

A PROCESS FOR THE SEMI-AUTOMATED GENERATION OF LIFE-SIZED, INTERACTIVE 3D CHARACTER MODELS FOR HOLOGRAPHIC PROJECTION

Xinyu Huang¹⁾, John Twycross²⁾, Fridolin Wild¹⁾

¹⁾ Performance Augmentation Lab
Oxford Brookes University
Oxford, UK
{18013560,wild}@brookes.ac.uk

²⁾ Institute of Education
UCL, UK
j.twycross@ucl.ac.uk

ABSTRACT

By mixing digital data into the real world, Augmented Reality (AR) can deliver potent immersive and interactive experience to its users. In many application contexts, this requires the capability to deploy animated, high fidelity 3D character models. In this paper, we propose a novel approach to efficiently transform – using 3D scanning – an actor to a photorealistic, animated character. This generated 3D assistant must be able to move to perform recorded motion capture data, and it must be able to generate dialogue with lip sync to naturally interact with the users. The approach we propose for creating these virtual AR assistants utilizes photogrammetric scanning, motion capture, and free viewpoint video for their integration in Unity. We deploy the Occipital Structure sensor to acquire static high-resolution textured surfaces, and a Vicon motion capture system to track series of movements. The proposed capturing process consists of the steps scanning, reconstruction with Wrap 3 and Maya, editing texture maps to reduce artefacts with Photoshop, and rigging with Maya and Motion Builder to render the models fit for animation and lip-sync using LipSyncPro. We test the approach in Unity by scanning two human models with 23 captured animations each. Our findings indicate that the major factors affecting the result quality are environment setup, lighting, and processing constraints.

Index Terms— 3D scanning, reconstruction, motion capture, Augmented Reality

1. INTRODUCTION

Augmented Reality (AR) enhances the user’s perception of reality. In an AR system, interactive virtual objects are placed in the real world to enhance the visual experience, and help users understand and achieve tasks in a natural way [1]. The quality of the interaction with the virtual elements is one of the main features determining the degree of immersion in AR [2]. Innovative user interfaces beyond mouse and

keyboard are still posing one of the top four challenges to the field, see Azuma [3]. Reactive 3D character models for holographic projection on Smart Glasses or spatially-aligned projectors (think ‘holographic AIs’ or ‘humanoid AR assistants’), generated with the process we propose in this paper, could well be the answer to that problem. Clearly, deficient character models and animations would affect negatively the user experience.

With the development of non-contact sensor technology [4], 3D scanners can capture efficiently shape and surface of human body in situ, providing less resource intensive means for character diversification and individuation. The Occipital Structure sensor is one such example of portable 3D scanning equipment, which allows to automatically capture meshes and textures, additionally with the possibility to gather quick feedback on the quality of the acquired models on an iPad within minutes. We complement this static scan with motion capture, a technique for tracking body-attached markers to record movement and extract data on corresponding dynamic joints behaviour.

There are, however, practical limitations that can impede significantly on the quality of the resulting 3D avatars: scanning environment and background, scan time, actor pose type (A-pose or T-pose) used. Balanced and natural lighting is an important factor, since too strong or too dark lighting can lead to overexposure, blurred or even distorted results. Problems emerging can be remediated in various ways. It is inevitable, however, that the reconstructed models with animation differ to a degree from the actors’ original appearance. The process we document in this paper serves to identify the sweet spot between efforts required and quality results achieved, putting emphasis on eliminating distortion when reproducing the actors’ performance on a pair of Smart Glasses (Microsoft HoloLens). Fig.1 shows three examples of such generated 3D characters in Unity. Each avatar is a full body model, the number of polygons is optimised for running on the HoloLens.

In this paper, we contribute a new process model for how to generate these 3D character models to then test it with a

case study using two models and 23 animations each to verify the approach, document experience, and extract lessons learned. The process model has three main advantages over existing approaches (cf. also Section 2): 3D scanning provides high resolution textures and meshes; a semi-automated method for reconstruction retains better the appearance of the actor; we add a method for facial expressions and motion capture for full body animation.

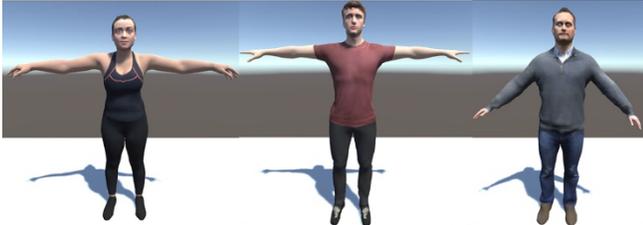


Fig.1. Reconstructed avatars.

The rest of this paper is organized as follows: Section 2 introduces related work in the field. Section 3 elaborates on the workflows, including 3D scanning, reconstruction, and animation. Section 4 presents experimental results. Section 5 concludes the paper and presents an outlook on future research.

2. RELATED WORK

The past few years have seen the application of 3D scanning in many fields, including – amongst others – fashion design, entertainment, or the digital preservation of cultural relics.

Lim et al. propose a system that can create avatars with fixed animations in few minutes by a depth camera [5]. They use a template model to transform and fit the actual 3D scanned model, and then attach the pre-set animation of the template model to the reconstructed 3D avatar. Peng et al. review several methods of obtaining virtual human shape. For instance, human body scanning techniques including a reconstruction facility [6] which can alter the shape of 3D human models to fit target models by taking a series of multi-view colour photos from the front, side, and back. They also review a full-body cloning method for lower lighting conditions to define and produce a virtual human from images using an algorithm. Silhouette extraction is a very important step as edges of the graph contain abundant data to display image features such as textures, colours, or greyscales. They summarize five methods for contour extraction, including methods using prior knowledge, a gradient-based way, level set approaches, and an active contour approach. Han et al. focus on a topological high-resolution template model to fit captured point clouds [7]. They create an environment to scan the full body using multiple depth cameras, set up redundantly with overlapping points. They provide a novel pipeline and algorithm to address artefacts, patch holes, and interpolate missing surface details. The

system is able to automatically generate satisfying reconstructed avatars.

Currently, although depth cameras can be used to capture dynamic point clouds to reconstruct animated meshes, still more than eight cameras are needed for good results [8] and, consequently, time needed for data cleaning and editing is long. For example, Collet et al. propose an end-to-end system for realistic and robust free-viewpoint videos using 106 cameras for image-based motion capture [9]. 3D posture and facial expressions are recovered using a 2D graph. These colour images, however, cannot be directly used for the character animations. Worse still, complex algorithms and big data are needed to recover missing point clouds in the reconstruction. Besides, although depth images are allowed to capture human joints, what is more important is the distances between depth cameras that are required to be calculated in advance to avoid occlusion [8].

In motion capture, Mocap, more mature and efficient technology is used to achieve more precise animation. Vicon cameras, for example, are used in recording and capturing motion in real-time. For natural motion capture, there are three sub processes: preparation, tracking, and post-processing. In this regard, Onder et al. propose a way for keyframe reduction of Mocap data [10]. They apply curve fitting algorithms to simplify data and reduce noise. Lu et al. present a framework to handle motion data and output animation in real-time [11]. They apply a two-stage coarse-to-fine analysis to fix missing data points and correct noisy markers. In terms of facial animation, Kouadio et al. create an animation system to capture the actor’s facial performance in real-time [12]. It relies on markers on the actor’s face to match the points in a model, and a linear combination is then calculated by the minimum gaps between markers and corresponding points. Lewis and Anjyo introduce a direct-manipulation method to edit facial expressions [13]. They mainly employ a blend shape system and inverse-kinematic knowledge to directly control animations. Moura et al. also focus on blend shapes to animate lip movements [14]. They offer an algorithm to synchronize lip animation with the sound based on Adobe After Effects and MEL script. We refer to these concepts of constructing as a template to reproduce a low-poly avatar with a series of animations, and the blend shape system applies to construct facial animations.

Many of the state-of-the-art methods listed here are not in production, lacking interfaces to establish interoperability with the standard editing tool chain. Moreover, where more accurate scanning results are being obtained, hundreds of (depth) cameras are needed in a professional environment with balanced lighting and green screens. Yet, deep-pocketed investment of this sort is not necessarily very suitable for individuals or small groups. We therefore propose to split the recording process, with the character model being created first using 3D scanning (1) and reconstruction (2) to then use independently recorded, re-usable mocap animation movement (3), complemented with the lip sync – with an

acceptable reduction in quality being a favourable trade-off for otherwise high resource intensity.

3. PROPOSED METHOD AND WORKFLOW

Fig.2 shows there are three main steps in this process: 3D scanning, reconstruction, and animation. The scanning environment is a spacious room with five long lights on the ceiling. Although all lights are on, some parts of the body, such as nose and neck, still produce shadows. To remediate this problem, we use an extra LED light attached to the iPad to ensure balanced brightness of the actor’s face. Considering actors’ height, scanning time, and characteristics of the head part, we separately scan and reproduce head and body.

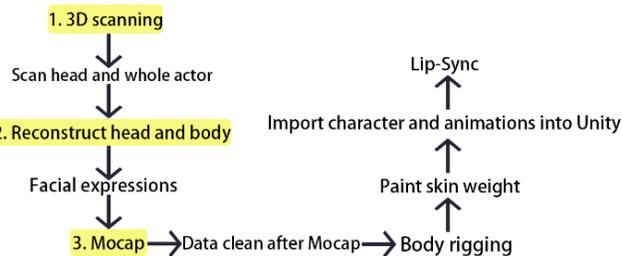


Fig.2. The process

3.1 3D Scanning



Fig.3. ItSeez3D app and Structure sensor capturing body model of person.

In Fig.3 ItSeez3D is an iOS application for 3D scanning, and it needs Internet access since the point cloud has to be processed in the cloud. There are three options for scanning, including body, bust, and object scans. The actors maintain a T-pose i.e. with arms extended out horizontally, for the duration of the scan, since a T-pose should also be captured at the beginning of the Mocap. The actor has to maintain the pose throughout the scanning process, which lasts between a minute and two.

Fig.4 shows that each 3D scanned model shares similar artefacts in the hair part in the head scan, with false colours seemingly derived from clothing or other objects, such as a chair or a reflection from the wall. With the actor maintaining the same position, we scan the whole actor from head to toe

to compare the two scanning results of the head. Fig.5 shows that data of the head top are missing, meshes around the ears mix with the mesh of the head, both sides of facial textures are distorted, and the resolution is lower. Although all fingers are stretched, the sensor cannot recognize and capture all of them in full. Fig.6 illustrates how data of the hands often disappear, and redundant meshes of fingers mash together. From the scans performed, we select the 3D scan models with the best quality for the next step, reconstruction.

We recommend to scan two good quality full-body models, checking the quality with the cloud render preview, rather than the local preview in ItSeez3D. Additionally, we recommend to collect two good quality bust scans for the head and hair. Hands and feet can be downloaded from Wrap 3 or other websites, because the scanning we propose to use typically cannot provide good enough quality for these (whether that is a problem of the actors not holding still long enough, or the sensor, is, however, not determined).



Fig.4. We scan the actor three times, and the shape of his ear is clear.



Fig.5. The result has more artefacts and distorted meshes for the ears, if the actor is scanned with the full body mode.



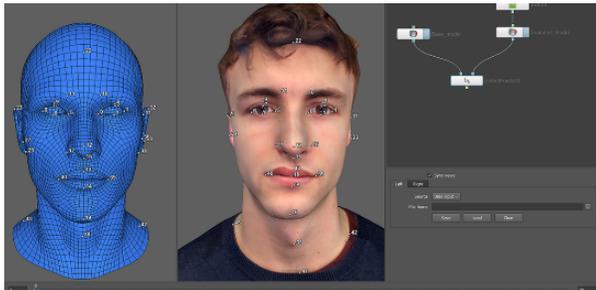
Fig.6. Scanning errors and artefacts of actor’s hands.

3.2 Reconstruction

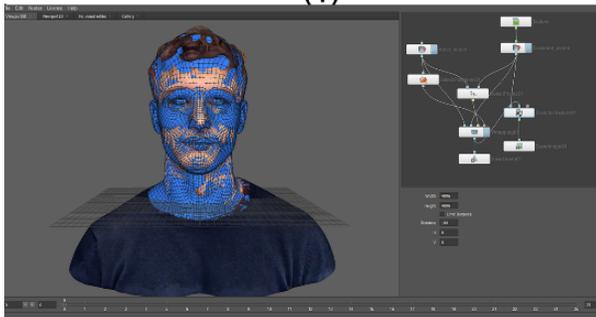
Errors such as false colours and distorted hands can be ignored during the 3D scan, since Photoshop can be employed to fix these artefacts. For reconstruction, we recommend to use Wrap 3 [15] to sanitize 3D scan models and textures.

Wrap 3 offers different types of low-poly models with corresponding UV maps (= the 3D model unfolded to 2D map)

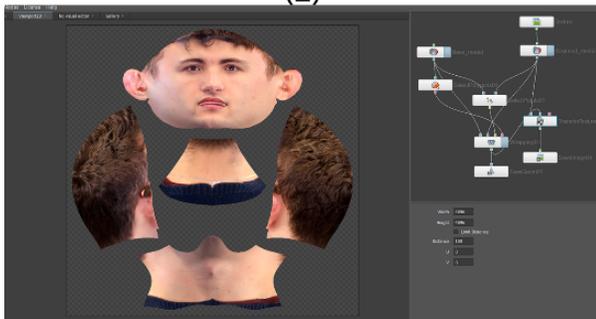
as templates for the transformation, which contain a human body, different shapes of hands, and a head. Those templates should fit the shapes of 3D scan models, and both of them should share similar size, nose, eyes, shoulders, and feet among other main positions corresponding to templates' ones. Only in this way, we can reproduce high-poly models, and also generate high-resolution textures by working with nodes. Fig.7 displays the steps needed for head reconstruction in Wrap 3: Fig.7(1) shows how the markers on the template model are placed consistently with the corresponding ones on the scanned model using the SelectPoints tool. The 'wrapping' node in Fig.7(2) applies the transformation of the captured model to the low-poly model by fitting the template, and the 'TransferTexture' node is used to automatically produce the texture. In this process, we manage to separately reconstruct models of head and full-body mesh. The reconstructed hands are distorted, the texture of hands is full of incorrect colours, and even the shapes of the feet depicted in Fig.8 shows distortion. Those body parts are to be addressed in the next stage of the reconstruction.



(1)



(2)



The texture of head

Fig.7. Reconstructing the head model in Wrap 3.

Fig.9 depicts how the transformed template consists of thousands of independent quadrangles, which suggests that they are not aligned to each other as a whole. When importing into Maya, however, all models exported from Wrap 3 should have all points merged together. Subsequently, in the reconstructed human body model, the meshes that are out of scale should be substituted accordingly with the 'raw' scan data.

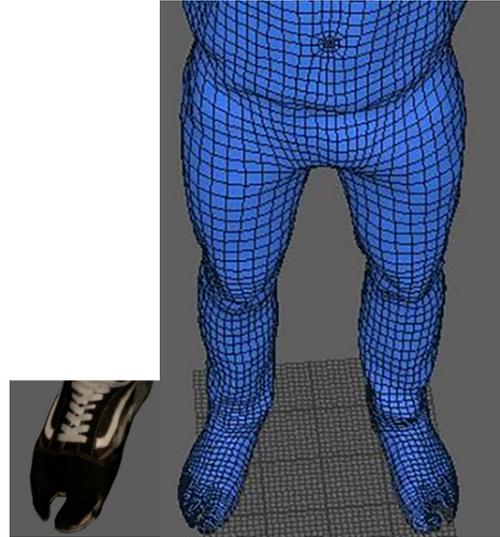


Fig.8. The template for the full body model is naked, which means that meshes of toes cannot be transformed properly on actors wearing shoes.

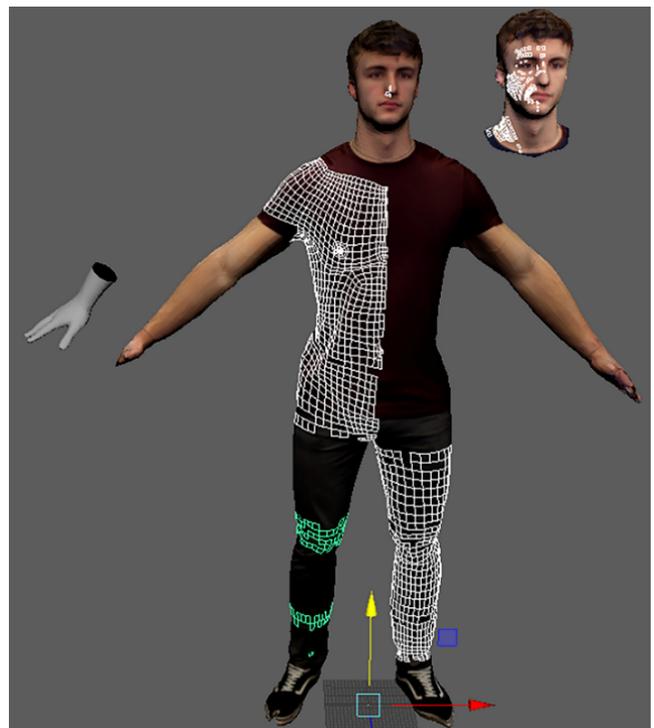


Fig.9. The reconstructed body model, head model, and the newly created hand model.

For example, typically head, hands, and feet may have to be substituted by the individual head and the newly created models of hands and shoes. Fig.8 also shows that there are some unnecessary edges, such as navel, which should be eliminated to reduce the number of polycounts as much as possible. In general, all meshes should be smooth and unfolded. Incorrect colours in textures are fixed with Photoshop by stamp copying over the correct ones, and other textures of human body are applied to make up missing data as well as add more detail. According to the requirement of Mocap, the A-pose (with arms extended down at 45°) in Fig.9 should be changed to the T-pose by rigging. Although Wrap 3 helps produce an integrated texture, the big problem is that the size is smaller than that of the UV map. Consequently, matching neighbouring parts of the mesh in the texture is time consuming since they are not in adjacent positions in the UV layout.

Additionally, the final 3D AR assistants also have affective and communication capabilities, and, accordingly, realistic teeth, tongue, eyes, and eyelashes have to be created to make the character model speak and blink. Each eye includes the models of conjunctive, cornea, lens, and iris, which enables the eyes to generate natural highlights and reflections during rendering.

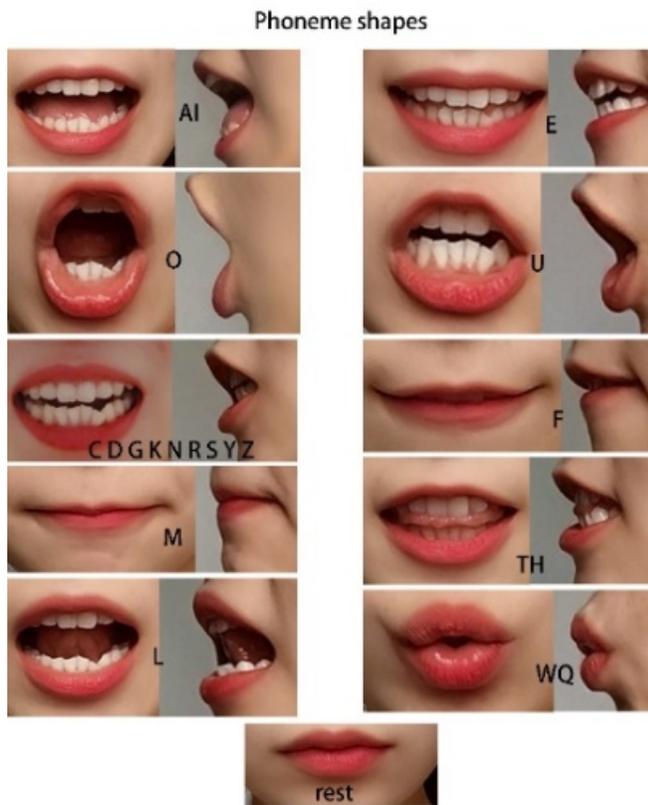


Fig.10. Reference of phonemes

Blend shapes in Maya are employed to generate affect changes. 13 main phonemes are constructed such as AI, E, O,

U, F, L, M, TH, and WQ, based on Fig.10. The smoothing tool is primarily used to separate the lip, and the grab tool serves to change the shape of the mesh. Considering that the surrounding muscles of the mouth differ during speaking, the shapes of chin and masseter also ought to produce detailed transformations. Finally, Fig.11 shows that the final rigged avatar has 20,197 polygons and two textures, including a transparent texture for the parts of eyes and eyelashes as well as a texture for body and head. Both of textures are 4096 by 4096 pixels to ensure good quality of resolution. The exported format is fbx as it can conserve the facial expressions and animation.

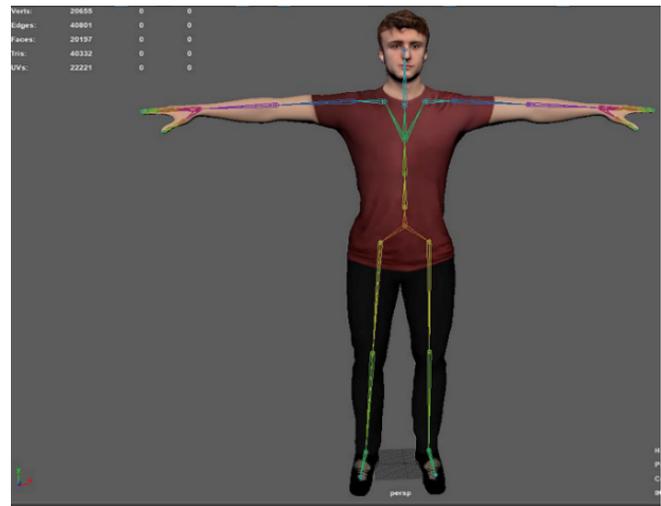


Fig.11. Information of the final male model on the top left.

3.3 Animations

Before being recorded, the actor needs to wear a black suit with 62 markers labelling it. All cameras must be warmed up. It takes four to five hours to prepare for the exporting of these animations.

Most animations are natural and smooth, but some are tainted with movements of fingers on the wrong trajectory in some keyframes. For example, we recorded a rehabilitative sports exercise called chair dip in which the actor needs to place his hands on the edge of a chair. During the process, each finger has motion, and the two thumbs cannot be captured as they cover each other. Therefore, the method uses adjacent correct keyframes to replace the false ones, and adjust the orientation slightly and reasonably to link the next keyframe to simulate a possible trajectory of motion. In this stage it is very challenging to simulate real states with a short length of non-smooth movements. Although there are 23 animated performances, only the T-pose character model needs painted skin weights (= the smoothing factors of the animation, applied using a paint brush tool).

Finally, the 23 animations in Fig.12 and the original one with facial expressions can be imported into Unity.

LipSync pro 1.5 [16] in Fig.13 is a plugin to edit facial animations, which is capable of supporting a blend system to reset those facial expressions created previously, and also generating corresponding phonemes from an audio source. It also offers nine phonemes, four emotions, and an eye controller to set eye blink rate and eyeball movement. The speaker's mood state, different emotions, represented in facial expressions such as smile, sadness, raised eyebrow or frowning, can be added to simulate emotional expressions. However, if the avatar also has eyeball movements controlled by the blend shape, such as looking up and down, the two eyeballs are about to move around in sync with setting of phonemes.



Fig.12. The 23 animations and character are imported into Unity.

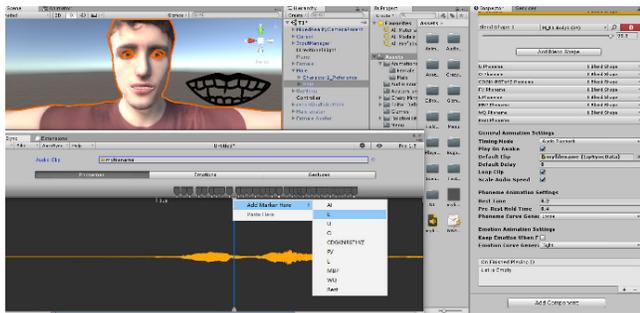


Fig.13. LipSync.

Therefore, in our set up, this is not necessary and we do not create this kind of animations. The eye controller is set to random looking. LipSync allows to add markers of phonemes based on audio source by hand, or automatically generate corresponding phonemes using the integrated SoX sound exchange. Additionally, we deploy three scripts, including a lip script, a record audio script in Fig. 13, and save audio script, that can directly record our voice in Unity as well as produce a supported audio format in the Unity assets. To be more specific, we use a UI to record the voice, stop capture, and check the sound. The lip script then connects to the plugin and saves the audio script, so that it can find the audio source. Subsequently, the character model automatically generates the lip animation. Fig.14 shows how to record the sound.

```

Debug.Log("starting");

Microphone.GetDeviceCaps(null, out minFreq, out maxFreq);
if (minFreq == 0 && maxFreq == 0)
{
    maxFreq = 44100;
}

if (!Microphone.IsRecording(null)) {
    goAudioSource.clip = Microphone.Start(null, false, 120, maxFreq);
}

```

Fig.14. The script of capturing the sound.

4. CASE STUDY: RESULTS AND ANALYSIS

The quality of reconstructed models is defined by how much data is successfully scanned, and that it has limited artefacts in textures, and no distorted or missing meshes. Tiny artefacts are tolerable, if the system is able to make up via adjacent pixels. Clean background and balanced lighting are crucial to obtain a high-quality texture with few artefacts, while colours of objects near the captured actor can bleed into the 3D scan.

For instance, it is possible that the colours of clothes or a chair can protrude onto skin or hair, even though those details can be addressed in the post-processing by editing the texture map. Capturing hands is the most difficult part in the full-body scan, as the meshes of fingers often are missing detail or merge together. The easier way is that the newly created model of hands replaces distorted ones, and the textures are drawn based on the colour of skins, rather than sculpting the scanned model. We suggest to not scan too many times in a row with the Structure sensor, as both sensor and actors tire. It is hard to keep a T-pose precisely for a longer time.

The reconstructed head with more precise meshes and texture represents actor's appearances and facial animation. Some polygons of clothes and trousers overlap after transformation, especially around elbows and knees, and those two parts are important in movement of joints. The smoothing tool in Maya is very helpful to unfold these. The template in Wrap 3 cannot fully reconstruct the scanned model, and automatically correct false meshes in one go. Shoes and hands have to be rebuilt separately, and then combined together. This kind of 3D reconstructed character is different from traditional human-like ones used in games. In particular, it is not advantageous to directly bake normal maps and ambient occlusion (AO) maps from 3D scan models, because it just can show a pure blue image for the normal map and a false white picture for AO. The reason is the 3D scan models can only generate very simple colour material to describe the basic and flat textures of clothes and skin without being able to capture tiny convex or concave details, such as pimples or different materials of clothes.



Fig.15. The final animated 3D character model in Unity.

Fig.15 shows an example of the final avatar with three animations in Unity, where the fingers and legs show flexible movements, and more than 20,000 polygons work very fluently in Unity. Those animations can accurately represent the actor's performances. Although the reconstructed model is similar to the real actor, hair and clothes are not reproduced. As a result, there is not any movement during the rendering, which prevents high fidelity. By copying the right keyframes we were able to compensate and achieve natural animation for the parts with incomplete motion capture. This, however, at times resulted in non-smooth arm movement. The most unnatural and unrealistic place are the ampits, which are slightly distorted when arms move.

The reason is that clothes and body are integrated into one model, and painted skin weights cannot prevent distortion of parts of the cuff during ampit movement. Moreover, we can conclude that it is difficult for 12 Vicon cameras to capture high precision exercises, especially where the arms are on the back and the hands cover some markers on shoulders. LipSync offers nine types of phonemes, including AI, E, U, O, CDGKNSTHYA, FV, L, MBP and WQ, and also each type can add multiple phonemes. It is sufficient for lip animation, while each type still has different viseme, such as F and V.

The scripts we created are able to support LipSync in order to automatically generate correct and corresponding phonemes according to recorded audio sources, yet it is not reliable and it is incomplete since missing data has to be added. Facial animation is one of the important parts in interaction, and simulative phonemes and emotions are flexible to represent some fixed expressions based on different tones. However, it is not natural, because when people talk, their skin can generate wrinkles, even if these are difficult to observe from the distance.

To sum up, this character model with clean meshes, high-resolution textures and smooth animations, is able to ensure its verisimilitude and interactivity, and an audio recorder script is convenient to generate the required audio source for Unity. The resulting models are realistic, but not fully identical with the original, as a result of reconstruction and editing. There are issues about fidelity in animation as some markers possibly cover each other, which leads to false or missing trajectories in those key rigs. Although these can be fixed, some key joints can affect the children's movements, which is similar to the relationship between shoulders and arms.

5. CONCLUSION

This study has presented a process pipeline and methods for 3D scanning, reconstruction, and motion capture serving the purpose to reproduce a realistic and interactive 3D holographic assistant in short time and with – in AR, media, or film making labs – widely available standard hardware and software. Our methods combine effective existing applications into an efficient workflow.

In the approach proposed in this paper, there is no need to set up a particular environment for 3D scanning, and this paper has elaborated on how to directly apply 3D scanning to achieve a low-poly model. 23 sports animations have been captured which can be applied to free-viewpoint video in AR. The approach proposed do not require prior knowledge of the 3D scan and reconstruction to clean artefacts and data of motion capture.

The case study offers evidence of generating high quality interactive 3D character models that work well on smart glasses like the Microsoft HoloLens. In the future, we also aspire to utilise additional equipment, such as a small array of Intel Realsense cameras and Kinects, to compare this approach with volumetric video capture for 3D reconstruction and recognition.

ACKNOWLEDGMENT

We would like to thank the members of our interdisciplinary cancer research group, Peter Wright, Jo Brett, Emma Davies, Will Guest, Ryan Pink, Eila Watson, Stan Windsor, and, last but not least, Arran Gulliver who is the main actor shown in the pictures of the scans, and Mikhail Fominykh for the photo in Fig.3. This work was supported by the European Commission under the Horizon 2020 programme (H2020), as part of ARETE (grant agreement no. 856533).

REFERENCES

- [1] M. Khedwala, F. Monmin, U. Pachhapure and S. Shaikh, "Analysis of Auto Generation of 3D Model Using Multiple 2D graphics to Manifest Through Augmented Reality," in *ICSCET conference*, Mumbai India, 2018, pp.1-5.
- [2] H. Park, S. Lee, J. Choi, "Wearable augmented reality system using gaze interaction," in *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, Cambridge UK, 2008, pp.175-176.
- [3] R. Azuma, "The Most Important Challenge Facing Augmented Reality", In: *Presence* 25(3):234-238.
- [4] X. Huang, S. Sun, "A rapid surface remodeling of human body based on scanned data," In *Fourth International Conference on Computational and Information Sciences*, Chongqing China, 2012, pp.227-230.
- [5] H. Lim. (2018, Aug. 30). Rapid 3D Avatar Creation System. Online: <https://www.youtube.com/watch?v=aKe4ntxjqYM>

- [6] H. Peng, X. Shang, C. Guo, G. Xiong, T. Nyberg, D. Fan, "Survey on Big Data for Human Body Shape," in *SOLI Conference*, Beijing China, 2016, pp.145-150.
- [7] Z. Liu, J. Huang, S. Bu, J.Han, X. Tang, X. Li, "Template Deformation-Based 3-D Reconstruction of Full Human Body Scans From Low-Cost Depth Cameras," in *IEEE Transactions on Cybernetics*, 2016, pp.695-708.
- [8] H. Li. (2018, Aug. 18). Deep Volumetric Video from Very Sparse Multi-View Performance Capture, *ECCV 2018* Online: <https://www.youtube.com/watch?v=sQnXIQ3GyKc>
- [9] A. Collect, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, S. Sullivan, "High quality streamable free-viewpoint video," In: *ACM Transactions on Graphics*, 2015, 34(4).
- [10] O. Onder, U. Gudukbay, B. Ozguc, T. Erdem, C. Erdem, M. Ozkan, "Keyframe Reduction Techniques for Motion Capture Data," in *3DTV Conference*, Istanbul Turkey, 2008, pp.239-296.
- [11] G. Lu, Y. Li, S. Jin, Y. Zhang, W. Chen, X. Zheng, "A real-time motion capture framework for synchronized neural decoding" in *IEEE International Symposium on VR Innovation*, Singapore, 2011.
- [12] C. Kouadio, P. Poulin, P. Lachapelle, "Real-time facial animation based upon a bank of 3D facial expressions", in *Computer Animation*, 1998.
- [13] J. Lewis, K. Aniyo, "Direct Manipulation Blendshapes", in *IEEE Computer Graphics and Applications*, 2010, pp.42-50.
- [14] A. Moura, I. Mazonaviciute, J. Nunes, J. Grigara, "Human lips synchronisation in Autodesk Maya", in *14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, Maribor, Slovenia, 2007.
- [15] Wrap 3. (2018). Wrap 3 Documentation, Online: <https://www.russian3dscanner.com/docs/Wrap3/>
- [16] Unity. (2018). LipSync Pro, Online: <https://assetstore.unity.com/packages/tools/animation/lipsync-pro-32117>