# A Digital Human System with Realistic Facial Expressions for Friendly Human-Machine Interaction

Anthony Condegni[1], Weitian Wang[1], Rui Li[1] (✉)

[1] School of Computing, Montclair State University, NJ 07043, USA
liru@montclair.edu

**Abstract.** Digital human technology can be applied to many areas, especially at the interaction of intelligent machines, digital 3D, and human-machine interaction. This paper develops a digital human system that enhances friendly interaction between humans and machines through synchronized and realistic facial expressions, based on tracking data from a human user. The details of this paper can be summarized in three parts. First, customize and build a 3D model for the digital human. Second, create a parameterized 3D model for different facial animations. Third, track the user for synchronizing the facial expressions of the customized digital human. The experimental results and analysis in this paper demonstrate the effectiveness and advantages of the developed system, which can create realistic and natural synthesized facial expressions. The system can be further generalized to other intelligent computing areas, such as smart medical care, autonomous vehicles, and companion robots.

**Keywords:** Digital Human, Intelligent Computing, Human-Machine Interaction

## 1 Introduction

While it is true that digital human technology has been widely used in the entertainment industry, its potential applications extend far beyond games and social media. In fact, digital human technology can pave the way for new possibilities in a wide range of fields, especially the intersection of intelligent machine, digital 3D and human-machine interaction. By incorporating digital human systems into intelligent machines, it is possible to create machines that can interact with humans in a more natural and engaging way. And this can be particularly valuable in areas like smart medical care, where intelligent machine can be used to assist with patient care and rehabilitation. For digital human technology, facial animation synthesis is a key component. The ability to create accurate facial expressions and emotions is very important in developing a useful and friendly digital human.

There are multiple techniques for simulating facial animation in digital humans, such as motion capturing, keyframe animation, and procedural animation. Motion capture records a user's facial expression via computer vision or motion sensors. The captured data is then mapped onto the face of the digital human for animation synthesis. Keyframe animation requires manually animating each facial parts, such as the eyes, mouth, and jaw for target expressions. Procedural animation is based on computer algorithms to generate facial expressions via tunning multiple parameters. Among these

animation techniques, motion capture based facial animation has multiple advantage over the other two techniques. One of the main advantages of motion capture-based facial animation is that it allows for quick and lifelike animation synthesis due to the realistic facial data that are automatically captured from real humans. However, this is difficult to achieve in keyframe animation and procedural animation, which require a large amount of manual input and tuning. Until now, many works have leveraged the power of motion capture technique for facial animation synthesis [1]– [4]. For example, Richard et al. proposed a variational autoencoder (VAE) based method for realistic facial expressions on a codec avatar [5]. Li et al. converted 2D captures into 3D animation based on the adaptive principal component analysis (PCA) [6]. As a linear model, the PCA method was widely used in facial tracking and animation synthesis due to its compact representation and computational efficiency. In these works, detailed and subtle facial expression synthesis has been achieved by capturing facial movement data from real humans.

Leveraging the power of motion capture technique, this paper developed a digital human system that enhances friendly interaction between humans and machines through synchronized and realistic facial expressions based on tracking data from a human user. The following content of the paper is organized as follows: Section 2 provides a survey of state-of-the-art works closely related to this paper. Section 3 presents a detailed system overview of the proposed system. The theoretical knowledge associated with the proposed system is summarized in Section 4. To demonstrate the effectiveness and efficiency of the proposed system, Section 5 provides a detailed description of the experimental design, results, and analysis. Conclusion and future works are stated in Section 6.

## 2 Related Works

The idea of a model being controlled by a real human has been researched in the past with varying outcomes. Past methods can be categorized into different categories based on their focus. Some approaches focus on facial reconstruction to create a realistic model [7]-[13], while others concentrate on 3D facial animation synthesis [14]-[19]. Another consideration is the simplicity and ease of implementation of the methods. For equipment, various past methods only require a mobile phone for facial tracking [2]-[4], [20], making them easy to use. As a result, some methods focus on a simple linear model, which is computationally efficient [21]-[26].

In recent years, the technology of facial synthesis has been successfully progressed. Liu et al. [27] and Richards et al. [5] both use a method of combining gaze and audio input with a VAE-based model by a Kinect sensor. Navarro et al. proposed a method of mapping facial synthesis onto a 3D cartoon character with no manual calibration [28]. Weise et al. focus on fast and robust mapping onto a character from video using temporal coherence [29]. Some works concentrate on creating accurate and realistic facial animation. Cao et al. synthesized facial animation using binocular video information for a regression model [14], which allows for animation on a photo-realistic avatar but requires training for different lighting scenarios and environments. Weng et

al. developed a facial animation system that can run on mobile devices using a regression method that remains accurate animation results when mapping onto the models. They also focus on training for different lighting conditions to improve robustness [20]. Li et al. use a calibration-free method for on-the-fly correctives when using facial synthesis on models. Their method transforms 2D real-time video into 3D animation using blendshape theory [6].

## 3 System Overview

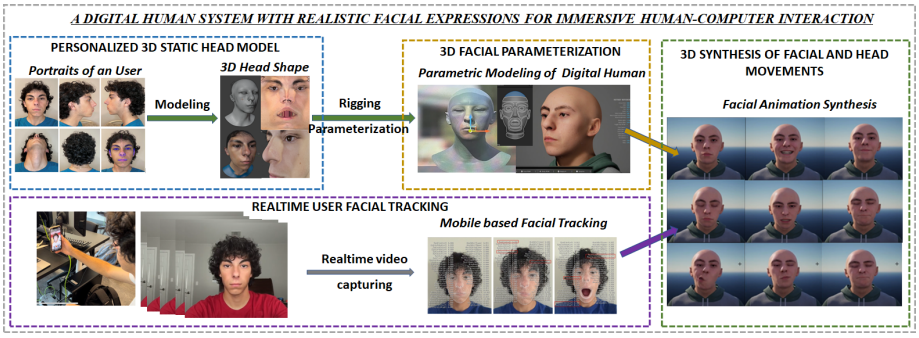



**Fig 1.** System framework. The proposed work consists of four main components: (1) personalized 3D head model reconstruction, (2) parametrization of the 3D model, (3) real-time facial tracking from a user, (4) 3D facial expression synthesis.

This paper presents a digital human system designed to enhance the human-computer interaction process by generating synchronized and realistic facial expressions based on real-time users' facial data. As shown in Fig. 1, the system framework captures a user's facial and head movements using a camera as the main input and displays the synchronized facial animation on a remote host computer as the output. The proposed system consists of four main parts: (1) personalized 3D static head modeling, (2) 3D facial parametrization, (3) user facial tracking, and (4) 3D facial expression synthesis.

The system begins with 3D modeling of a user's head. For this process, portrait photos of a user from five angles (front, back, left side, right side, and upward) are used as reference for 3D reconstruction of head model [30]. After obtaining the 3D model of the digital human, we manually refine the textures of the model to ensure accurate and realistic results and fix any texture seams to avoid distortion. To prepare the digital human for generating specific facial movements, the 3D model is further parameterized for facial tracking and movement control. In addition, we use a mobile-based movement tracking method to capture multiple facial features from a human for animating subtle and realistic facial expressions. These captured data are then mapped onto parametric model of the digital human to drive the facial and head movements.

# 4 Proposed Method

## 4.1 3D Reconstruction of a User

To create a customized 3D shape of the digital human, multiple 2D portrait photos of the user are used. These photos are captured using a mobile phone and taken in the same lighting and position conditions to ensure consistency. The 3D reconstruction process involves using portrait photos from five different angles (front, back, left side, right side, and upward) as shown in Fig. 2 (a) to Fig. 2(e). Fig. 2 (a) to Fig 2(e) show examples of these five photos that are used. From Fig 2(a) to Fig 2(e) are viewports from left, front, back, right side, and upward angles, respectively.

The resulting head model reconstruction, shown in Fig. 2 (f) to Fig. 2(g), accurately reflects the user's face and head shape. However, to achieve a realistic virtual representation of the user, the model is further fine-tuned by manual sculpting. This process ensures that the 3D mesh is both accurate and realistic, which is crucial for an immersive experience. The textures and texture seams of the model are also manually fixed to avoid distortion. The above preparation works result in a customized 3D static model that provides an accurate and realistic representation of the user.

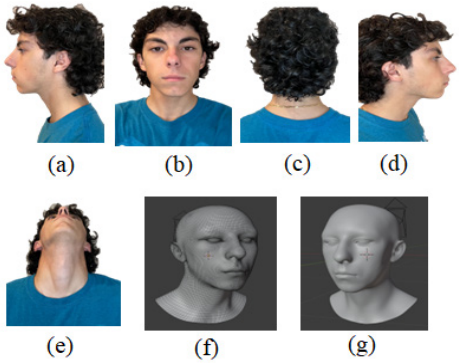



Fig 2. 4.1 3D Reconstruction of a User. (a) to (e) display the user's portrait photos of the left side, front, back, right side, and upward views, respectively, which are used reconstruction of the 3D static head model. (f) and (g) show the created 3D static head model with and without meshes information, seperately.

## 4.2 Head Rotation

Once the static head model is sculpted and created, it needs to be further parametrized for head movements. In 3D graphics and animation, the movements of the 3D model are generated by the movements of points on the model. The mathematical representation of these points can be expressed as a list of tuples, with each tuple representing the coordinates of a point on the model:

$$H_0 = (p_0, p_1, p_2, \dots p_k \dots, p_N), \quad (1)$$

where $H_0$ is the head model, $p_k = (x_k, y_k, z_k)$ is the coordinate of the kth point for the model mesh, $k = 1,2,3, \dots N$. N is the total number of points.

To enable realistic head movements, the coordinates of the points need to be updated in response to the movements of the user. In the context of our research, we focus on head rotations as the primary movement to be captured and reflected in the 3D model. The head rotation can be calculated by Euler angles $\theta, \rho, \gamma$ for a combination of pitch, yaw, and roll rotation. Here, the right-handed coordinate system is used. Three distinct formulas are defined for these rotations respectively:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}, \quad (2)$$

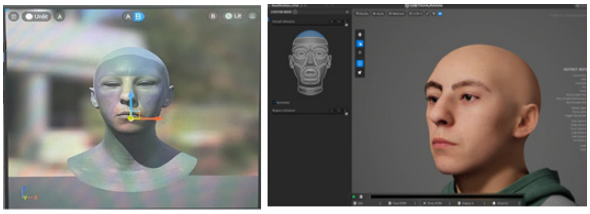
$$R_y(\rho) = \begin{pmatrix} \cos(\rho) & 0 & \sin(\rho) \\ 0 & 1 & 0 \\ -\sin(\rho) & 0 & \cos(\rho) \end{pmatrix}, \quad (3)$$

$$R_z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (4)$$

where $R_x(\theta), R_y(\rho), R_z(\gamma)$ are three matrices used for calculating the rotation around the x, y, and z axes. The complex head rotation can be realized by matrix multiplication. And this process can be represented by the following equation:

$$\begin{cases} \Phi = R_x(\theta)R_y(\rho)R_z(\gamma) \\ H_j = \Phi \cdot H_0 \end{cases}, \quad (5)$$

where $\Phi$ is the matrix multiplication of $R_x, R_y, R_z(\gamma)$. $H_j$ is the target head pose. Through multiplying the $\Phi$ with head static model $H_0$, the target pose of the head can be achieved.



(a) (b)

Fig 3. Examples of creating parametrical head model. (a) Example of the static model that is created and will be parametrized. (b) Example of the parametrical head model.

Fig. 3 shows examples of the created parametrical head model in Unreal Engine Metahuman library [32]. Fig. 3 (a) shows the static model that is created and will be parametrized. Fig. 3 (b) shows the parametrized head model which is ready for animating head rotation and facial animation. The Blendshape theory is used for efficiently generating facial expression and it will be introduced in the following section.

### 4.3 Blendshape Theory

In addition to head rotation, facial expressions are also considered in this work to improve human-computer interaction. Facial expression is complex, and its related points do not move as simply as those involved in head rotation. This is because facial skin undergoes deformations to express different emotions. Therefore, we have built a parametric facial model based on the Blendshape theory to realize these deformations. The model can be represented by:

$$s = (p_i, p_{i+1}, p_{i+2}, \ldots . p_M), \quad (6)$$

where $\{p_i, p_{i+1}, p_{i+2}, \ldots . p_M\} \subseteq \{p_0, p_1, p_2, \ldots p_k \ldots, p_N\}$ , $0 \leq M \leq N$ and $(p_i, p_{i+1}, p_{i+2}, \ldots . p_M)$ indicate all points in the face region. Fig. 4 takes a point on the upper lip as an example to show how changing point's coordinates can deform the shape of the face. Fig. 4 (a) shows a point $p_i$'s position on the upper lip for neutral face. Fig. 4(b) shows the point $p_i$ moves its position from original $p_i(0)$ to $p_i(1)$ for mouth open shape.

To synthesize specific facial expressions, this paper utilizes the Blendshape theory [22], which is a linear model for generating synthesized facial expressions. Compared to other parametric and physiological models, the Blendshape theory has been widely applied in numerous applications, such as animation, films, and mobile apps, due to its computational efficiency. The mathematical representation of the Blendshape theory can be expressed by the following equation:

$$f = s_0 + \sum_{k=1}^{n} w_k (s_k - s_0), \tag{7}$$

where $f$ is the current face that is being shown. $s_0$ is the neutral face with no facial expression. $s_k$ is the target face, $k = 0, 1, 2, \ldots, n$ . $s_k - s_0$ is the difference between the kth target face and the neutral face. $w_k$ is the influence weight for the kth target face, meaning the intensity of the movement of the related points on the face model. n is the total number of the target faces. $w_k \in [0,1]$, where 0 means that there is no influence, and 1 means the maximum influence for a target facial expression.

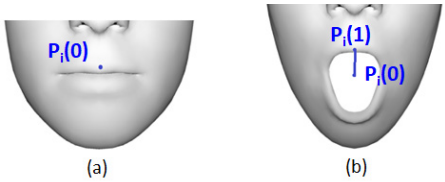



Fig. 4. Example of how point movements are used to generate facial expressions (the origional pictures [34] were used and modified here for better explaning the theory in this paper.) (a) a point's position for neutral face. (b) the point's position change for open mouth.

# 5 Experiments

## 5.1 Experimental Setup

The proposed system in this paper is implemented on a remote laptop and a mobile phone. The laptop, installed with Unreal Engine, is used to calculate the facial animation and display the synthesized graphics results. The mobile phone is used to track the user's facial information and head rotation in real-time. The laptop and the mobile phone are connected wirelessly, and the "live link" [33] library installed in the mobile phone is used to track facial expression and head pose. The tracked facial information and head rotation angles are then transferred to the laptop for animating the 3D digital human. Fig. 5 shows an example of the experimental setup where the user uses the camera on the mobile phone to track facial and head information. This easy-to-implement experimental setup allows for quick facial and head animation synthesis.

## 5.2 Experimental Results and Analysis

The tracking results are portrayed on the application while facial tracking in a live view. As discussed, there are plenty of points marked on the face based on a neutral pose. These points are altered based on the expressions portrayed. This is done using the

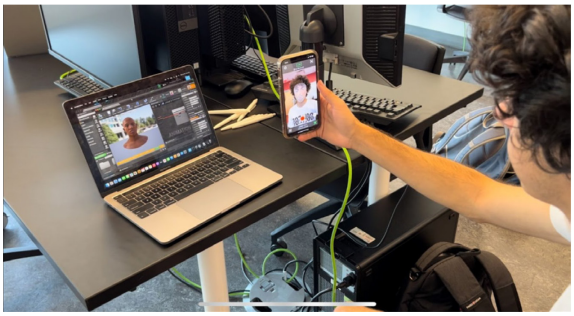


Fig. 5. The experimental setup consists of a mobile phone for facial tracking and a laptop for animation synthesis, connected wirelessly.

blendshape algorithm, gathering numerical data for each point of the face called blendshape data. The different numbers show the weight of the specific blendshape expressions (from the AR-Kit library [34]). To better observe the facial tracking, Fig. 6 shows tracking results of six different facial movements. Specifically, movements tracking of eyes, mouth, cheek, and head pose are tested. Fig. 6 (a) shows the tracking of neutral face for comparison with other facial expression. Fig. 6 (b)-Fig. 6 (f) shows the tracking results for smile, head tilt up, cheek puff, eye wink, and mouth open. In addition to track the facial movements, the utilized tracking methods also track the user's eye ball movements and head poses, which are very important for realistic facial and head animation synthesis. Fig 6 (a) shows the tracking results for neutral face. All the blendshape data are close to 0 as there is no expression was made for neutral face. When there is a new expression, Fig. 6(b) to Fig. 6(f), the corresponding blendshape data will be changed. For instance, blinking the left eye will change the values of the blendshapes for "left eye blink", "left eye squint", and "left eyebrow down" (Fig. 6 (b)). When the mouth is open, the value of blendshape for "jaw being opened" is greatly affected (Fig. 6 (c)). Similarly, obvious blendshape values are being altered based on expressions of smiling (Fig. 6(d)), cheek puffing (Fig. 6(e)). In addition to the blenshape values for facial expressions. There are three values for recording head poses: yaw, pitch, and roll. Fig. 6 (f) shows the tracking results for head upwards.

The tracked facial and head pose data are then mapped onto the synthesized human head to simulate facial expressions for the digital human. Fig. 7 shows detailed synthesis results for head pose, eye movements, mouth shapes, and combination effects. In Fig. 7(a), different head movements (yaw, roll, pitch, and combination) are synthesized. Fig. 7(a-1) shows the synthesis result of yaw head movements that frequently occur during human communication. Fig.7(a-2) shows the synthesis result of roll head

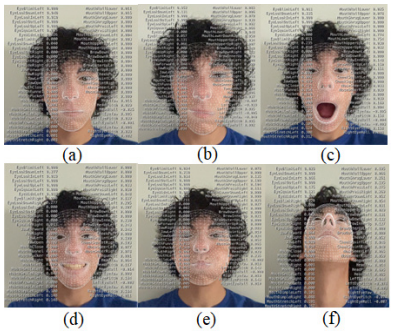



Fig. 6. Examples of facial tracking results for different facial expression. From (a) to (f), the expressions are neutral, smile, head tilt up, cheek puff, left eye wink, and mouth open.

movements. Fig. 7(a-3) shows the synthesized pitch head movements. A synthesized combination of head movements is shown in Fig. 7(a-4).

Fig. 7(b) shows the synthesis results of eye balls movements and eye lids close. Fig. 7 (b-1) shows the eyes status of neutral face for comparison. Fig. 7 (b-2) and Fig. 7 (b-3) show shows the results of looking to the left and right respectively. Fig. 7(b-4) shows eye lids close synthesis.

Fig. 7 (c) shows the different synthesis results of mouth movements. Fig. 7 (c-1) shows the synthesis result of mouth open with teeth visible when the digital human is angry. Fig. 7(c-2) shows the result of left corner of the mouth open wider with visible lower teeth. Fig. 7 (c-3) shows results of jaw dropped down and mouth open wide. This movements commonly happened for the face of surprise. Fig. 7 (c-4) shows pursed lips to the right side. This movements usually happened along with other facial movements. For better observation of mouth movements, the other facial movements are eliminated in this synthesis result.


Fig. 7. More detailed results for facial, head, and eyes movements. (a) head movements (yaw, roll, pitch, and combination) synthesis. (b) synthesis results of eye balls movements and eye lids closes. (c) a combination results with head, face, and eyes movements.

Fig. 7 (d) shows a combination synthesis results with head, face, and eyes movements. Fig. 7 (d-1) shows the neutral face for reference. Fig. 7 (d-2) shows smile face with eyes look a little bit downward, lips are pulled back, visible teeth and without head movements. Fig. 7 (d-3) shows a result of eyes look to the left side, yaw head movement to the right, lips pulled back, and visible teeth. Fig. 7(d-4) shows a result of eyes look to forward, roll head movement to the left, lips pulled back, and visible teeth.

### 5.3 Comparison and Evaluation

To prove the effectiveness of the proposed system, a comparison experiment to the user's facial expression is conducted. The results, as shown in Fig. 8, present a detailed comparison between six basic facial expressions: happy, sad, anger, fear, disgust, and surprise, with the user's facial expressions. For each frame, there are user's picture on the left and synthesis result on the right for comparison. Fig. 8 (a-1) to Fig. 8 (a-5) show the frame sequence examples of happy. There are eyes, mouth, and head's movements synthesis in them. The digital mimicry of the user's movements and expressions, such as the smile and head tilt up, are evident in Fig. 8 (a-1) to Fig. 8 (a-3). From Fig. 8 (a-3) to Fig. 8 (a-5), the digital human generates the animation for the gradually disappear of the smile according to the user.


**Fig. 8**. Comparision results of facial expression synthesis for six basic emotions. From (a) to (f) are comparision examples of 3D synthesis results for happy, sad, anger, fear, disgust, and surprise.

Fig. 8 (b-1) to Fig. 8 (b-5) show the comparison frames for the sadness facial expression. Specifically, Fig. 8 (b-1) to Fig. 8 (b-3) show the synthesized sad facial expression with eye closed and tilt down. These are also appeared on the user's face. Following the user's facial expression changes, a disappearance of sadness with eye gradually open and head return to the neutral pose is evidenced in Fig. 8 (b-3) to Fig 8 (b-5). Fig. 8 (c-1) to Fig. 8 (c-5) show the comparison frames for the anger facial expression. Similar to the user's facial expression changes, there are eyebrows, eyes, mouth, chin, and head movements for the anger facial synthesis. Fig. 8 (d-1) to Fig. 8 (d-5) show the comparison frames for the anger facial expression. Similar to the user's

eyebrows and jaw movements, the eyebrows raised and jaw dropped open are synthesized and presented in the result's sequence. Fig. 8 (e-1) to Fig. 8 (e-5) shows frame sequence example of disgust. Lip compression effects are synthesized in Fig. 8 (e-1) to Fig. 8 (e-3) while head shaking result is shown in Fig. 8 (e-2) to Fig. 8 (e-3). Synthesis of nose twitches for disgust appear in Fig. 8 (e-4). The expression recovers to neutral in Fig. 8 (e-5). After comparing the digital human and the user in Fig. 8 (e-3), the digital human is not able to mimic the user's slightly head pitch movements. This can be improved in the future work. Fig. 8 (f-1) to Fig. 8 (f-7) show the process of surprise with starting at neutral facial expression. Eyebrows raised, eye open wide, and jaw dropped down are synthesized in Fig. 8 (f-2) to Fig. 8 (f-4). The expression recovers to neutral in Fig. 8 (f-5). The comparison results prove the effectiveness of the proposed system. Video results of this paper are available here: https://youtu.be/52nNyxf_CSk

## 6 Conclusions and Future Work

In conclusion, this paper presents a digital human system that utilizes real-time users' facial data to generate synchronized and realistic facial expressions. The work in this paper accomplished three tasks: First, 3D reconstruction from multiple portrait photos of a user. Second, creation of a parameterized 3D model for different facial animations. Third, synthesis of 3D facial and head movements for the digital human system. The experimental results and analysis in this paper demonstrate the effectiveness and advantages of the developed system, which can create realistic and natural synthesized facial expressions. Moreover, the system can be further generalized to other intelligent computing areas, such as smart medical care, autonomous vehicles, and companion robots. Future work will focus on integrating the developed system with intelligent machines for better human-machine interaction.

**Acknowledgements:** Anthony Condegni gratefully acknowledges support from the MSU CSAM Faculty-Student Summer Program. This work was also supported in part by the National Science Foundation under Grant CNS-2104742 and in part by the National Science Foundation under Grant CNS-2117308.

## References

[1] H. Y. Ping, L. N. Abdullah, P. S. Sulaiman, and A. A. Halin, "Computer facial animation: a review," International Journal of Computer Theory and Engineering, vol. 5, no. 4, pp. 658–662, 2013.

[2] P. A. Tresadern, M. C. Ionita, and T. F. Cootes, "Real-time facial feature tracking on a mobile device," International Journal of Computer Vision, vol. 96, no. 3, pp. 280–289, 2012.

[3] Y. C. Van Wettum, "Facial landmark tracking on a mobile device," no. December, pp. 1–16, 2016.

[4] X. Liu, J. Wang, W. Zhang, Q. Zheng, and X. Li, "EmotionTracker: a mobile real-time facial expression tracking system with the assistant of public AI-as-a-Service," Proceedings of the 28th ACM International Conference on Multimedia, pp. 4530–4532, 2020.

[5] A. Richard, C. Lea, S. Ma, J. Gall, F. De La Torre, and Y. Sheikh, “Audio- and gaze-driven facial animation of codec avatars,” Proceedings of IEEE Winter Conference on Applications of Computer Vision, pp. 41–50, 2021.
[6] H. Li, J. Yu, Y. Ye, and C. Bregler, “Realtime facial animation with on-the-fly correctives,” ACM Transactions on Graphics, vol. 32, no. 4, 2013.
[7] M. Sela, E. Richardson, and R. Kimmel, “Unrestricted facial geometry reconstruction using image-to-image translation,” Proceedings of the IEEE International Conference on Computer Vision, vol. Oct, pp. 1585–1594, 2017.
[8] A. Lattas et al., “AvatarMe: realistically renderable 3d facial reconstruction ‘in-the-wild,’” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 757–766, 2020.
[9] G. Gafni, J. Thies, M. Zollhöfer, and M. Nießner, “Dynamic neural radiance fields for monocular 4D facial avatar reconstruction,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 8645–8654, 2021.
[10] J. Jo, H. Choi, I. J. Kim, and J. Kim, “Single-view-based 3D facial reconstruction method robust against pose variations,” Pattern Recognition, vol. 48, no. 1, pp. 73–85, 2015.
[11] E. Richardson, M. Sela, and R. Kimmel, “3D face reconstruction by learning from synthetic data,” Proceedings of 4th International Conference on 3D Vision, pp. 460–467, 2016.
[12] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, “High-quality single-shot capture of facial geometry,” ACM SIGGRAPH 2010 Papers, SIGGRAPH 2010, 2010.
[13] G. Schwartz et al., “The eyes have it: an integrated eye and face model for photorealistic facial animation,” ACM Transactions on Graphics, vol. 39, no. 4, 2020.
[14] C. Cao et al., “Real-time 3D neural facial animation from binocular video,” ACM Transactions on Graphics, vol. 40, no. 4, pp. 1–17, 2021.
[15] F. Pighin and D. H. Salesin, “Realistic facial animation using image-based 3D morphing,” Microsoft Research, pp. 1–26, 1997.
[16] J. Lou et al., “Realistic facial expression reconstruction for VR HMD users,” IEEE Transactions on Multimedia, vol. 22, no. 3, pp. 730–743, 2020.
[17] D. Das, S. Biswas, S. Sinha, and B. Bhowmick, “Speech-driven facial animation using cascaded GANs for learning of motion and texture,” Proceedings of Computer Vision-ECCV, pp. 408–424, 2020.
[18] D. Jiang, Y. Zhao, H. Sahli, and Y. Zhang, “Speech driven photo realistic facial animation based on an articulatory DBN model and AAM features,” Multimedia Tools and Applications, vol. 73, no. 1, pp. 397–415, 2014.
[19] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou, “Real-time facial animation with image-based dynamic avatars,” ACM Transactions on Graphics, vol. 35, no. 4, pp. 1–13, 2016.
[20] Y. Weng, C. Cao, Q. Hou, and K. Zhou, “Real-time facial animation on mobile devices,” Graphical Models, vol. 76, no. 3, pp. 172–179, 2014.
[21] E. Chuang and C. Bregler, “Performance driven facial animation using blendshape interpolation,” Computer Science Technical Report, vol. 2, no. 2, p. 3, 2002.
[22] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng, “Practice and theory of blendshape facial models,” Eurographics, vol. 1, no. 8, pp. 1–23, 2014.
[23] S. Bouaziz, Y. Wang, and M. Pauly, “Online modeling for realtime facial animation,” ACM Transactions on Graphics, vol. 32, no. 4, 2013.

[24] C. Cao, Y. Weng, S. Lin, and K. Zhou, “3D shape regression for real-time facial animation,” ACM Transactions on Graphics, vol. 32, no. 4, 2013.
[25] H. X. Pham, Y. Wang, and V. Pavlovic, “End-to-end learning for 3D facial animation from speech,” in ACM International Conference on Multimodal Interaction, 2018, pp. 361–365.
[26] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, “Learning controls for blend shape based realistic facial animation,” SIGGRAPH 2006 - ACM SIGGRAPH 2006 Courses, 2006.
[27] Y. Liu, F. Xu, J. Chai, X. Tong, L. Wang, and Q. Huo, “Video-audio driven real-time facial animation,” ACM Transactions on Graphics, vol. 34, no. 6, 2015.
[28] I. Navarro et al., “Fast facial animation from video,” ACM SIGGRAPH 2021 Talks, SIGGRAPH 2021, 2021.
[29] T. Weise, S. Bouaziz, H. Li, and M. Pauly, “Realtime performance-based facial animation,” ACM Transactions on Graphics, vol. 30, no. 4, pp. 1–10, 2011.
[30] “Blender 3.1.” Blender Foundation, Community, 2022.
[31] KeenTools, “KeenTools FaceBuilder for Blender.” KeenTools, 2022.
[32] “MetaHumans.” Epic Games, 2022.
[33] “ARKit.” [Online]. Available: https://developer.apple.com/documentation/arkit.
[34] “ARKit Face Blendshapes (Perfect Sync).” [Online]. Available: https://arkit-face-blendshapes.com/.