] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. on Graphics (TOG)*, vol. 36, no. 6, p. 245, 2017.

[340] A. Ranjan, D. T. Hoffmann, D. Tzionas, S. Tang, J. Romero, and M. J. Black, "Learning multi-human optical flow," *Int. Journal of Computer Vision (IJCV)*, vol. 128, pp. 873–890, January 2020.

[341] D. T. Hoffmann, D. Tzionas, M. J. Black, and S. Tang, "Learning to train with synthetic humans," in *German Conference on Pattern Recognition (GCPR)*. Cham, Switzerland: Springer, Sep 2019, pp. 609–623.

[342] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," *arXiv preprint arXiv:1807.09384*, vol. 1, pp. 1–10, 2018.

[343] M. J. Black, P. Patel, J. Tesch, and J. Yang, "BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 8726–8737.

[344] Z. Cai, W. Yin, A. Zeng, C. Wei, Q. Sun, Y. Wang, H. E. Pang, H. Mei, M. Zhang, L. Zhang, C. C. Loy, L. Yang, and Z. Liu, "Smpler-x: Scaling up expressive human pose and shape estimation," *Adv. Neural Inform. Process. Syst. (NeurIPS)*, vol. 1, 2023.

[345] P. Patel, C.-H. P. Huang, J. Tesch, D. T. Hoffmann, S. Tripathi, and M. J. Black, "AGORA: Avatars in geography optimized for regression analysis," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021.

[346] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[347] B. O. Community, "Blender - a 3D modelling and rendering package," http://www.blender.org, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[348] Autodesk, INC., "Maya," https:/autodesk.com/maya.

[349] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.

[350] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for perceiving and processing reality," in *Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2019.

[351] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[352] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 2794–2802.

[353] J. H. Lim and J. C. Ye, "Geometric GAN," *arXiv preprint arXiv:1705.02894*, 2017.

[354] M. Ariz, J. J. Bengoechea, A. Villanueva, and R. Cabeza, "A novel 2D/3D database with automatic face annotation for head tracking and pose estimation," *Computer Vision and Image Understanding*, vol. 148, pp. 201–210, 2016.

[355] M. Selim, A. Firintepe, A. Pagani, and D. Stricker, "AutoPOSE: Large-scale automotive driver head pose and gaze dataset with deep head orientation baseline," in *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, 2020.

List of Abbreviations

Abbreviation	Definition
3DA-2D	3D Aware 2D Landmarks
3DMM	3D Morphable Model
AAM	Active Appearance Model
ASM	Active Shape Model
AE	Autoencoder
AUC	Area Under the Curve
AUs	Action Units
AW	Adaptive Wing Loss
BB	Bounding box
BL	Bottom left
BR	Bottom right
BU	Boston University
CED	Cumulative Error Distribution
CHM	Cylindrical Head Model
CLM	Constrained Local Model
CNN	Convolutional Neural Network
CRM	Cascaded Regression Method
DEM	Dynamic 3D Expression Model
DLT	Direct Linear Transform
DoF	Degrees of Freedom
DVF	Distance Vector Fields
DNN	Deep Neural Networks
DTW	Dynamic Time Warping
EHM	Ellipsoidal Head Model
EKF	Extended Kalman Filter
ESQ	Extended SuperQuadric
FAST	Features from Accelerated Segment Test
FC	Fully-Connected Layer
FID	Fréchet Inception Distance
FL	Facial Landmarks
FPS	Frames per Second
FR	Failure Rate
FT	Encoder Fine-Tuning
GAN	Generative Adversarial Network
GAVAM	Generalized Adaptive View-based Appearance Model
GMAC	Giga Multiply-Accumulate Operation

Abbreviation	Definition
GTE	Ground Truth Error
HMM	Hidden Markov Model
HPE	Head Pose Estimation
HOG	Histogram of Oriented Gradients
HG	Hourglass
IO	Inter-Ocular
ITL	Interleaved Transfer Layers
Inv	In-domain Inversion
IR	Infrared
KF	Kalman Filter
KL	Kullback–Leibler divergence
KLT	Kanade Lucas Tomasi
KP	Keypoints
LLS	Linear Least Squares
LN	Layer Normalization
LSTM	Long Short Term Memory
MMOD	Max-Margin Object Detection
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MSA	Multi-Head Self Attention
MSE	Mean Squared Error
NFC	Normal Flow Constraint
NLLS	Non-linear Least Squares
NLP	Natural Language Processing
NME	Normalized Mean Error
OAM	Online Appearance Model
PAM	Parameterized Appearance Models
PCA	Principal Component Analysis
PE	Position Errors
POSIT	Pose from Orthography and Scaling by Iterations
PnP	Perspective-n-Point
PSNR	Peak Signal-to-Noise Ratio
RANSAC	Random Sample Consensus
ReLU	Rectified Linear Units
RNN	Recurrent Neural Network
ROI	Region of Interest
RMSE	Root Mean Square Errors
SDM	Supervised Descent Method
SF	Salient Features

Abbreviation	Definition
SIFT	Scale Invariant Feature Transform
SL	Sign Language
SLP	Sign Language Production
SOTA	State of the Art
SSIM	Structural Similarity Index Measure
STD	Standard Deviation
SVD	Singular Value Decomposition
SVM	Support Vector Machines
SW-MSA	Shifted Windows Multi-Head Attention
TL	Top left
TR	Top right
TSP	Triangular Surface Patch
VAE	Variational Autoencoder
ViT	Vision Transformer
W-MSA	Windows Multi-Head Attention

List of Tables

1.1 Challenging images from AFLW2000-3D [13] and ParFace. 2
3.1 Evaluation on the BU benchmark with uniform illumination. 61
3.2 Evaluation on the BU benchmark with varying illumination. 62
3.3 Errors on the translation estimation on BU benchmark with uniform illumination. . 62
3.4 Errors on the translation estimation on BU benchmark with varying illumination. . 62
3.5 RMSE of the head orientation based on the chosen weight threshold on the weighting scheme. 62
3.6 HPE performance with and without facial landmark refinement with triangulation. 64
3.7 Evaluation of HPE from K.P., F.L. and fusion approach on challenging sequence. . 64
3.8 Averaged runtime of keypoints-based method for 150 runs. 68
3.9 Averaged runtime of HPE by measurement fusion for 100 runs. 68
3.10 Averaged runtime of HPE by fusion of global-local motion for 100 runs. 68
4.1 Publicly available datasets used for training our architecture. 82
4.2 NME (3DA-2D) and GTE (3D) on AFLW2000-3D, for different yaw angles. . . . 85
4.3 Performance metrics on ParFace. 86
4.4 Ablation study on AFLW2000-3D and ParFace. 89
4.5 Ablation study on 3DA-2D and 3D face alignment. 90
4.6 Performance metrics after fine-tuning with training sets from ParFace. 91
4.7 Comparison with 3FabRec on 2D face alignment. 92
4.8 Ablation study on 2D face alignment. 92
4.9 Ablation study on 2D face alignment after training the self-supervised stage with portions of the data. 93
4.10 Performance of different Kalman Filter configurations on the training set of Menpo 3DA-2D benchmark and ParFace. 98
4.11 Performance of face detection activation mechanism vs using the face detector with FAN on Menpo 3DA-2D benchmark. 100
4.12 RMSE, Area under the curve and failure rate (%) on Menpo 3DA2D tracking test set. 102
4.13 RMSE, Area under the curve, failure rate (%) and GTE on ParFace. 102
4.14 Area under the curve and Failure Rate on Menpo 3DA2D tracking benchmark. . . 102
5.1 Averaged runtime of the proposed performance-driven animation pipeline. . . . 113
5.2 Comparison of rotational MAE on the Boston University dataset. 117
6.1 Large-Scale Word and Sentence-level Sign Language Datasets. 124
6.2 Quantitative evaluation of different generators on Phoenix 2014T 134
6.3 Quantitative evaluation of different generators on SynthSL 134
6.4 Quantitative evaluation of different generators on BosphorusSign22k 134
6.5 Quantitative evaluation of different SwinGenerator architectures on SynthSL. . . 135
6.6 Quantitative evaluation with different sizes of Swin V2 on SynthSL. 135
6.7 Quantitative evaluation of different adversarial loss functions on SynthSL. 135

6.8 Quantitative evaluation on Phoenix 2014T and BosphorusSign22k with pre-trained models. 136
6.9 Runtime analysis for image synthesis. 136

List of Figures

2.1 Pinhole camera model. . . . 11
2.2 Triangulation. . . . 14
2.3 Feature detection and tracking. . . . 16
2.4 Kalman Filter diagram. . . . 17
2.5 U-Net architecture. . . . 20
2.6 ResNet block 21
2.7 ResNet-34 architecture. . . . 21
2.8 Transformer architecture and Multi-head attention. . . . 22
2.9 Vision Transformer (ViT) architecture. . . . 23
2.10 Swin-Transformer architecture (Swin-T). . . . 24
2.11 Shifted windows between adjacent layers. . . . 25
2.12 Windows padding using cyclic shift. . . . 25
2.13 Swin Transformers V1 vs. V2. . . . 26
2.14 Variational Autoencoder (VAE) architecture. . . . 27
2.15 Generative Adversarial Network (GAN) architecture. . . . 27
2.16 Diffusion model architecture. . . . 28
2.17 Facial landmarks aligned to a detected face. . . . 31
2.18 Predefined set of landmarks in the literature. . . . 31
3.1 Head pose orientation. . . . 38
3.2 Proposed HPE pipeline based on keypoints. . . . 42
3.3 Location of the CHM/EHM in the 3D space. . . . 43
3.4 Initialization of the CHM to fit the user's head. . . . 44
3.5 Ellipsoidal head model. . . . 45
3.6 Keypoints detected using FAST-ER and distance transform on given ROI. . . . 45
3.7 Computation of 3D keypoints. . . . 46
3.8 ROI computation for 2D keypoint detection. . . . 48
3.9 Pipeline of the proposed HPE approach. . . . 50
3.10 Selected facial landmarks in 2D. . . . 50
3.11 Proposed HPE pipeline, from Keypoints (blue) and Facial Landmarks (red). . . . 55
3.12 Selected facial landmarks in 3D head model. . . . 56
3.13 Head pose estimation on challenging sequence. . . . 65
3.14 Rotation error on challenging sequence. . . . 66
4.1 Face alignment on patient with palsy. . . . 72
4.2 Face alignment on asymmetrical expressions. . . . 78
4.3 Architecture of ParFace-Net (PF-Net). . . . 79
4.4 Samples images of ParFace. . . . 81
4.5 3DA-2D and 3D Face alignment on AFLW2000-3D. . . . 84
4.6 CED curves for 3DA-2D and 3D face alignment on ParFace. . . . 86
4.7 Face alignment on ParFace. . . . 87

4.8 3D face alignment on ParFace. . . . 87
4.9 3DA-2D landmarks detected with and without in-domain inversion (IDI). . . . 89
4.10 Failure cases. . . . 94
4.11 Landmark grouping in Kalman Filters. . . . 96
4.12 CED curve for FAN with and without Kalman Filter, with face detector or activation mechanism. . . . 99
4.13 CED curve with and without Kalman Filter on the Menpo 3DA2D test set. . . . 101
4.14 CED curve for face alignment alignment methods with and without Kalman Filter on ParFace. . . . 101
4.15 CED curve on Menpo 3DA2D tracking benchmark. . . . 103
5.1 Overview of the proposed pipeline for 2D-3D facial expression transfer. . . . 108
5.2 Registration of blendshape targets from a reference to a target model. . . . 110
5.3 Our function to limit the blendshape target weights. . . . 112
5.4 Identity set from the COMA dataset [273]. . . . 113
5.5 Blendshape targets. . . . 114
5.6 Energy minimization results under different parameters. . . . 115
5.7 Runtime of the energy minimization for varying ρ and ϵ. . . . 115
5.8 Results of our performance-driven facial animation approach. . . . 116
6.1 Sign language representation overview. . . . 121
6.2 Rendering pipeline for generating synthetic data. . . . 126
6.3 Selection of signers from SynthSL. . . . 127
6.4 Selected frames from SynthSL. . . . 127
6.5 Proposed framework for the synthesis of sign language images. . . . 129
6.6 Architecture of the proposed SwinGenerator. . . . 130
6.7 Synthesized sign images on SynthSL. . . . 132
6.8 Synthesized sign images on Phoenix 2014T. . . . 133
6.9 Synthesized sign images on BosphorusSign22k. . . . 133
6.10 Failure cases. . . . 137

Curriculum Vitae

Jilliam María Díaz Barros

Education

2012 - 2014 — **Université de Bourgogne**
Le Creusot, France
M.Sc. in Computer Science and Electronics

2005 - 2010 — **Universidad del Norte**
Barranquilla, Colombia
Bachelor in Electronics Engineering

Professional experience

2020 - Present — **ANNA Technologies GmbH & ANNA Healthcare Saarland UG**
Kaiserslautern, Germany
Software engineer

2019 - Present — **German Research Center for Artificial Intelligence, DFKI**
Kaiserslautern, Germany
Research assistant and Researcher, Augmented Vision Department

2015 - 2018 — **IEE S.A.**
Contern, Luxembourg
PhD research trainee

2014 — **SnT, University of Luxembourg & Laboratoire Le2i, Université de Bourgogne**
Luxembourg & France
Research intern

2013 — **CNRS, Laboratoire Le2i, Laboratoire ARTHEIS & Laboratoire ICube**
Le Creusot, France
Research intern

2011 - 2012 — **Universidad del Norte**
Barranquilla, Colombia
Electrical and Electronics Engineering Laboratories Coordinator

Publication list

1. Jilliam M. Díaz Barros, F. Garcia, B. Mirbach and D. Stricker, "Real-Time Monocular 6-DOF Head Pose Estimation From Salient 2D Points", in IEEE International Conference on Image Processing (ICIP), Beijing, China, pp. 121-125. 2017.

2. Jilliam M. Díaz Barros, F. Garcia, B. Mirbach, K. Varanasi, and D. Stricker. "Combined Framework for Real-time Head Pose Estimation using Facial Landmark Detection and Salient Feature Tracking", in International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Madeira, Portugal, pp. 123-133. 2018.

3. Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker. "Fusion of Keypoint Tracking and Facial Landmark Detection for Real-Time Head Pose Estimation", in IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, pp. 2028-2037. 2018.

4. Jilliam M. Díaz Barros, B. Mirbach, F. Garcia, K. Varanasi, D. Stricker, "Real-Time Head Pose Estimation by Tracking and Detection of Keypoints and Facial Landmarks", in Computer Vision, Imaging and Computer Graphics Theory and Applications. VISIGRAPP. Communications in Computer and Information Science (CCIS), vol 997. Springer, Cham. 2018.

5. Jilliam M. Díaz Barros, V. Golyanik, K. Varanasi and D. Stricker, "Face It!: A Pipeline for Real-Time Performance-Driven Facial Animation", in IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 2209-2213. 2019.

6. Jilliam M. Díaz Barros, J. Rambach, P. Murthy and D. Stricker, "In-Domain Inversion for Improved 3D Face Alignment on Asymmetrical Expressions", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024.

7. Jilliam M. Díaz Barros, C-Y. Wang, D. Stricker and J. Rambach, "Confidence-Aware Clustered Landmark Filtering for Hybrid 3D Face Tracking", in IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia. 2023.

8. Jilliam M. Díaz Barros, C-Y. Wang, J. Malik, A. Arafa and D. Stricker, "SynthSL: Expressive Humans for Sign Language Image Synthesis", in IEEE International Conference on Automatic Face and Gesture Recognition (FG), Istanbul, Turkey. 2024.