Reconstructing Facial Expressions of HMD Users for Avatars in VR

Christian Felix Purps[1(✉)], Simon Janzer[1], and Matthias Wölfel[1,2]

[1] Faculty of Computer Science and Business Information Systems
Karlsruhe University of Applied Sciences, Karlsruhe, Germany
christian_felix.purps@h-ka.de.com

[2] Faculty of Business, Economics and Social Sciences, Stuttgart, Germany

Abstract. Real-time recognition of human facial expressions and their transfer and use in software is now established and can be found in a variety of computer applications. Most solutions, however, do not focus on facial recognition to be used in combination with wearing a head-mounted display. In these cases, the face is partially obscured, and approaches that assume a fully visible face are not applicable. To overcome this limitation, we present a systematic approach that covers the entire pipeline from facial expression recognition using RGB images to real-time facial animation of avatars based on blendshapes for virtual reality applications. To achieve this, we (a) developed a three-stage machine learning pipeline to recognize mouth areas, extract anthropological landmarks, and detect facial muscle activations and (b) created a realistic avatar using photogrammetry, 3D modeling, and applied blendshapes that closely follow the facial action coding system (FACS). This provides an interface to our facial expression recognition system, but also allows other blendshape-oriented approaches to work with our avatar. Our facial expression recognition system performed well on common metrics and under real-time testing. Jitter and the detection or approximation of upper face facial features, however, are still an issue that needs to be addressed.

Keywords: Facial expressions · Avatars · HMD · Virtual reality

1 Introduction

Human facial expressions are a crucial aspect of nonverbal communication and a way of displaying emotions that are continuously interpreted by interlocutors [1]. As communication becomes increasingly computer-mediated for a variety of reasons (e.g., COVID-19, reduced traveling time), methods to overcome the limitations of 2D video conferencing are required. For instance, such systems cannot provide eye contact or transfer proxemic information. VR-mediated communication relying on 3D scans of the participants (e.g., as point clouds or voxels)

M. Wölfel et al. (Eds.): ArtsIT 2021, LNICST 422, pp. 61–76, 2022.
https://doi.org/10.1007/978-3-030-95531-1_5

or represented as animated avatars is able to overcome the aforementioned limitations. Both forms of representation should be able to convey human social signals of any kind in the virtual world in the same way that real people otherwise do in reality. In current applications, however, such as social VR platforms that provide a shared virtual space where people are represented by avatars or 3D scans, this goal is far from being achieved due to current limitations [2,3].

One reason for this is the lack of visualization of realistic facial expressions on an avatar's face in VR, which has several causes that could lead to inaccurate results and might cause the character to fall into the uncanny valley [2]. A key problem here is that current imaging techniques cannot correctly sense the upper face because it is occluded by the head-mounted display (HMD). Another problem lies in retargeting or mapping issues, as well as the individuality of faces, which can also greatly affect the quality of avatar facial animation.

Although several approaches to this challenge have recently emerged, RGB(D)-based real-time facial reconstruction for HMD users has not yet been made available to a wide audience. There are several solutions that address the problems on a component-by-component basis, but these components are usually not combined into a holistic system and often require complicated individual hardware installations. Hence, we present an approach that covers the entire pipeline from face tracking to sophisticated avatar face rendering and animation in HMD VR.

2 Related Work

Numerous commercial solutions[1,2,3,4] and scientific approaches [4–6] have been presented for capturing human body signals for various applications (e.g., avatar-mediated communication) including the animation of an avatars face. However, most of these approaches are not suitable for VR applications where large parts of the face are occluded by an HMD, which prevents the use of common methods, pre-trained neural networks, and commercial solutions. Moreover, these mostly machine learning-based algorithms cannot be applied only to the lower face, as they require full face recognition. Looking at the few existing sophisticated approaches that address this problem, it is clear that facial expression recognition still plays a niche role for HMD users. Only a few solutions address this challenge in general and most of them are limited to the recognition of expressions of the lower half of the face.

An early approach to facial expression tracking with retargeting in avatars comes from Wei et al. in which facial features are recognized and a 3D face model is animated using a dynamic inference algorithm and a transformation of facial motion parameters into facial animation parameters [5]. Although their

[1] https://developer.apple.com/augmented-reality/arkit/.
[2] https://developers.google.com/ar/discover.
[3] https://www.banuba.com/.
[4] https://visagetechnologies.com/facetrack/.

approach is quite old and does not address VR challenges (e.g., partially occluded facial parts), the two main system components remain basically the same: real-time image-based tracking of facial features and real-time generation of facial expressions on a 3D facial model of the avatar.

Brito and Mitchell propose a method for reusing landmark datasets for real-time face detection of HMD users and avatar animations [7]. Their method separates a given dataset into local regions for eyes and mouth, and then uses different machine learning approaches for landmark extraction. Although this system provides robust face tracking, facial regions that cannot be tracked by optical systems remain unaddressed.

Hickson et al. developed a system for classifying facial expressions in VR using eye-tracking cameras only. They used a CNN to classify 5 emotions and 10 facial action units. Their system achieved an F1 score of 0.73 for emotion and 0.68 for facial action unit detection [8]. However, their approach does not show the level of muscle activation for the facial action units because their data only included the presence or absence of activation of a specific muscle group and therefore is not suitable for continuous avatar facial animations.

Just recently, HTC released the VIVE Facial Tracker[5], one of the few commercial solutions capable of capturing facial expressions from the lower half of the face, providing an easy-to-use application for real-time animation of avatar faces. The solution uses RGBD images and is calibrated for use with a VIVE device, but also works with devices from other manufacturers. Approximate eyebrow tracking is technically possible with the build-in camera for eye-tracking (e.g., of a VIVE Pro Eye HMD), but achieves poor results in reality. Therefore, VIVE face tracking remains a solution to provide lower facial expressions.

Lou et al. presented one of the few solutions to fully reconstruct realistic facial expressions for VR HMD users [9]. They attached electromyography (EMG) sensors to the frame of an HMD, tracked muscle movements, and then used preprocessed EMG signals to reconstruct facial action units of the covered facial regions. Common imaging techniques were used to track the lower face. Their system achieved decent results by accurately assigning facial expressions to an avatar. However, the system has two major drawbacks: it requires additional hardware on each HMD frame and it cannot achieve real-time performance, which prevents its applicability for avatar-mediated communication (AMC).

A closer look at the few existing approaches reveals that there is still a need for research and development in this area to find a straightforward, real-time, and robust solution for rendering facial expressions in HMD VR.

3 System Architecture

To overcome the challenges mentioned in related work, we developed a holistic system that controls every aspect of the processing pipeline from the initial raw

[5] https://www.vive.com/de/accessory/facial-tracker/.

image acquisition until the animation of the 3D face mesh (see Fig. 1). Our main goal was to implement a solution through a simplified approach that does not require complicated or costly additional hardware. Therefore, we decided to use an ordinary RGB webcam and to keep our system flexible by loose coupling of the individual components. Therefore, the approach can be simply adjusted to different use cases and with different hardware setups. In general, our solution can be divided into two basic components as previously proposed by Wei et al. [5]. The first component (Sect. 4) is responsible for real-time recognition of facial expressions and their conversion into muscle activation values. The second component (Sect. 5) involves the creation of a rigged avatar model whose facial expressions are adjustable by morphing the facial geometry using blendshapes. The two components are connected via an interface adapted from the facial expression recognition system (FACS) standard and thus allowing other components to be compatible (e.g. the VIVE Facial Tracker).

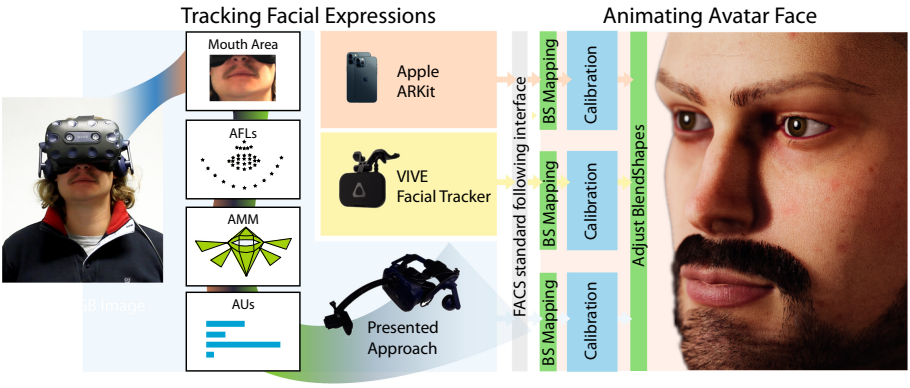

Fig. 1. Sketched architecture of the presented system. Adapted from the FACS standard an interface enables to animate our avatars face through BlendShape activation based on data coming not only from our own solution, but also from different input devices.

4 Sensing Lower Facial Expressions

Image processing and pattern recognition are typical approaches for detecting and tracking facial features. In our approach, we use a standard RGB webcam attached to the HMD with an action cam attachment. The captured RGB image stream is then processed in different stages (mouth detection, anthropological face landmark extraction, active appearance model calculation, and action unit regression, see Fig. 1) using machine learning approaches to enable facial animation of avatars.

All computations and algorithm training described in this section, as well as the live test application, interfaces, etc., were implemented using Python 3.6, with openCV 3.4.2 for image processing and rendering, numpy 1.14.2 for matrix computations, and dlib 18.18, scikit-learn 0.20 and keras 2.2.2 for artificial neural network model development and training.

Fig. 2. HTC Vive Pro Eye with a common RGB-Webcam attached

4.1 Datasets Used

For our supervised learning-based approach to avatar facial animations for HMD users, we required multiple datasets including faces, labeled with anthropological face landmarks (AFL), FACS, and classified emotions. All the required datasets came in very different file structures, formats, and labeling styles, which meant we had to homogenize the datasets in a first pre-processing step.

Anthropological Face Landmarks are a basic notation for describing facial features in terms of x,y coordinates of relevant feature points of a face. There exist different standards with different numbers of landmarks (detail levels) that can be used to describe facial features. A common notation provides 68 landmarks and is labeled in datasets such as iBug W-300 [10] or Kaggel's Facial Keypoints (68), which we merged and used as training data.

Facial Action Units are an integral part of the FACS, that describes activation of the different facial muscle groups and is commonly used as a basic notation for labeled facial expression datasets and an implementation of avatar facial animation with minor modifications (e.g. OpenFACS) [11,12]. There are few dataset that have human portrait pictures labeled with action unit (AU) activation, such as FERA [13], DISFA [14], and FEAFA [4]. However, FERA and DISFA lack information about AUs that could be symmetrically distinguished (e.g., left/right mouth corner). In addition, AU intensity is only provided in five discrete levels which made FEAFA the favored dataset for our consideration.

Emotions can be described as a combination of the simultaneous activation of AUs. Dataset that include pictured labeled with the matching emotion are e.g. the CK+ [15] and the FACES [16] dataset. The FACES database provides a wide age range of faces with different labeled expressions, making it well suited for studying developmental and other research questions about emotions. We decided to use the FACES dataset because of its variety in individuals and better picture quality.

Although emotion recognition is not yet implemented in the current system, it is still worth mentioning that the avatar we designed already meets all the requirements for easy implementation in future work.

4.2 Augmentation of Training Data

Supervised training of a machine learning algorithm based on RGB data requires datasets of training images. To train the different algorithms, different datasets (see Table 2) were needed for the different processing stages (especially Stage 1 and Stage 2) and data augmentation. Since the learning algorithms are to be used to recognize faces or parts of faces that are partially occluded by an HMD, these HMDs must also already be included in the training images for this use case (this is not required for development/debugging purposes where no HMD is set up). Although the parts of the face that are not occluded should be detected, the recognition of the mouth, for example, is more difficult if parts of the HMD are still visible at the edge of this facial region. Thus, we used an approach similar to Suresh et al. who used a dataset (Face Mask Detection Dataset[6]) where face masks have been automatically added to non-masked faces and then jointly recognized and classified [17]. We then automatically added an overlay image showing a VIVE Pro HMD to all images from the target datasets (300-W, Kaggle FK68), and then placed, scaled, and rotated it based on the given 68 landmark notations (see Fig. 3). An excerpt from the augmentation result is depicted in Fig. 4.

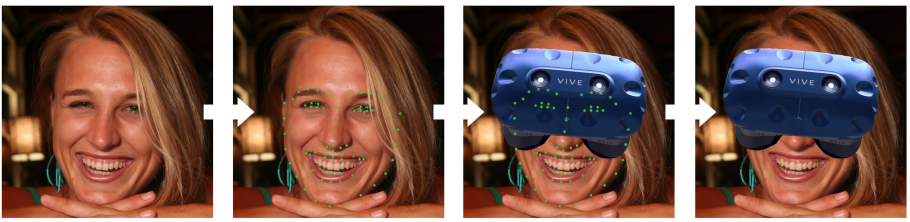

Fig. 3. Augmentation process steps from raw training image to HMD augmented picture based on 68 landmark coding

[6] https://www.kaggle.com/omkargurav/face-mask-dataset.

4.3 Mouth Detection

The recognition of the mouth area is required for all further processing and calculation steps. In particular, the accuracy of facial feature extraction (see Sect. 4.4) is highly influenced by the precision of mouth detection. To recognize the mouth within RGB images, we chose a common approach using a convolutional neural network (CNN). First introduced by Simonyan and Zisserman, the VGG16 network architecture has been very successful in large-scale image and video recognition [18]. Since it has also proven successful in similar tasks for estimating regions in images [19], we have adopted it in a slightly different configuration. We flattened the last output layer of the VGG16 network and added three dense layers and a four-unit output layer describing the normalized coordinates of the bounding box of the bottom of the face and mouth, respectively (see Fig. 5). We trained the network (VGG16 layers freezed) based on 10.300 of our preprocessed annotated images derived from the 300-W and FK68 datasets with a train/test split of 70% training, 20% validation and 10% test data using mean squared error (MSE) as loss function and adaptive moment estimation (ADAM) optimization.

Fig. 4. Excerpt of training images augmented with HMD

4.4 Facial Landmark Extraction

Object detection based on the histogram of gradients (HoG) is a popular computer vision method for detecting semi-rigid objects and has gained acceptance in the field of face analysis. Dlib's solution, which integrates an HoG detector, can represent both appearance and shape information [20]. Dlib also offers easy implementation of predicting facial landmarks using an ensemble of regression

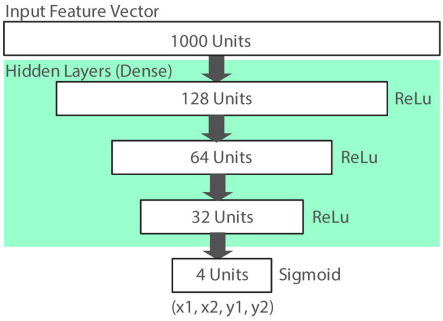

Fig. 5. Structure and parameters auf the network used for bounding box prediction of the lower facial area

trees (ERT) in an image and provides real-time high-quality predictions that perform better than other approaches, such as CNNs. Therefore, we used its shape predictor, which takes an image region containing an object and outputs a set of point positions that define the pose of the object, in our case the AFLs of the mouth, the bottom jaw, and the tip of the nose. Before training our shape predictor, we first applied our trained mouth detection algorithm to our concatenated and preprocessed training set (10,300 images) to identify the mouth region.

Moreover, it was required to crop the 68 AFLs (representing the entire face) to represent only the mouth region. For this reason, we reduced the landmarks to the lower part of the jaw (landmarks 3–14), the lower part of the nose (landmarks 30–36), and the mouth (landmarks 48–68), resulting in a total of 37 AFLs.

We then trained the shape predictor (tree depth: 5, cascade depth: 25, feature pool size: 400, oversampling amount: 5, jitter correction: 0.1) with the cropped images from the datasets 300-W and FK68.

4.5 Face Muscle Activation

There have been several approaches to animate avatar faces that estimate facial muscle activation [4,21,22]. Facial muscle activation can be normalized and retargeted to a 3D model to modify its blendshapes for facial morphological changes. Based on the detected landmarks (see Sect. 4.4), we created an active appearance model (AMM) [23]. The approach utilizing an AMM is varying from the original approach of Yan et al. who have created the FEAFA-A dataset for avatar facial animation. However, as the dataset includes faces of Asian ethnicity only it was required to find a more abstract representation to be used with faces of other ethnicities before training. Thus, we used an AMM consisting of eight polygons to describe the lower face region and its facial features (Fig. 6). Using this AMM for AU detection required us to process the FEAFA-A dataset by applying our mouth detection and facial landmark extraction algorithm on every image. Then, the new dataset used for training was created by calculating the values describ-

ing polygons of the AMM for each image. To train the algorithm, these polygons are represented as a flattened input feature vector of size 86 containing the normalized vectors representing each polygon. To measure facial muscle activation we used a different deep learning approach. For this approach, we had created our own AU mapping based on the FACS standard with slight modifications (see Table 1) as output. In addition, we had to subdivide some of the FACS AUs into separate units because it is required to be able to detect asymmetric movements of the mouth (e.g., left/right lip corners). As a network model, we used a simple 3-dense-layer fully connected artificial neural network with 86 input units (flattened polygon vectors) and 14 output units (AU activation predictions).

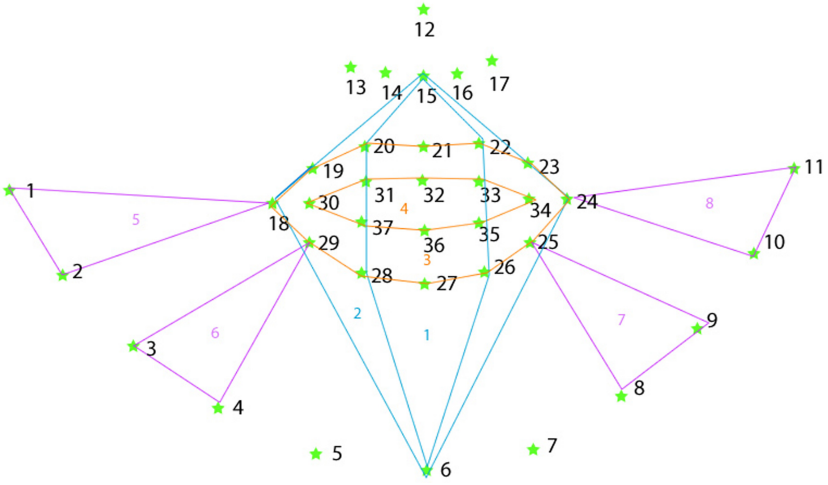

Fig. 6. Our AMM consisting of 8 polygons (43 2-dimensional vectors) created based on 37 AFLs recognized as basis for AU recognition algorithm training. The AMM was developed experimentally according to observational experience with regard on the anatomy of the facial musculature.

We trained our network based on 99,300 training examples from the preprocessed FEAFA-A dataset in a train/test split of 70% training, 20% validation and 10% test data using mean squared error (MSE) as loss function and adaptive moment estimation (ADAM) optimization.

5 Avatar Animation

Talking avatars using various facial animation techniques are ubiquitous in various media [6]. Approaches such as physics-based facial modeling and animation promise sophisticated results by taking into account potential energies and physical interaction of passive flesh, active muscles, rigid bone structure, etc., which

Table 1. Recognized action units of the lower face and the corresponding FACS AUs respectively ADs. Additionally, our AU2 - AU9 subdivide the original FACS into two distinct AUs.

AU	Our definition	Original FACS	AU	Our definition	Original FACS
1	Jaw drop	AU26 jaw drop	8	Upper lip suck	AU28 lip suck
2	Jaw slide left	AD30 jaw sideways	9	Lower lip suck	AU28 lip suck
3	Jaw slide right	AD30 jaw sideways	10	Jaw thrust	AD29 jaw thrust
4	Left lip corner pull	AU12 lip corner P.	11	Upper lip raise	AU10 upper lip raiser
5	Right lip corner pull	AU12 lip corner P.	12	Lower lip depress	AU16 lower lip Dep.
6	Left lip corner Str.	AU20 lip stretcher	13	Chin raise	AU17 chin raiser
7	Right lip corner Str.	AU20 lip stretcher	14	Lip pucker	AU18 lip pucker

offers the potential to compensate for the unnatural morphological deformations caused by HMDs [24]. However, the blendshape-based animation approach has become the most popular and is currently the leading approach for realistic facial animation [25]. This approach requires the development of a facial rig model for a 3D avatar, which is time-intensive but also provides great control over the various facial animations. For the creation of blendshapes, the FACS also provides a good standard and guidance. Even though the FACS are not equivalent to the required blendshapes for facial animation, a general reference, a standardized interface, and decoupling to the detection techniques are provided. Therefore, our approach to avatar facial animation allows the use of the methodology described in Sect. 4, but can also be used in conjunction with other, e.g., commercial systems (Fig. 1).

There are a variety of commercial rigged avatars, but our approach has specific rigging/blendshaps and interface requirements, so we created a new avatar from scratch. We used various tools and techniques for sophisticated avatar creation and implemented our final avatar including logic and interface in Unreal Engine 4.26 for rendering, applying, testing, and created a heads-up display (HUD) for monitoring and control.

5.1 Face Modeling

To create a realistic avatar face for self-representation, 3D scanning methods are crucial to achieve the most realistic results. For our face scan, we used Reality Capture[7], one of the leading photogrammetry tools that can quickly and easily create a 3D model and export both textures and polypaint. We then created a character base mesh, unwrapped the character model, and created UV maps. To match the base mesh with the created scan model, we projected the character base mesh onto the scan in Zbrush[8]. Then we adjusted the base mesh, subdivided it, and projected it again. This procedure was done until the base mesh was

[7] https://www.capturingreality.com/.
[8] http://pixologic.com/features/about-zbrush.php.

identical to the scan, or until all the details we wanted to project onto the base mesh were transferred. The resulting character base mesh consists of 32k vertices, of which almost 12k are for the head area, which is in line with today's general game engine recommendation for a character with a poly count of about 10k–100k. In this step, we also transferred the polypaint to the base mesh, which could later be used to create a colormap texture. For texturing, we used skin materials (physical-based rendering) based on the Digital Human Materials offered by Epic Games. For the creation of hair, we used an approach that adapts the method introduced by d'Eon et al., a reflection model for dielectric cylinders that has high fidelity for rough surfaces such as human hair fibers [26].

5.2 Rigging/Blendshapes

Since our approach is oriented on FACS-based blendshape, we started with Facit-BlendShapes as a basis. However, only a few of the base models BlendShapes are used for further development. The majority had to be created completely from scratch by hand. The character control rig was created using the Blender addon Auto-Rig-Pro. All weight painting had to be done by hand to achieve good deformation results. The face area itself has no weight painting, as all face movements are controlled by the BlendShapes. For detailed facial features additional shape keys have been introduced to the avatar model. The bones created for the FACS emotions are only visual bones that have no influence on the mesh itself and only activate the corresponding BlendShapes via drivers.

5.3 Calibration

Since each face to be tracked has its individual characteristics, more accurate results can be provided by calibrating the blendshape model for each user. Therefore, we have provided various modifiers to adjust the blendshape weights as well as maximum and extreme value constraints. Depending on the input device for the blendshape control, remap curves, size curves, or just manual adjustments of the float values can be used for calibration and fine-tuning.

6 Results

Under the composition of the two main components, we were able to test the overall system and its substructures individually and also the interplay in real-time. The recognition part can be quantified in numbers as well as being evaluated in real-time tests and observations. Table 2 shows the accuracy of the trained algorithms with the respective metrics. The trained network in Stage 1 for recognition of the lower face area achieved an intersection over union (IoU) of 0.91. The shape predictor trained in Stage 2 achieved an MAE of 0.52. Training of the face muscle recognition algorithm in Stage 3 resulted in an MSE of 0.015 for the network whereas AU3 Jaw Slide Right was most precise (MSE: 0.009) and AU14 Lip Pucker was most unprecise (MSE: 0.02).

Table 2. Machine learning pipeline, key figures and results

	Stage 1	Stage 2	Stage 3
Purpose	Mouth recognition	Landmark extraction	Face muscle activation
Dataset	300-W[a], Kaggle FK68[b]	300-W, Kaggle FK68	FEAFA-A[c]
Algorithm	VGG16 + BBR	HOG + ERT	FC Neural network
Input	RGB-Image	RGB-Image	AMM
Output	RGB-Image	AFLs	FACS AUs
Metrics	IoU	MAE	MSE
Result	0.91	0.52	0.015

[a] https://ibug.doc.ic.ac.uk/resources/300-W/
[b] https://www.kaggle.com/tarunkr/facial-keypoints-68-dataset
[c] https://www.iiplab.net/feafa/

The results of the avatar creation (Fig. 7), rendered in real-time by the Unreal Engine show a high degree of natural fidelity. Figure 8 shows the interaction of the tracking (incl. rendering the lower face bounding-box, AFLs, and AMM) and the avatar components and thus the corresponding facial expression of the avatar with that of the tracked person face.

Fig. 7. "In game" screenshots of our photogrammetry scan based and rigged avatar model rendered with Unreal Engine 4.26.

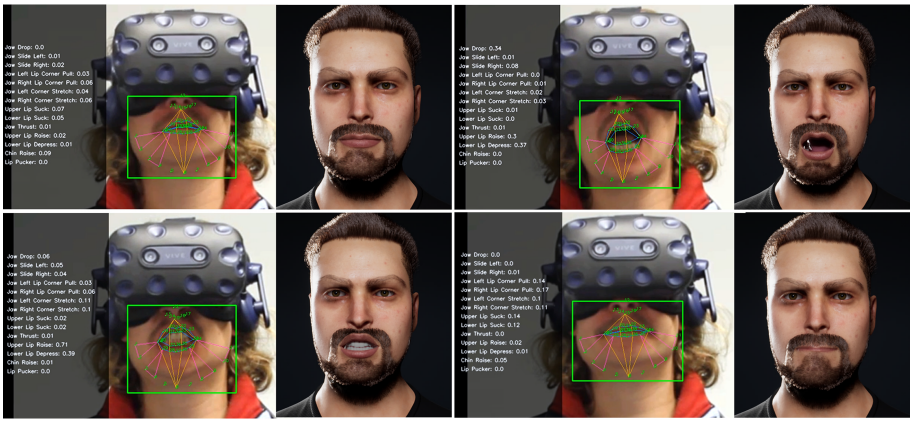

Fig. 8. Examples for FACS AUs activation (Real-time experts) of the lower face. For each of the 4 divisions it is showing: (Left) The computed AU activation as a result of our machine learning pipeling. (Middle) Real-time facial scan displaying mouth- and AFL detection and AMM. (Right) The resulting animated avatar facial expression. Considering the AU activation numbers, it is visible, that the neutral facial expression (top-left) shows almost no activation (all values close to 0). In the top-right picture the "jaw drop", "upper lip raise" and "lower lip depress" AUs are activated. The angry facial expression (bottom-left) causes only the "upper lip raise" and "lower lip depress" AUs to be activated while the bottom-right picture shows slight activation of the interacting AUs "lip corner pull/stretch", "upper lip raise" and "lower lip depress".

7 Conclusion

In this paper, we suggested another approach to visualize authentic facial expressions to be used in combination with wearing an HMD. We established a three-stage process (mouth detection, anthropological face landmark extraction, and action unit prediction) using artificial neural networks and machine learning. We created an avatar based on photogrammetry data that offers blendshape animation that has been created following the FACS standard and can be thus animated by the predicted AU values coming from our last stage of the trained neural networks in real-time via an interface. An application containing our avatar model was created using the Unreal Engine, which is loosely coupled and thus can receive data from different hardware (our own or third-party) to animate the avatar's face in real-time according to the tracked individual facial expressions.

Although the concept works in its entirety, there are some limitations. Error accumulation across the three stages of facial expression recognition often leads to a significant jitter effect and thus a loss of quality. Here, it is important to keep in mind that our AMM was created based on trials and thus was created relatively arbitrarily. Also, no hyperparameter optimization for the networks was included what could further reduce the jitter effect. Also, the implementation of a jitter correction (e.g. Kalman filter) could show improvement here.

Furthermore, it has to be considered that the FEAFA-A data set contains only Asian faces which may cause a significant bias for faces of other ethnicities. A general comparison of our approach compared with commercial ones shows still major quality differences, however, it is to mention that our model is purely based RGB data in contrast to others that also use a depth channel. Furthermore, we must state that the current version of the Unreal Engine has a known bug considering our applied hair rendering technique in VR, which is why other low-quality hair rendering techniques have to be used in this context.

However, although there are still challenges to address, key figures and the final testing results show that the concept of our approach generally works successfully and is worth being further developed.

8 Future Work

Improving the robustness and accuracy of our machine learning pipeline for facial expression recognition is a high priority in our future work, as all further progress depends on it. Thus, hyperparameter-optimization and AMM adjustments, as well as jitter reduction, have to be addressed. Furthermore, we want to take animation of the upper facial expressions into account as we already met all preconditions for emotion detection based on the mouth area. According to Blais et al. the mouth area is the most important cue for both dynamic and static facial expressions [27]. Guarnera et al. compared the ability to recognize emotions from the eye and mouth area in children and adults. Their data shows, that some basic emotions (disgust, happiness, surprise, and neutral) can be decoded just by having information about the mouth area while other emotions (anger, sadness, fear) require information about the eyes [28]. An approach to emotion recognition using imaging techniques and restricting the information to the mouth area was made by Biondi et al. [29]. They trained a CNN to classify happiness, disgust, and neutral facial expressions and achieve precise results. We want to train another artificial neural network that uses the output values (AU activation) of stage 3 (Subsect. 4.5) as an input vector to classify emotions. Additional investigation is needed to determine whether emotion recognition based on image data can produce more accurate results than based on AU or AFL input data [30]. In a natural facial expression (not a grimace), the activation of the lower and upper facial muscles is usually activated in unison, resulting in the representation of a recognizable, believable emotion. Thus, in our future work, we plan a derivation of the upper facial expression based on information about the lower which seems legitimate even if it does not represent reality. A major point missing furthermore is the classification of the missing emotions (sadness, anger, fear). An approach to this challenge can be to consider more relevant tracking data from the available sensors in a virtual world. It has been shown, that especially sadness and fear can be estimated from the body posture. As body posture data is mostly available through tracking systems in VR, this may be an approach considered to form a more holistic system. The final goal should be to detect and map the human facial expressions as realistic as possible to be even able to use this representation for interpretation in the field of human studies as no real image can be taken from participants wearing an HMD [31].

References

1. Argyle, M.: Bodily Communication, 2nd edn., pp. 1–111. Routledge, London (1986)
2. Hepperle, D., Purps, C.F., Deuchler, J., Wölfel, M.: Aspects of visual avatar appearance: self-representation, display type, and uncanny valley. Vis. Comput. (2021). https://doi.org/10.1007/s00371-021-02151-0
3. Yu, K., Gorbachev, G., Eck, U., Pankratz, F., Navab, N., Roth, D.: Avatars for teleconsultation: effects of avatar embodiment techniques on user perception in 3D asymmetric telepresence. IEEE Trans. Vis. Comput. Graph. **27**, 4129–4139 (2021)
4. Yan, Y., Lu, K., Xue, J., Gao, P., Lyu, J.: FEAFA: a well-annotated dataset for facial expression analysis and 3D facial animation, April 2019. arXiv:1904.01509 [cs, eess, stat]
5. Wei, X., Zhu, Z., Yin, L., Ji, Q.: A real time face tracking and animation system. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop, pp. 71–71, June 2004
6. Zhang, J., Chen, K., Zheng, J.: Facial expression retargeting from human to avatar made easy. IEEE Trans. Vis. Comput. Graph. **28**, 1274–1287 (2020). Conference Name: IEEE Transactions on Visualization and Computer Graphics
7. Brito, C.J.D.S., Mitchell, K.: Recycling a landmark dataset for real-time facial capture and animation with low cost HMD integrated cameras. In: The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry, VRCAI 2019, pp. 1–10. Association for Computing Machinery, New York (2019)
8. Hickson, S., Dufour, N., Sud, A., Kwatra, V., Essa, I.: Eyemotion: classifying facial expressions in VR using eye-tracking cameras. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1626–1635 (2019). ISSN: 1550–5790
9. Lou, J., et al.: Realistic facial expression reconstruction for VR HMD users. IEEE Trans. Multimedia **22**(3), 730–743 (2020). Conference Name: IEEE Transactions on Multimedia
10. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: database and results. Image Vis. Comput. **47**, 3–18 (2016)
11. Ekman, P., Rosenberg, E.L.: What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS). Oxford University Press, Oxford (1997). Google-Books-ID: KVmZKGZfmfEC
12. Cuculo, V., D'Amelio, A.: OpenFACS: an open source FACS-based 3D face animation system. In: Zhao, Y., Barnes, N., Chen, B., Westermann, R., Kong, X., Lin, C. (eds.) ICIG 2019. LNCS, vol. 11902, pp. 232–242. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34110-7_20
13. Valstar, M.F., et al.: FERA 2015 - second facial expression recognition and analysis challenge. In: 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), vol. 06, pp. 1–8, May 2015
14. Mavadati, M., Sanger, P., Mahoor, M.H.: Extended DISFA dataset: investigating posed and spontaneous facial expressions, pp. 1–8 (2016)
15. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, pp. 94–101, June 2010. ISSN: 2160–7516

16. Ebner, N.C., Riediger, M., Lindenberger, U.: FACES-a database of facial expressions in young, middle-aged, and older women and men: development and validation. Behav. Res. Methods **42**(1), 351–362 (2010). https://doi.org/10.3758/BRM.42.1.351
17. Suresh, K., Palangappa, M., Bhuvan, S.: Face mask detection by using optimistic convolutional neural network. In: 2021 6th International Conference on Inventive Computation Technologies (ICICT), pp. 1084–1089 (2021)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition arXiv:1409.1556, April 2015
19. Zhihong, C., Hebin, Z., Yanbo, W., Binyan, L., Yu, L.: A vision-based robotic grasping system using deep learning for garbage sorting. In: 2017 36th Chinese Control Conference (CCC), pp. 11 223–11 226, July 2017. ISSN: 1934–1768
20. King, D.E.: Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. **10**(60), 1755–1758 (2009)
21. Tian, Y.-L., Kanade, T., Cohn, J.F.: Recognizing action units for facial expression analysis. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2), 19 (2001)
22. Onizuka, H., Thomas, D., Uchiyama, H., Taniguchi, R.-I.: Landmark-guided deformation transfer of template facial expressions for automatic generation of avatar blendshapes. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), pp. 2100–2108. IEEE (2019)
23. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence
24. Ichim, A.-E., Kadleček, P., Kavan, L., Pauly, M.: Phace: physics-based face modeling and animation. ACM Trans. Graph. **36**(4), 153:1-153:14 (2017)
25. Lewis, J.P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F., Deng, Z.: Practice and Theory of Blendshape Facial Models, p. 23 (2014)
26. d'Eon, E., Francois, G., Hill, M., Letteri, J., Aubry, J.-M.: An energy-conserving hair reflectance model. Comput. Graph. Forum **30**(4), 1181–1187 (2011)
27. Blais, C., Roy, C., Fiset, D., Arguin, M., Gosselin, F.: The eyes are not the window to basic emotions. Neuropsychologia **50**(12), 2830–2838 (2012)
28. Guarnera, M., Hichy, Z., Cascio, M., Carrubba, S., Buccheri, S.L.: Facial expressions and the ability to recognize emotions from the eyes or mouth: a comparison between children and adults. J. Genet. Psychol. **178**(6), 309–318 (2017). https://doi.org/10.1080/00221325.2017.1361377
29. Biondi, G., Franzoni, V., Gervasi, O., Perri, D.: An approach for improving automatic mouth emotion recognition. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11619, pp. 649–664. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24289-3_48
30. Dinculescu, A.: Automatic identification of anthropological face landmarks for emotion detection. In: 2019 9th International Conference on Recent Advances in Space Technologies (RAST), pp. 585–590 (2019)
31. Wölfel, M., Hepperle, D., Purps, C.F., Deuchler, J., Hettmann, W.: Entering a new dimension in virtual reality research: an overview of existing toolkits, their features and challenges. In: International Conference on Cyberworlds (CW) (2021)