Deep Learning Augmented Realistic Avatars for Social VR Human Representation

Matthijs van der Boon
mjvanderboon@gmail.com
ETH Zurich
Zurich, Switzerland
TNO
The Hague, the Netherlands

Leonor Fermoselle
leonor.fermoselle@tno.nl
TNO
The Hague, the Netherlands

Frank ter Haar
frank.terhaar@tno.nl
TNO
The Hague, the Netherlands

Sylvie Dijkstra-Soudarissanane
sylvie.dijkstra@tno.nl
TNO
The Hague, the Netherlands

Omar Niamut
omar.niamut@tno.nl
TNO
The Hague, the Netherlands

ABSTRACT

Virtual reality (VR) has created a new and rich medium for people to meet each other digitally. In VR, people can choose from a broad range of representations. In several cases, it is important to provide users with avatars that are a lifelike representation of themselves, to increase the user experience and effectiveness of communication. In this work, we propose a pipeline for generating a realistic and expressive avatar from a single reference image. The pipeline consists of a blendshape-based avatar combined with two deep learning improvements. The first improvement module runs offline and improves the texture map of the base avatar. The second module runs inference in real-time at the rendering stage and performs a style transfer to the avatar's eyes. The deep learning modules effectively improve the visual representation of the avatar and show how AI techniques can be integrated with traditional animation methods to generate realistic human avatars for social VR.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Virtual reality**; *Rendering*; Animation.

KEYWORDS

virtual reality, generative adversarial networks, real-time deep learning, human avatars

ACM Reference Format:
Matthijs van der Boon, Leonor Fermoselle, Frank ter Haar, Sylvie Dijkstra-Soudarissanane, and Omar Niamut. 2022. Deep Learning Augmented Realistic Avatars for Social VR Human Representation. In *ACM International Conference on Interactive Media Experiences (IMX '22), June 22–24, 2022, Aveiro, JB, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3505284.3532976

IMX '22, June 22–24, 2022, Aveiro, JB, Portugal

ACM ISBN 978-1-4503-9212-9/22/06.
https://doi.org/10.1145/3505284.3532976

1 INTRODUCTION

The COVID-19 pandemic has highlighted that working from home will remain the norm for millions of employees. Especially when working remotely, it is important to feel a sense of presence in digital meetings. Virtual Reality (VR) has proven to increase the sense of social presence over 2D videoconferencing systems [4, 7]. With sufficient technological advancements, VR meetings may become the norm over 2D video-based meetings due to the benefits they offer in terms of users feeling a heightened social presence.

A variety of platforms already exist for meeting in VR environments [20, 31–33, 35]. However, VR meeting platforms are currently still lacking regarding the visual representation of users. Users are often portrayed as cartoon avatars with limited facial expressions transferred to these avatars. While these avatars do allow for the user to be represented in the VR environment in a rudimentary way, research has shown that a realistic representation of the user's facial expressions is very relevant for effective social communication [1]. A correct eye gaze is particularly important for communication using avatars [16].

The importance of representing facial expressions is also the reason current direct capture methods of the user, such as volumetric video, are troublesome for social VR applications, since the VR head-mounted display (HMD) occludes the majority of the user's face and thus facial expressions and eye gaze.

Some work has been conducted in the field of animatable photorealistic human avatars. However, little progress has been made towards a pipeline that is usable in a VR scenario for social communication. Most photorealistic solutions [3, 26, 28, 30, 36, 37, 45] require a cumbersome amount of data per user, barring real-world adoption, or solutions provide highly realistic avatars that are unfortunately not animatable [6, 17, 26, 29, 40].

In this work, we propose a framework for generating realistic animatable human head avatars from a single reference image. To overcome the limitations of using only a single reference image, we employ a deep learning UV texture completion solution to generate the full avatar texture. Furthermore, to increase the quality of the avatar's eyes we train a deep learning style transfer network to generate photorealistic eyes in real-time when rendering. The entire

framework is implemented as a demo in the Unity game engine [39].

The contribution of this work-in-progress paper lies in the combination of the following elements:

- The implementation of a blendshape-based user avatar in Unity, which can be generated from a single reference image and animated through an HMD with eye and mouth trackers;
- A deep learning UV texture completion module that improves the UV texture of the avatar;
- A deep learning style transfer network that operates in real-time at the rendering stage to photorealistically synthesize the avatar's eyes.

2 RELATED WORK

2.1 Expression tracking for avatar animation in VR applications

For portraying a user in VR with an avatar, it is necessary to track the emotions the user exhibits and map them to the avatar. Especially eye gaze is an important social cue that is required for effective communication between users [1, 38]. Various VR headsets [14, 41, 43] already include hardware and software to track expressions through facial features. Infrared cameras that are mounted inside the headset are used to track eye and eyebrow movements. Some headsets further extend the expression tracking capability by mounting a camera below the headset that captures the user's mouth. In this work, the Vive Pro Eye augmented with such a facial tracker will be used. The headset offers an Eye and Facial Tracking SDK [42] that measures the activation of 52 semantic expressions such as "Left Eye Open" or "Right Mouth Down".

2.2 Human head avatars

Avatars for the human head are a widely researched field. Perhaps the most well-known paper in the field is [5]. Using data from 3D scans a parametric model for human head geometry is constructed. By varying the parameters of the model, a textured head mesh can be generated that resembles the user.

In the last five years, other proposals have improved the base model proposed by [5]. This surge of new approaches was possible thanks to the availability of more data and computational power. Current methods can roughly be categorized into three types, i.e. i) blendshape-based models, ii) rigged models and iii) implicit models. These three categories will be shortly explained hereafter. For a more extensive overview on human avatar models, the reader is referred to [10].

Blendshape-based models [12, 46] modulate a base mesh using a vector of blendshapes. Each blendshape corresponds to a semantic facial expression. For each blendshape a deformed mesh is created and during runtime, the vertices of the base mesh are displaced by linearly interpolating between the position of the vertices in the deformed meshes. Blendshapes are a widely used concept in animation and are thus supported in various platforms that can be used for making VR applications such as Unity [39] or Unreal Engine [15].

Rigged avatars [9, 11] are distinct from blendshape avatars in that they rig a base mesh using virtual bones which are then used to locally deform the mesh. At runtime, the contraction of the various bones can be controlled to generate an expressive mesh. The most recent advances in rigged human avatars incorporate machine learning to obtain state-of-the-art results which require less manual rigging. However, in practice, the user is still required to perform some steps in the process manually.

Implicit models [2, 3, 26, 28, 30, 36, 37, 45] do not use either blendshape or rig-based animation but instead encode the necessary animation information in latent codes which are passed through a decoder to generate a mesh.

2.3 Image-to-image translation using GANs

In this paper, only a short definition of two sub-problems of image-to-image translation using generative adversarial networks (GANs) will be given. For a more extensive overview on this topic, the reader is referred to [19].

2.3.1 Image completion. Image completion is the task of filling in missing or distorted parts of images [44, 47]. GANs have shown great performance in this task. The GANs are trained in an unsupervised, adversarial manner to synthesize the missing regions in an image. In this context, GANs have been explored for use in VR applications for HMD removal processes. [34] for example, synthesizes a face from an RGB+depth capture where the face in the original image is occluded with an HMD.

2.3.2 Style transfer. Style transfer involves generating an output image that is based on an input image. Depending on the application, the network can be trained on supervised or unsupervised data. One well-known example of unsupervised style transfer is the horse-to-zebra translation [48]. The supervised problem is similar, however, in this case a data set with paired input-output images is available. A common example of a supervised style transfer is to generate a realistic image from a segmentation map [21].

3 PROPOSED VIRTUAL HUMAN FRAMEWORK

In this work, we propose a pipeline that takes a single reference image of a user as input and outputs a VR ready avatar model. The work-in-progress pipeline is composed of a blendshape-based avatar model with two image-to-image translation modules to improve the results of the visual representation. The process consists of three steps:

(1) Avatar generation
(2) UV texture completion
(3) Real-time style transfer for the eye region

The first two steps are performed offline for each user and generate a user-specific avatar, which is imported in the Unity game engine. The final step is performed at runtime within the rendering pipeline available in Unity.

3.1 Avatar generation

In this work, we use a blendshape-based model for the accessible integration in the Unity animation system. We use the FaceScape model [46] to generate the base avatar. FaceScape allows to generate an animatable avatar from a single input image of a user. The

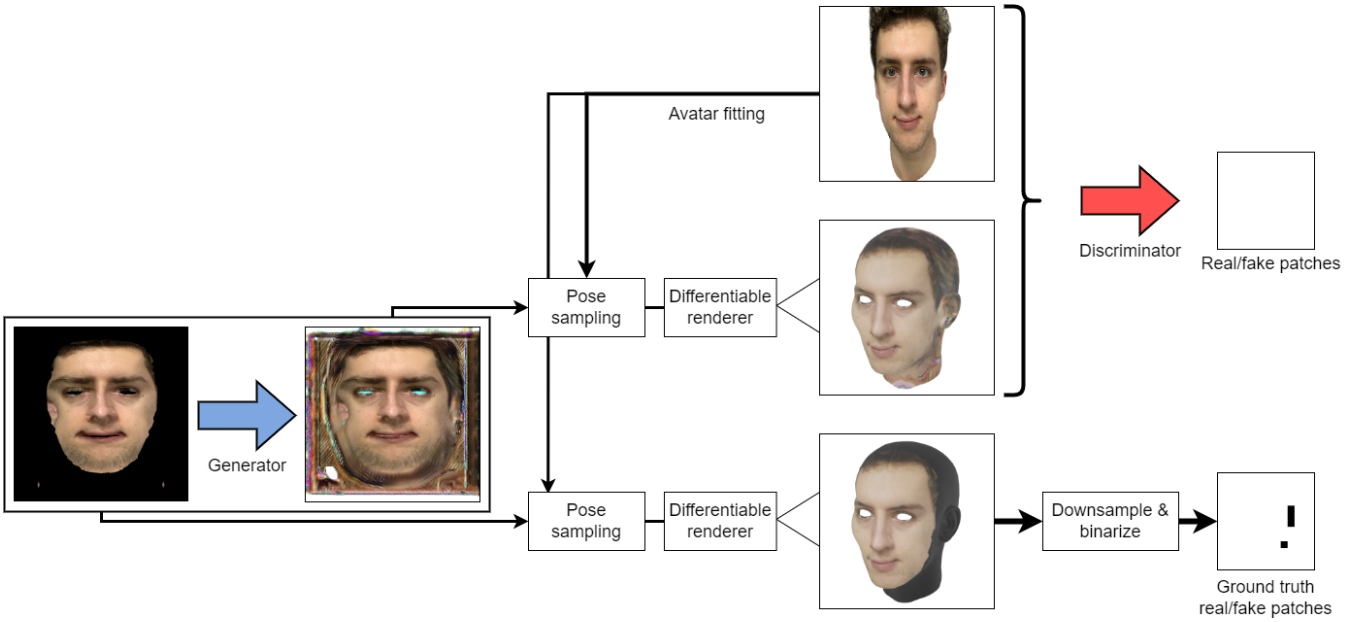


Figure 1: The architecture used to train the UV texture completion module. A generator network synthesizes an augmented UV texture from a UV texture with black occluded regions. Both the input and output UV textures are wrapped to the avatar mesh generated from the input image. The textured meshes are then rendered using a differentiable renderer. The image rendered using the original UV texture is downsampled and binarized to form a ground truth for training the discriminator network. The original input image and the image rendered using the synthesized UV texture are passed through the discriminator network. The discriminator network predicts if the input image is 'real' or 'fake' in a 14x14 grid in image space. The output for the original input image is used to train the discriminator, since all output labels should be 'real'. The output from the synthesized UV texture is used to train the generator in an adversarial manner.

generated avatar is animatable using semantic blendshapes that can be mapped to the output of the Eye and Facial Tracking SDK. Furthermore, because the avatar is based on blendshapes, it is possible to integrate the model in the animation system of Unity. The FaceScape model is a bilinear model that operates by decomposing the human head geometry along an expression and identity dimension. The identity vector $\boldsymbol{w}_{id} \in \mathbb{R}^{938}$ encodes the face geometry that is unique to each subject. The expression vector $\boldsymbol{w}_{exp} \in \mathbb{R}^{52}$ encodes the facial expression for 52 semantic blendshapes. To generate the mesh, the identity and expression vector are multiplied with a core tensor C_r as shown in Eq. 1 resulting in a mesh of 26317 vertices.

$$V = C_r \times \boldsymbol{w}_{exp} \times \boldsymbol{w}_{id} \tag{1}$$

In the proposed pipeline, the FaceScape model is used to generate a base model and 52 blendshapes for a user from a single input image. A UV texture is unwrapped onto the mesh for the parts of the face that are visible in the input image. The mesh, blendshapes, and UV texture are imported into Unity.

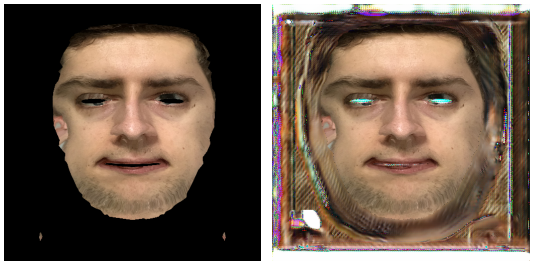
3.2 UV texture completion

The base model includes a UV texture map. However, for any general input image, the unwrapped texture will always contain regions that are occluded by the face itself. Typically, the back and side of the face are occluded in near-frontal images.

Other works have employed deep learning solutions to counteract the issue of an incomplete unwrapped texture. One approach is to train a generative network to synthesize the complete texture map using supervised complete texture maps [8]. Such a complete texture map data set does not exist publicly and would have to be created by a graphics artist. Instead, an approach inspired by [24] was taken to train a GAN to synthesize the complete texture map without supervision from complete texture maps. For a more detailed overview of this approach, please see [24] or contact the authors. Only a brief overview of the method will be given hereafter.

The generative network is trained using images from the FFHQ data set [23]. The previously discussed avatar model is used to unwrap an (incomplete) texture map for each user. The incomplete texture map is run through a generator network. The synthesized texture map is then wrapped to the face mesh and the mesh is rendered using a differentiable renderer [22] from a randomly sampled viewpoint. The random sampling ensures that areas of the texture map which were occluded in the original input image are now visible on the rendered image. The generator is trained in an adversarial manner on the rendered image. An overview of the approach is shown in Fig. 1.

An example of a completed UV texture is shown in Fig. 2. The UV texture completion network does not synthesize the texture in the occluded regions perfectly, the results are, however, a large improvement over having occluded parts of the UV texture black.



(a) Original UV texture (b) Synthesized UV texture

(c) Wrapped original UV texture (d) Wrapped synthesized UV texture

Figure 2: The result of the UV completion module. The original UV texture (2a) contains black areas that are not visible in the frontal input image. This results in an avatar with black regions as well (2c). The UV texture is passed through a generator network which is tasked with filling in the black regions to generate a full UV texture (2b). Using the synthesized UV texture the avatar has a more convincing texture (2d) in areas where the base avatar is black.

3.3 Real-time style transfer on eyes

The base avatar with the completed UV texture is ready to be used in any VR application. However, the avatar does not provide an entirely convincing representation of the user. An example of the base avatar can be seen in Fig. 3.

Figure 3: The base avatar with manually added eyeballs. The eyeballs are modelled as spheres with the texture of an eye applied on them. It is apparent that the eyes do not create a convincing representation.

The main cause for the unconvincing representation was identified to be a lacking representation of the user's eyes. The eyes are each modelled with a basic eyeball mesh, which is animated using the eye gaze measurements from the headset, through the SRanipal SDK [18]. We observed the lack of connection between the face and eye meshes to cause a jarring, sliding effect between the eyeball and the head mesh. Furthermore, because the base avatar mesh is not connected to the eyeball, gaps are visible from some viewpoints. While it may be possible to perform manual adjustments of the base and eyeball mesh for each different user to alleviate these issues, this is undesirable from an ease-of-use perspective. We propose instead to improve the base model by running a style transfer network in real-time, just before rendering to the VR head-mounted display, that is tasked with generating realistic eye images from cartoon style input images. The network runs just before each frame is rendered onto the headset of the VR user. The network style transfers the left and right eye of the avatar separately.

Figure 4: The results of the style transfer network on testing data. From left to right the columns display input, output and ground truth images. The rows show the robustness of the network to differences in iris colour and gaze direction. The iris colour is encoded in the colour of the iris in the input image. It should be noted that the colour of the iris in the input image is calculated by taking the mean of the colour in the mask for the sclera, pupil and iris combined, therefore the colour in the input image does not necessarily match the colour of the iris alone.

3.3.1 Network and training data. For the style transfer network, we use the publicly available implementation of [21]. This network maps an input image from a source domain to a target domain. For the target domain, we process the FFHQ data set [23] and use dlib [25] to crop images around the eye. For each eye image, a paired input image is generated by first segmenting out the sclera, iris, and pupil, and then colouring a circle at the center of the pupil to encode the gaze direction of the target image. For the segmentation step, a face parse network [27] is used which is trained on the CelebAMask-HQ data set. The pupil location is manually annotated

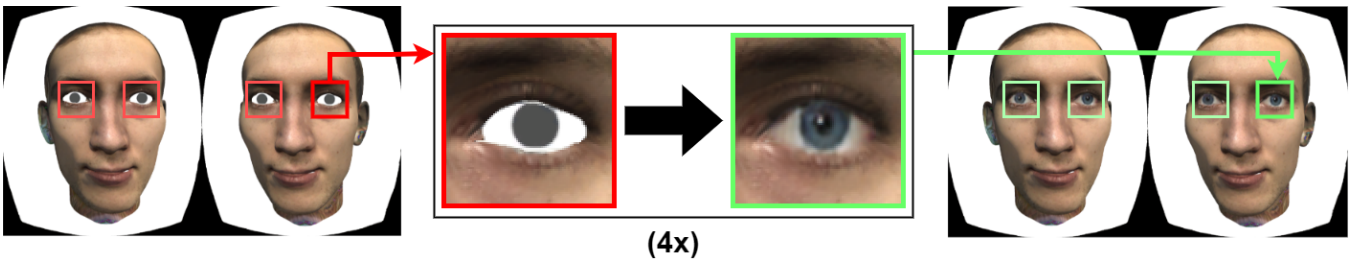


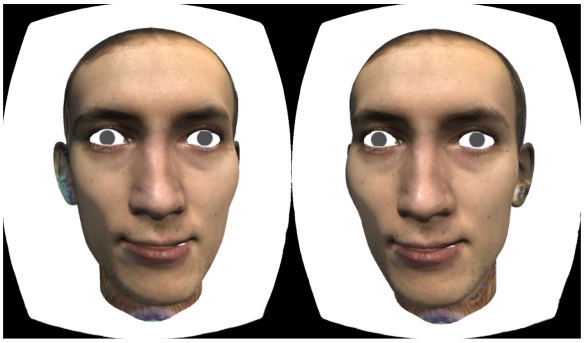
Figure 5: The structure of the implementation of the eye style transfer network. The eye style transfer network operates just before images are rendered to the HMD. The left stereoscopic image is the pre-render image for the current frame. In total the network is run four times: twice for each of the target's eyes, and twice for both stereoscopic parts of the current frame. Using the current location of each eyeball and a world-to-screen transformation the pre-render images are cropped to each eye and resized to the expected input size of the style transfer network. After running the cropped and resized image through the style transfer network the output is resized to the original resolution of the cropped patch and pasted back in the stereoscopic image. The output stereoscopic image is finally rendered to the HMD.

for 4,500 training images. To achieve robustness to different iris colours, the circle drawn on the location of the pupil is coloured with the mean colour in the segmentation mask from the face parse network. The final result of the style transfer network is shown in Fig. 4 along with the input and target images.

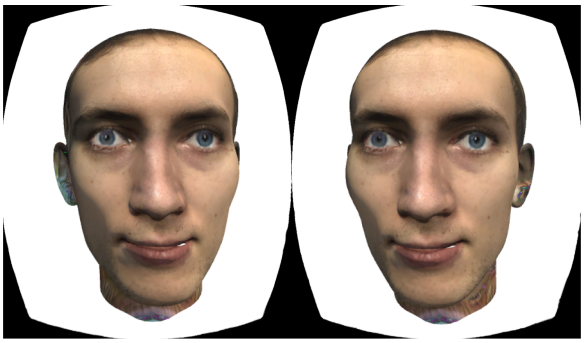
3.3.2 Integration into Unity rendering pipeline. An overview of the rendering pipeline is shown in Fig. 5. The style transfer network is integrated in Unity using the Unity Barracuda package. This package allows for the inference of neural networks in the `.onnx` format [13] in Unity. The eyes of the base avatar model need to look similar to the training data of the style transfer network. To achieve this, two white spheres are added to the base avatar as eyeballs. The spheres' texture has a circle in the middle with the desired iris colour.

4 RESULTS

With our pipeline, we succeeded in generating a 3D realistic avatar that can render facial expressions dynamically in real-time with the use of the eye and face tracking information of the Vive Pro Eye. Fig. 6 shows the results achieved using the proposed pipeline. A base avatar is generated from a single input image (Sec. 3.1). Using the UV texture completion method (Sec. 3.2) the base avatar is improved. At runtime, the eye region is passed through the style transfer network (Sec. 3.3). The avatar is animated with the eye gaze and facial expressions of the user using the Eye and Facial Tracking SDK. Fig. 7 shows the expressiveness of the avatar, and the robustness of the style transfer network to different gaze directions. The current implementation of the pipeline runs a demo where the ego user can see their own avatar. The demo application runs at 13 fps with the style transfer network running on an NVIDIA GeForce RTX 2080 Ti.



(a) Avatar without style transfer module



(b) Avatar with style transfer module

Figure 6: The results of the eye style transfer network. The style transfer network synthesizes realistic eyes for the user's avatar. It can be seen that there is high consistency of the eye gaze and iris colour between the input and output images.

5 DISCUSSION AND FUTURE WORK

In this work-in-progress paper, we presented a pipeline for generating realistic avatar representations for VR applications. A blendshape-based model is created from a single reference image. A generative network is used to fill in the occluded areas on the initial texture map. At runtime, a style transfer network is used to generate a realistic eye region in the rendering layer.

The UV completion module improves the initial texture map of the base model by synthesizing occluded areas. The current implementation still has room for improvement. It is, however,

(a) Examples of different facial expressions

(b) Examples of different eye gaze directions

Figure 7: A few examples of the expressiveness of the avatar for different facial expressions and eye gaze directions. For these figures the avatar was animated in real-time by a user wearing the HMD.

already successful in improving the baseline UV texture in which the occluded regions of the input image are black. The output of the eye style transfer network is a realistic rendering of the user's eye. The issue of the eyeball appearing to be disconnected and to slide around in the socket is effectively eliminated, and the user's eye gaze and iris colour are successfully transferred to the output avatar.

Future work is recommended to focus on improving the inference speed of the eye style transfer implementation. Currently, the demo application does not run at the typically required 90 fps for VR applications. It may be possible to reduce the complexity of the eye style transfer network, or other methods such as pruning may be employed to improve the inference speed. Additionally, offloading some of the inference steps to additional GPUs is a straightforward way to increase the fps.

Realistic avatars have great potential for VR communication. This work-in-progress paper contributes to implementable realistic user avatars for VR.

REFERENCES

[1] Benoit A Aubert and Barbara L Kelsey. 2003. Further understanding of trust and performance in virtual teams. *Small group research* 34, 5 (2003), 575–618.

[2] Ziqian Bai, Zhaopeng Cui, Xiaoming Liu, and Ping Tan. 2021. Riggable 3D Face Reconstruction via In-Network Optimization. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers Inc., Nashville, TN, 6212–6221.

[3] Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. 2021. Deep Relightable Appearance Models for Animatable Faces. *ACM Trans. Graph.* 40, 4, Article 89 (jul 2021), 15 pages. https://doi.org/10.1145/3450626.3459829

[4] Frank Biocca, Judee Burgoon, Chad Harms, and Gates Stoner. 2001. Criteria And Scope Conditions For A Theory And Measure Of Social Presence. *Presence: Teleoperators and virtual environments* 10 (01 2001).

[5] Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. Association for Computing Machinery, USA, 187–194. https://doi.org/10.1145/311535.311556

[6] Anpei Chen, Zhang Chen, Guli Zhang, Ziheng Zhang, Kenny Mitchell, and Jingyi Yu. 2019. Photo-Realistic Facial Details Synthesis from Single Image. https://doi.org/10.48550/ARXIV.1903.10873

[7] Carlos Coelho, Jennifer Tichon, Trevor Hine, Guy Wallis, and Giuseppe Riva. 2012. 2 Media Presence and Inner Presence: The Sense of Presence in Virtual Reality Technologies. *Emerging Communication: Studies in New Technologies and Practices in Communication* 9 (01 2012).

[8] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. 2017. UV-GAN: Adversarial Facial UV Map Completion for Pose-invariant Face Recognition. https://doi.org/10.48550/ARXIV.1712.04695

[9] Ziva Dynamics. 2022. Ziva Dynamics | Simulation-Ready Characters. https://zivadynamics.com/ziva-characters Accessed: 2022-03-07.

[10] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 2019. 3D Morphable Face Models – Past, Present and Future. https://doi.org/10.48550/ARXIV.1909.01815

[11] Unreal Engine. 2022. Digital Humans | MetaHuman Creator. https://www.unrealengine.com/en-US/digital-humans Accessed: 2022-03-07.

[12] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. 2021. Learning an Animatable Detailed 3D Face Model from In-the-Wild Images. *ACM Trans. Graph.* 40, 4, Article 88 (jul 2021), 13 pages. https://doi.org/10.1145/3450626.3459936

[13] The Linux Foundation. 2022. Open Neural Network Exchange. https://onnx.ai/ Accessed: 2022-03-07.

[14] HP Reverb G2. 2022. HP Reverb G2 Omnicept Edition. https://www.hp.com/us-en/vr/reverb-g2-vr-headset-omnicept-edition.html Accessed: 2022-03-14.

[15] Epic Games. 2022. Unreal Engine. https://www.unrealengine.com/en-US/ Accessed: 2022-03-14.

[16] Maia Garau, Mel Slater, Simon Bee, and Martina Angela Sasse. 2001. The Impact of Eye Gaze on Communication Using Humanoid Avatars. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 309–316. https://doi.org/10.1145/365024.365121

[17] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. 2019. GANFIT: Generative Adversarial Network Fitting for High Fidelity 3D Face Reconstruction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers Inc., Long Beach, CA, USA, 1155–1164. https://doi.org/10.1109/CVPR.2019.00125

[18] Tobii HTC Vive. 2019. Eye and Facial Tracking SDK. https://developer-express.vive.com/resources/vive-sense/eye-and-facial-tracking-sdk/ Accessed: 2022-03-06.

[19] He Huang, Philip S. Yu, and Changhu Wang. 2018. An Introduction to Image Synthesis with Generative Adversarial Nets. https://doi.org/10.48550/ARXIV.1803.04469

[20] VRChat Inc. 2019. VRChat. https://vrchat.com/ Accessed: 2020-03-06.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers Inc., Honolulu, HI, USA, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

[22] Justin Johnson, Nikhila Ravi, Jeremy Reizenstein, David Novotny, Shubham Tulsiani, Christoph Lassner, and Steve Branson. 2020. Accelerating 3D Deep Learning with PyTorch3D. In *SIGGRAPH Asia 2020 Courses* (Virtual Event) *(SA '20)*. Association for Computing Machinery, New York, NY, USA, Article 10, 1 pages. https://doi.org/10.1145/3415263.3419160

[23] Tero Karras, Samuli Laine, and Timo Aila. 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis % Machine Intelligence* 43, 12 (12 2018), 4217–4228. https://doi.org/10.1109/TPAMI.2020.2970919

[24] Jongyoo Kim, Jiaolong Yang, and Xin Tong. 2021. Learning High-Fidelity Face Texture Completion without Complete Face Texture. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Institute of Electrical and Electronics Engineers Inc., Montreal, Canda (Online), 13970–13979. https://doi.org/10.1109/ICCV48922.2021.01373

[25] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.

[26] Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. 2020. AvatarMe: Realistically Renderable 3D Facial Reconstruction "In-the-Wild". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers Inc., Seattle, Online, USA, 760–769.

[27] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, Online, USA, 5548–5557. https://doi.org/10.1109/CVPR42600.2020.00559

[28] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep Appearance Models for Face Rendering. *ACM Trans. Graph.* 37, 4, Article 68 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201401

[29] Huiwen Luo, Koki Nagano, Han-Wei Kung, Mclean Goldwhite, Qingguo Xu, Zejian Wang, Lingyu Wei, Liwen Hu, and Hao Li. 2021. Normalized Avatar Synthesis Using StyleGAN and Perceptual Refinement. https://doi.org/10.48550/ARXIV.2106.11423

[30] S. Ma, T. Simon, J. Saragih, D. Wang, Y. Li, F. La Torre, and Y. Sheikh. 2021. Pixel Codec Avatars. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers Inc., Los Alamitos, CA, USA, 64–73. https://doi.org/10.1109/CVPR46437.2021.00013

[31] Meta. 2022. Horizon Worlds. https://www.oculus.com/horizon-worlds/ Accessed: 2022-03-14.

[32] Microsoft. 2022. Mesh. https://www.microsoft.com/en-us/mesh Accessed: 2022-03-14.

[33] Mozilla. 2022. Hubs. https://hubs.mozilla.com/ Accessed: 2022-03-14.

[34] Nels Numan, Frank ter Haar, and Pablo Cesar. 2021. Generative RGB-D Face Completion for Head-Mounted Display Removal. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. Institute of Electrical and Electronics Engineers Inc., Lisbon, Portugal, 109–116. https://doi.org/10.1109/VRW52623.2021.00028

[35] Martin J Prins, Simon NB Gunkel, Hans M Stokking, and Omar A Niamut. 2018. TogetherVR: A framework for photorealistic shared media experiences in 360-degree VR. *SMPTE Motion Imaging Journal* 127, 7 (2018), 39–44.

[36] Alexander Richard, Colin Lea, Shugao Ma, Juergen Gall, Fernando de la Torre, and Yaser Sheikh. 2021. Audio- and Gaze-driven Facial Animation of Codec Avatars. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Institute of Electrical and Electronics Engineers Inc., Waikoloa, HI, USA, 41–50. https://doi.org/10.1109/WACV48630.2021.00009

[37] Gabriel Schwartz, Shih-En Wei, Te-Li Wang, Stephen Lombardi, Tomas Simon, Jason Saragih, and Yaser Sheikh. 2020. The Eyes Have It: An Integrated Eye and Face Model for Photorealistic Facial Animation. *ACM Trans. Graph.* 39, 4, Article 91 (7 2020), 15 pages. https://doi.org/10.1145/3386569.3392493

[38] Abigail J Sellen. 1995. Remote conversations: The effects of mediating talk with technology. *Human-computer interaction* 10, 4 (1995), 401–444.

[39] Unity Technologies. 2022. Unity Real-Time Development Platform. https://unity.com/ Accessed: 2022-03-14.

[40] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gerard Medioni. 2017. Extreme 3D Face Reconstruction: Seeing Through Occlusions. https://doi.org/10.48550/ARXIV.1712.05083

[41] Varjo. 2022. Varjo Aero. https://varjo.com/products/aero/ Accessed: 2022-03-14.

[42] VIVE. 2022. VIVE Eye and Facial Tracking SDK. https://developer-express.vive.com/resources/vive-sense/eye-and-facial-tracking-sdk/overview/ Accessed: 2022-03-14.

[43] VIVE. 2022. VIVE Pro Eye. https://www.vive.com/us/product/vive-pro-eye/overview/ Accessed: 2022-03-14.

[44] Miao Wang, Guo-Ye Yang, Ruilong Li, Run-Ze Liang, Song-Hai Zhang, Peter. M. Hall, and Shi-Min Hu. 2019. Example-Guided Style-Consistent Image Synthesis From Semantic Labeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 1495–1504. https://doi.org/10.1109/CVPR.2019.00159

[45] Shih-En Wei, Jason Saragih, Tomas Simon, Adam W. Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. 2019. VR Facial Animation via Multiview Image Translation. *ACM Trans. Graph.* 38, 4, Article 67 (7 2019), 16 pages. https://doi.org/10.1145/3306346.3323030

[46] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers Inc., Seattle, Online, USA, 598–607. https://doi.org/10.1109/CVPR42600.2020.00068

[47] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. 2019. Free-Form Image Inpainting With Gated Convolution. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Institute of Electrical and Electronics Engineers Inc., Seoul, Korea, 4470–4479. https://doi.org/10.1109/ICCV.2019.00457

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. Institute of Electrical and Electronics Engineers Inc., Venice, Italy, 2242–2251. https://doi.org/10.1109/ICCV.2017.244