frontiers in Virtual Reality

ORIGINAL RESEARCH
published: 01 November 2021
doi: 10.3389/frvir.2021.739010

Deep4D: A Compact Generative Representation for Volumetric Video

João Regateiro[1,2]*, *Marco Volino*[1] *and Adrian Hilton*[1]

[1]Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, United Kingdom, [2]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), LJK, Grenoble, France

This paper introduces Deep4D a compact generative representation of shape and appearance from captured 4D volumetric video sequences of people. 4D volumetric video achieves highly realistic reproduction, replay and free-viewpoint rendering of actor performance from multiple view video acquisition systems. A deep generative network is trained on 4D video sequences of an actor performing multiple motions to learn a generative model of the dynamic shape and appearance. We demonstrate the proposed generative model can provide a compact encoded representation capable of high-quality synthesis of 4D volumetric video with two orders of magnitude compression. A variational encoder-decoder network is employed to learn an encoded latent space that maps from 3D skeletal pose to 4D shape and appearance. This enables high-quality 4D volumetric video synthesis to be driven by skeletal motion, including skeletal motion capture data. This encoded latent space supports the representation of multiple sequences with dynamic interpolation to transition between motions. Therefore we introduce Deep4D motion graphs, a direct application of the proposed generative representation. Deep4D motion graphs allow real-tiome interactive character animation whilst preserving the plausible realism of movement and appearance from the captured volumetric video. Deep4D motion graphs implicitly combine multiple captured motions from a unified representation for character animation from volumetric video, allowing novel character movements to be generated with dynamic shape and appearance detail.

Keywords: volumetric video, generative networks, motion graphs, animation, performance capture

OPEN ACCESS

Edited by:
Fabien Danieau,
InterDigital, France

Reviewed by:
Anna Hilsmann,
Heinrich Hertz Institute (FHG),
Germany
Weiya Chen,
Huazhong University of Science and
Technology, China

****Correspondence:***
João Regateiro
j.regateiro@inria.fr

Specialty section:
This article was submitted to
Technologies for VR,
a section of the journal
Frontiers in Virtual Reality

Received: *09 July 2021*
Accepted: *20 September 2021*
Published: *01 November 2021*

Citation:
Regateiro J, Volino M and Hilton A
(2021) Deep4D: A Compact
Generative Representation for
Volumetric Video.
Front. Virtual Real. 2:739010.
doi: 10.3389/frvir.2021.739010

1 INTRODUCTION

Volumetric video is an emerging media that allows free-viewpoint rendering and replay of dynamic scenes with the visual quality approaching that of the of captured video. This has the potential to allow highly-realistic content production for immersive virtual and augmented reality experiences. Volumetric video is produced from multiple camera performance capture studios that generally consist of synchronised cameras that simultaneously record a performance (Collet et al., 2015; Starck and Hilton, 2007; de Aguiar et al., 2008; Carranza et al., 2003). The generated content usually consists of 4D dynamic mesh and texture sequences that represent the visual features of the scene, for example, shape, motion and appearance. This allows replay of the performance from any viewpoint and moment in time, although it requires a huge computational effort to process and store. Volumetric video capture is currently limited to replay of the captured performance and does not support animation to modify, combine or generate novel movement sequences. Previous work has introduced methods for animation from volumetric video based on re-sampling and concatenation of volumetric sequences (Huang et al., 2015; Prada et al., 2016).

Rendering realistic human appearance is a particularly challenging problem. Humans are social animals that have evolved to read emotions through body language and facial expressions (Ekman, 1980). As a result, humans are extremely sensitive to movement and rendering artefacts, which gives rise to the well-known uncanny valley in photo-realistic rendering of human appearance. Recently there has been significant progress using deep generative models to synthesise highly realistic images (Goodfellow et al., 2014; Kingma and Welling, 2013; Zhu et al., 2017; Isola et al., 2016; Ulyanov et al., 2016; Ma et al., 2017; Siarohin et al., 2017; Paier et al., 2020) and videos (Vondrick et al., 2016; Tulyakov et al., 2017) of scenes, which is important for applications such as image manipulation, video animation and rendering of virtual environments. Human avatars are typically rendered using detailed, explicit 3D models, which consist of meshes and textures, and animated using tailored motion models to simulate human behaviour and activity.

Recent work Holden et al. (2017) has shown that it is possible to learn and animate natural human behaviour (e.g. walking, jumping, etc.) from human skeletal motion capture data (MoCap) of actor performance. On the other hand, designing a realistic 3D model of a person is still a laborious process. Given the tremendous success of deep generative models (Goodfellow et al., 2014; Kingma and Welling, 2013; Zhu et al., 2016; Karras et al., 2017; Isola et al., 2016), the question arises, why not also learn to generate realistic rendering of a person? By conditioning the image generation process of a generative model on additional input data, mappings between different data domains are learned (Zhu et al., 2017; Isola et al., 2016; Johnson et al., 2016), which, for instance, allows for controlling and manipulating object shape, turning sketches into images and images into paintings. Generative methods have improved recently on the resolution and quality of images produced (Karras et al., 2017; Miyato et al., 2018 Brock et al., 2018). Yet generators continue to operate as black boxes, and despite recent efforts, the understanding of various aspects of the image synthesis process is unknown. The properties of the latent space are also poorly understood, and the commonly demonstrated latent space interpolation (Dosovitskiy et al., 2015; Sainburg et al., 2018; Laine, 2018) provide no quantitative way to compare different generators against each other. Motivated by recent advances in generative networks (Karras et al., 2018; Karras et al., 2017; Goodfellow et al., 2014) we propose an architecture for learning to generate dynamic 4D shape and high resolution appearance that exposes ways to control image synthesis. Our appearance generator starts from a learned motion space and adjusts the resolution of the image at each convolution layer based on the latent motion code, therefore directly controlling the strength of image features at different scales.

This work proposes Deep4D, a deep generative representation of dynamic shape and appearance from 4D volumetric video of a human character. The proposed approach learns an efficient compressed latent space representation and generative model from 4D volumetric video sequences of a person performing multiple motions. Compact latent space representation is achieved using a variational encoder-decoder to learn the mapping from 3D skeletal motion to the corresponding full 4D volumetric shape, motion and appearance. The encoded latent space supports interpolation of dynamic shape and appearance to seamlessly transition between captured 4D volumetric video sequences. This work presents Deep4D motion graphs, which exploit generative representation of multiple 4D volumetric video sequences in the learnt latent space to enable interactive animation with optimal transition between motions. The primary novel contributions of this paper are:

- Deep4D, a generative shape and appearance representation for 4D volumetric video that enables compact storage and real-time interactive animation.
- Mapping of skeletal motion to 4D volumetric video to synthesise dynamic shape and appearance.
- Deep4D motion graphs, an animation framework built on top of the Deep4D representation that allows high-level of 4D characters enabling synthesis of novel motions and real-time user interaction.

2 RELATED WORK

4D Volumetric Video: has been an active area of research (Starck and Hilton, 2007; Collet et al., 2015; Carranza et al., 2003; de Aguiar et al., 2008), that has emerged to address the increasing demand for realistic content of human performance. Recently, Collet et al. (2015) presented a full pipeline to capture, reconstruct and replay high-quality volumetric video. The system uses approximately 100 synchronised cameras that simultaneously capture the volume from multiple viewpoints. Volumetric video captures the dynamic surface geometry and photo-realistic appearance of a subject. This unlocks enormous creative potential for highly realistic animated content production based on the captured performance. Recent research provides frameworks to ease the manipulation of this content (Huang et al., 2015; Prada et al., 2016; Tejera and Hilton, 2013; Budd et al., 2013; Cagniart et al., 2010; Vlasic et al., 2008; Regateiro et al., 2018; Casas et al., 2014), allowing an artist to perform manual adjustments on 4D dynamic geometry and combine multiple sequences in a motion graph. However, use of 4D volumetric video in content production remains limited due to the challenge of manipulation, animation and rendering of shape sequences whilst maintaining the realism of appearance and clothing dynamics.

Learnt Mesh Sequence Representations: Tejera and Hilton (2013) proposed a part-based spatio-temporal mesh sequence editing technique that learns surface deformation models in Laplacian coordinates. This approach constrains the mesh deformation to plausible surface shapes learnt from a set of examples. Part-based learning of surface deformation allows local manipulation of the mesh and achieves greater animation flexibility, allowing the generation of novel posed meshes. Tan et al. (2018) use a variational autoencoder (VAE) to learn a representation of parameterised dynamic shapes. Their network trains on a pre-processed feature space of the training data, demonstrating very low reconstruction error for the ground truth shapes. Lombardi et al. (2018) proposed a learnt model of shape and appearance conditioned on viewpoint allowing recovery of view-dependent texture detail. This network demonstrates the ability to learn 3D dynamic shapes from vertices, avoiding the need to pre-process information. This demonstrates the real-time capabilities of VAEs, being able to decode shape and appearance in less than 5 milliseconds. Recently, Regateiro et al. (2019) demonstrated the capabilities of learning 3D dynamic shapes to produce realistic animation using a VAE to learn the geometric space of a human character and re-use the decoder in real-time to synthesise 3D geometry.

Learnt Representation of Appearance: Recently, Esser et al. (2019) presented an approach towards a holistic learning framework for rendering human behaviour trained from skeletal motion capture data for realistic control and rendering. They learn a mapping from an abstract pose representation to target images conditioned on a latent representation of a VAE for appearance. Karras et al. (2017) propose a novel training methodology for generative networks. that progressively grows both the generator and discriminator, starting from a low image resolution and ending at the original image resolution. They demonstrate that the model increasingly learns fine details as the training progresses, hence improving training speed and stability, and producing high-quality images. Although photorealism is a hard problem to solve, this approach is a step towards recreating high quality images that are indistinguishable from real images. More recently, Karras et al. (2018) redefine the architecture of generative networks for style-based transfer. Using a similar approach to Karras et al., 2017, they have demonstrated high quality images results, for example, the ability to learn the exact placement of hair, stubble, freckles, or skin pores. This demonstrates the potential to synthesise high resolution images of humans, whilst preserving natural details that are essential for perception of realism.

4D Volumetric Video Animation: Motion graphs for character animation from skeletal motion capture sequences (Arikan et al., 2003; Kovar et al., 2002; Tanco and Hilton, 2000) use a structured graph representation to enable interactive control. The skeletal motion graphs are constructed using a frame-to-frame similarity metric which identifies similar poses and motion. The concept of motion graphs has been applied to volumetric video using both unstructured meshes (Starck et al., 2005; Huang et al., 2009; Hunag et al., 2015; Prada et al., 2016) and temporally consistent structured meshes (Casas et al., 2014; Boukhayma and Boyer, 2017; Hilsmann et al., 2020). Initial approaches (Starck et al., 2005; Huang et al., 2009) concatenate unstructured dynamic mesh sequences without temporal consistency of the mesh connectivity based on shape and motion similarity. Prada et al. (2016) instead performs mesh and texture alignment at defined transitions points to ensure smooth blending. This overcomes the challenging problem of global mesh alignment and only considers alignment of geometry and texture where necessary. In contrast, Boukhayma and Boyer (2017) and Casas et al. (2014) leverage global alignment of the mesh sequence to obtain temporally consistent mesh connectivity from the volumetric video. This allows 4D motion graphs with mesh blending for high-level parametric control of the motion and smooth transitions between motions.

In this paper we introduce Deep4D, a learnt generative representation of volumetric video sequences, presented in **Section 3**. Deep4D provides compact representation, which overcomes the memory and computation requirement of previous approaches to explicitly represent all captured sequences at run-time through the learnt parameters of the network. In **Section 4** we present Deep4D motion graphs, a direct application of the proposed generative network to produce seamless animations of both dynamic shape and appearance between learnt captured motion sequences. Finally, **Section 5** presents a quantitative and qualitative evaluation of the proposed method.

3 DEEP4D REPRESENTATION

The work presents a step forward to allow control and synthesis of 4D volumetric video, while preserving the realism of dynamic shape and appearance. This section introduces the use of a generative network to represent 4D volumetric video content from performance capture data efficiently. Pre-processing of the captured volumetric video into a form suitable for neural networks is first presented. The generative network for the learning of 4D shape from captured volumetric sequences is described, together with the use of a variational encoder-decoder to ensure a compact latent space representation mapping from 3D skeletal pose to corresponding 4D dynamic shape. Finally, we present a generative network for 4D video appearance that learns to synthesise high-resolution dynamic texture appearance from the compact latent space representation, **Figure 1**. Enforcing a compact latent space representation enables interpolation between skeletal poses to generate plausible intermediate mesh shape and appearance. These sections individually describe the contribution of the generative network, illustrated in **Figure 1**. Deep4D generative representation enables the generation of realistic renderings of human characters, with the ability to re-target new skeletal motion information.

3.1 Volumetric Video Pre-processing

In the context of this work, 4D volumetric video represents 4D mesh sequences M_t^s, 2D textures T_t^s and 3D skeletal motion p_t^s computed from multiple view video capture. A 4D volumetric video dataset consists of N_S sequences $s = [1 \dots N_S]$ and each sequence consists of N_T^s frames at a time instance $t = [1 \dots N_T^s]$.

State-of-the-art volumetric performance capture of people with loose clothing and hair (Collet et al., 2015) results in high resolution reconstructed shape and texture appearance. Raw volumetric video typically results in an unstructured mesh sequence where both the mesh shape and connectivity changes from frame-to-frame (Prada et al., 2016). Several approaches have been introduced for temporal alignment over short subsequences to compress the storage requirements (Collet

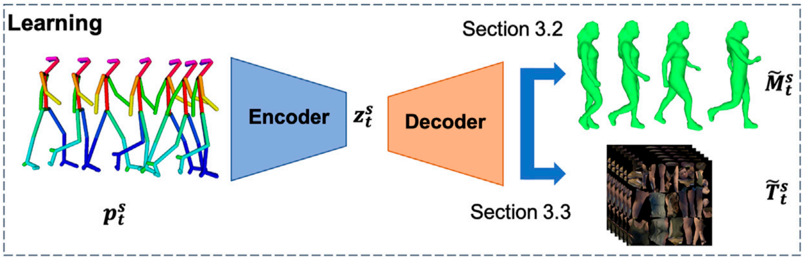

FIGURE 1 | The generative network is driven from 3D skeletal motion to synthesise 4D volumetric video interactively.

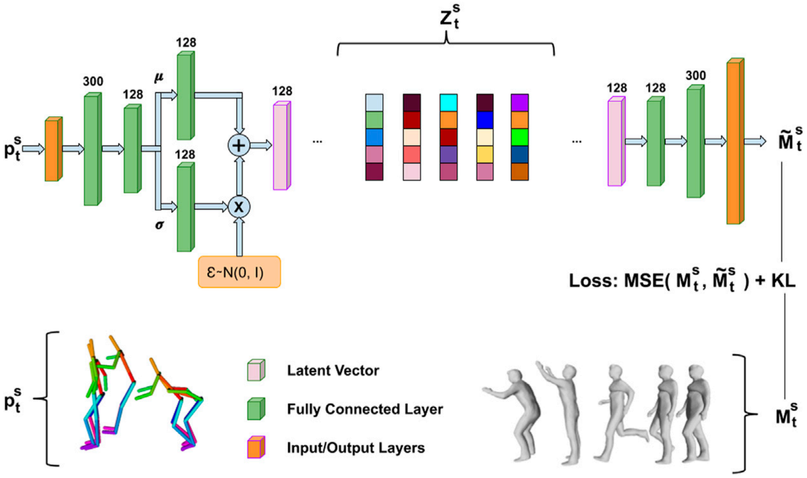

FIGURE 2 | Pose2Shape network overview. The input and the output of the encoder and decoder is 3D skeletal motion and 4D shape respectively.

et al., 2015) or global alignment across complete sequences (Huang et al., 2011; Cagniart et al., 2010; Regateiro et al., 2018).

In this work we employ the skeleton-driven volumetric surface alignment framework (Regateiro et al., 2018) to pre-process captured 4D volumetric video of people to obtain a temporally coherent mesh structure across multiple sequences. This framework receives as input synchronised multiple view video from calibrated cameras and returns 3D skeletal joints and temporally consistent 3D meshes with the same mesh connectivity at every frame. The texture appearance is retrieved by re-mapping the original multiple view camera images onto the temporally consistent 3D meshes providing a dynamic texture map with consistent coordinates for all captured frames. The input to the deep network presented in the following sections consists of centred 4D temporally consistent mesh sequences with the corresponding 2D texture maps and 3D skeletal joint locations.

3.2 Deep4D: Pose2Shape Network

Variational networks have become a popular approach to learn a compact latent space representation which can integrate with deep neural networks. In this section, we employ a variational encoder-decoder to learn a compact latent space mapping between 3D skeletal pose and the corresponding 4D shape, illustrated in **Figure 2**. The generative network architecture maximises the probability distribution of the 3D skeletal joint positions $p = \{\{p_t^s\}_{t=1}^{N_T^s}\}_{s=1}^{N_S}$, encoded in the latent space $z = \{\{z_t^s\}_{t=1}^{N_T^s}\}_{s=1}^{N_S}$, and learns the generative mapping of the decoder to the corresponding 4D mesh $\tilde{M}_t^s$. While we define input p as 3D skeletal joint positions, it can be replaced with other pose representations consisting of 3D landmarks, e.g. facial keypoints.

Generative networks learn dependencies from the input data and capture them in a low-dimensional latent vector z_t^s, creating compact representations $z_t^s \in \mathbb{R}^d$, where d is the latent space dimension (128 dimensions throughout this work). The probability density function $P(p)$ for the skeletal pose is given by:

$$P(p) = \int P(p \mid z)\, P(z) dz \qquad (1)$$

The distribution $P(p|z)$ denotes the maximum likelihood estimation of dependencies of p over the latent vector z, and $P(z)$ is the prior probability distribution of a latent vector z. To ensure a compact representation $P(p|z)$ is modelled as a Gaussian distribution with mean $\mu(z)$ and diagonal co-variance $\sigma(z)$

multiplied by the identity $\boldsymbol{I}$, which implicitly assumes independence between the dimensions of z.

$$P(p \mid z) = N(p \mid \mu(z),\ \sigma(z)^2 * \boldsymbol{I}) \tag{2}$$

The Pose2Shape network architecture is composed of an encoder, which receives 3D skeletal joint positions as input, and a decoder, see supplementary material network details, that generates high resolution 3D meshes. The encoder is trained to map the posterior distribution of data samples p to the latent space z, meanwhile forcing the latent variables z to comply with the prior distribution of $P(z)$. However, both the posterior distribution $P(z|p)$ and $P(p)$ are unknown. Therefore, variational networks give the solution that the posterior distribution is a variational distribution $Q(z_t^s|\tilde{M}_t^s)$. In order to make $Q(z_t^s|\tilde{M}_t^s)$ consistent with the distribution $P(z)$, we use the Kullback-Leiber (KL) divergence (Kingma and Welling, 2013):

$$KL(Q(z_t^s \mid \tilde{M}_t^s) \parallel P(z_t^s)) \tag{3}$$

The decoder is trained to regress from any latent vector z_t^s in the learnt space z to a 4D mesh representation $\tilde{M}_t^s$. **Eq. (4)** defines the loss function minimised by the network to achieve a compact latent space representation and generative network output.

$$L = (Q(P(p_t^s \mid z_t^s) \mid \tilde{M}_t^s) - M_t^s) + \omega KL(Q(z_t^s \mid \tilde{M}_t^s) \parallel P(z_t^s)) \tag{4}$$

This is an optimal approximation of the true samples M_t^s, where ω weighs the importance of the KL divergence, and M_t^s is the ground truth 4D mesh for the 3D skeletal pose p_t^s of sequence s at time t.

3.2.1 Training Details

The network architecture used to regress 3D skeletal pose to 4D mesh shape is summarised in **Figure 2**. The network was empirically found to learn a good latent space distribution with accurate 4D shape generation using a training cycle of 10^4 epochs, which is optimised through validation data to avoid over-fitting with a learning rate of 0.001. The datasets are split by randomly selecting frames from each motion sequence with ≈80% used for training and ≈20% used for validation. We set the prior probability over latent variables to be a Gaussian distribution with zero mean and unit standard variation, $p\ (z) = N\ (z; 0, \boldsymbol{I})$. We use Adam optimisation (Kingma and Welling, 2013) with a momentum of 0.9 to optimise **Eq. (4)** between the reconstructed and ground truth mesh vertices, and simultaneously the KL divergence of the 3D skeletal pose distribution. Evaluation of the performance of the network for shape representation from skeletal pose is given in **Section 5**.

3.3 Deep4D: Pose2Appearance Network

In this section, we propose the use of a Pose2Appearance network for the synthesis of high-resolution dynamic mesh texture maps from the encoded skeletal pose latent space representation. A similar approach described as the progressive growing of GANs was first introduced by Karras et al. (2017) to improve image synthesis quality and training stability of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014).

A GAN consists of two networks, a generator and a discriminator. The generator produces images from a latent code, and the distribution of these images should be indistinguishable from the training distribution. The discriminator evaluates the quality of the images produced by the generator, forcing the generator to learn how to produce high-quality images so that the discriminator cannot tell the difference. A progressive generator generally consists of a network where the training begins with a low-resolution image and progressively increases the resolution until it reaches a target resolution. This incremental multi-resolution approach allows the training first to discover the large-scale structure of the distribution of the images and then shifts the attention to finer-scale details, whereas in traditional GAN architectures, all scales are learned simultaneously.

In this section, we adapt the generator from the progressive growing of GANs (Karras et al., 2017) to learn how to synthesise high-resolution texture appearance from the latent probability distribution learned from 3D skeletal motion, **Section 3.2**. The proposed Pose2Appearance for high-resolution texture map synthesis from the latent space vector is illustrated in **Figure 3**.

The Pose2Appearance initially starts with a small feed-forward network, see supplementary for details, which consists of four fully-connected layers, where the input consists of learnt latent vector z_t^s of dimension 128, which corresponds to the dimensions of the latent space learned from the Pose2Shape network, and the output dimension of the fourth layer is 512, to match the input size requirements of the first convolutional layer, as illustrated in **Figure 3**. The convolutional layers consist of nine blocks, where each block represents a different resolution, and its output is a high-resolution texture $\tilde{T}_t^s$.

We also experimented with a VAE network for appearance synthesis. This experiment was found to result in significant blur and loss of detail. The VAE assumes the same input and output, hence not being a suitable architecture for the problem. For this reason, a more sophisticated network approach is required, **Section 5.3** for comparison with state-of-the-art methods.

3.3.1 Training Details

The Pose2Appearance training starts with a 4 × 4 resolution and progressively grows the network layers until it reaches 1,024 × 1,024 resolution. The network progresses through the training by adding new layers with double the size. There are two stages for training the growing process (**Figure 4**), the first stage is when a new layer is added a fading stage begins where the new layer will be smoothly added to the network. This new layer will operate as a residual block, whose weight τ increases linearly from 0 to 1. When the fading stage is over the second stage is initiated, the stabilising stage, where the new layer is fully integrated with the network, and it iterates over another training cycle. This training pattern repeats until it reaches the full resolution of 1,024 × 1,024. For every stage, we gradually decrease the minibatch size, vary the stabiliser number of training iterations and vary the convergence tolerance. These parameters are necessary to avoid exceeding the available memory budget and decrease the training time.

The generator network is trained using Adam (Kingma and Welling, 2013), with a constant learning rate of 0.001 across the

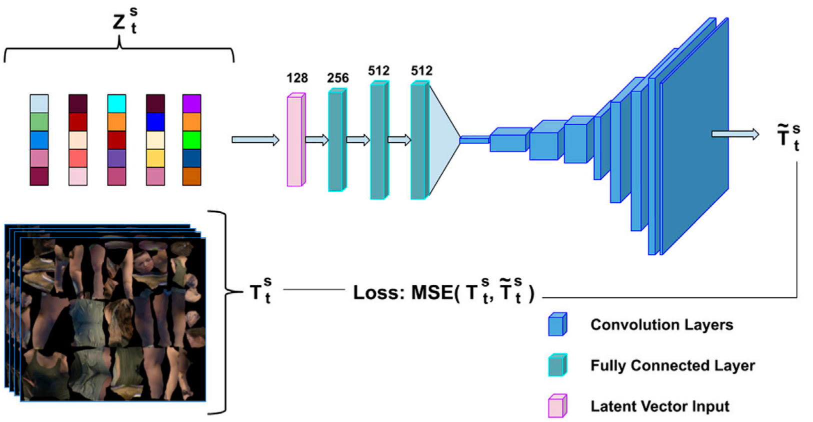

FIGURE 3 | Progressive appearance network overview. The input is learnt latent vectors from 3D skeletal motion, and the output is a high-resolution 2D texture appearance.

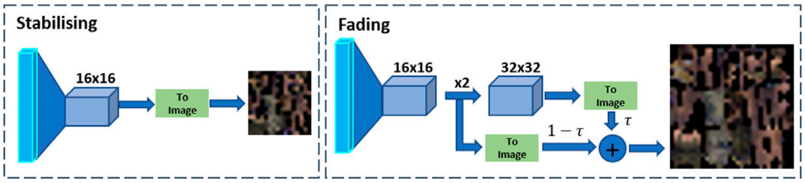

FIGURE 4 | Progressive appearance training overview. Convolutional layers training stages, Stabilising stage with 16 × 16 resolution, and Fading stage to the next 32 × 32 resolution layer.

full training. We use leaky ReLU (Tan et al., 2018) with a leakiness value of 0.2, equalised learning rate for all layers, except the last layer that uses linear activation, and pixel normalisation of the feature vector after each Conv 3 × 3 layer. All weights of the convolutional, fully-connected and affine transform layers are initialised using a Gaussian distribution with zero mean and unit standard variation, $p(z) = N(z; 0, \boldsymbol{I})$. Stochastic gradient descent with a momentum of 0.9 is used to minimise the mean squared error (MSE) loss between reconstructed image $\tilde{T}_t^s$ and the ground truth samples T_t^s.

3.4 4D Volumetric Video Synthesis

The latent space of the learnt motion allows the pre-trained generators for shape $Q(z_t^s|\tilde{M}_t^s)$ and texture $A(z_t^s|\tilde{T}_t^s)$ to interpolate between the captured 4D volumetric video shape and appearance sequences. Because the variational encoder-decoder produces a compact latent space it is possible to generate novel content by sampling from the learned space or interpolation of sampled latent vectors. Sampling of the latent space allows reproduction of the original 4D volumetric video sequences with a low reconstruction error. The sampling can be performed in two ways: random walk in the latent space that fits in the Gaussian distribution learned; or through 3D joint positions given as input to the networks. In this work, sampling is performed through 3D joint position as input. Interpolation in the learnt latent space allows transitions between observed sequences to create plausible novel motions. Interpolating the latent space is only possible because of the compact space representation produced by the generative network. This is performed, firstly, by sampling latent vectors using 3D joint position as input to the network. Once the latent vector is computed for two motion frames then interpolation is performed according to **Eq. 5**.

$$z_i = (1-\alpha)z_{t_1}^{s_1} + \alpha z_{t_2}^{s_2} \tag{5}$$

where z_i is the interpolated latent vector, α defines a normalised weighting [0..1] between latent vectors $z_{t_1}^{s_1}$ and $z_{t_2}^{s_2}$. Intermediate 4D shape and texture frames are synthesised to qualitatively evaluate how well the network is representing the 4D shape and appearance, **Section 5.2**.

4 DEEP4D MOTION GRAPHS

The following section introduces Deep4D motion graphs, a novel approach to generate motion graphs (Casas et al., 2012; Casas et al., 2013; Boukhayma and Boyer, 2015; Boukhayma and Boyer, 2019) from the Deep4D representation introduced in **Section 3**. The motion graphs, animation and rendering blocks presented in **Figure 5** are discussed in detail to demonstrate the steps taken to generate motion graphs capable of animating learnt characters from 4D volumetric video datasets. The goal is to merge the popular deep learning research field with traditional animation

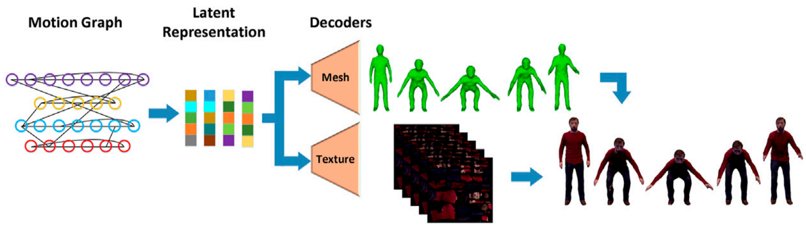

FIGURE 5 | Animation framework pipeline. The image illustrates the online block that generates high-resolution 4D volumetric video. The 4D Motion Graph contains the learnt motion representation allowing the decoders in real-time to synthesise 4D volumetric video sequences consisting of 4D mesh, 2D texture appearance.

pipelines to begin a new era for computer graphics, creating novel mechanisms to produce realistic human animations.

Firstly, we discuss the input data to the animation framework along with the pre-requisites for initialisation. Secondly, the generation of motion graphs for learnt 4D volumetric video is presented along with a discussion of the metrics chosen to evaluate similarity and transition costs between motion frames. Finally, a real-time motion synthesis approach to generate 4D video sequences with interactive animation control by concatenating and blending between the captured motion sequences is presented.

4.1 Input Data

The framework receives as input, skeletal motion data from 4D volumetric video estimated using a Skeleton Driven Surface Registration (SDSR) framework (Regateiro et al., 2018) and latent vectors z_t^s of each motion sequence M_t^s learned in **Section 3** for 4D shape and appearance learnt from a skeletal pose.

In the context of this section, a sequence of motion frames $\{\{F_t^s\}_{t=1}^{N_T^s}\}_{a=1}^{N_S}$ refers to collections of frames which contain representative latent vectors, and skeletal structures given by the SDSR framework as follows, $F_t^s = \{z_t^s, S_t^s\}$, where S_t^s is a skeletal structure from a motion sequence, which contains N_T^s number of frames $t = [1 \ldots N_T^s]$, representative of the original motion dataset. Lastly, it is necessary to utilise the pre-trained mesh generator $Q(z_t^s|\tilde{M}_t^s)$ and the appearance generator $A(z_t^s|\tilde{T}_t^s)$ from **Section 3** to interpret each latent vectors z_t^s stored as a motion frame in latent motion sequence F_t^s.

The generative networks synthesise $\tilde{M}_t^s$ meshes and $\tilde{T}_t^s$ texture maps for every $z_t^s \in F_t^s$, which represents a temporally consistent 4D mesh and appearance, i.e. the topology, vertex connectivity and texture coordinates are constant across all frames and sequences. The construction of a motion graph is independent of the learnt model, allowing the framework to generalise its application to other types of models. A motion graph is interpreted as a directed weighted graph structure built from captured 4D volumetric video sequences, where graph nodes represent frames that contain latent vectors which hold information about shape, motion and appearance, and edges link nodes together to represent motion pathways between frames.

4.2 Pre-processing

The data is required to be pre-processed; this offline process starts with training the generative networks described in **Section 3** for a skeleton motion sequences of a human character. Once training is complete the generators $Q(z_t^s|\tilde{M}_t^s)$ and $A(z_t^s|\tilde{T}_t^s)$ are used to recover the 3D meshes and 2D textures represented by each latent vector z_t^s, to allow the pre-processing step to be automated. The first step in the pre-processing stage is to connect frames within the same sequences automatically, and if possible create loops for cyclic motions, consequently a sequence can infinitely repeat itself. Loops are generated via searching on a similarity matrix $\boldsymbol{SIM_T}(F_{t_u}^i, F_{t_v}^j)$ for all pairs of frames in the same sequence to automatically choose the minimum cost, **Section 4.3** and **Section 4.4**. Transitions within the same sequence should produce the most natural motion; hence the shape and motion cost should be small.

The next step is to fully connect the graph by adding all possible transition combinations between sequences to allow better path estimations to be found for all frames. This step will generate a fully connected graph with appropriated edge weights using shape, motion and dynamic time warping metrics, as detailed in the following sections. Lastly, the graph is optimised using Dijkstra's algorithm to minimise the number of transition in the final motion graph, as detailed in **Section 4.5**.

4.3 Shape Similarity Metric

Similarity is computed for every pair of frames in the input 4D volumetric video sequences $\boldsymbol{SIM_T}(F_{t_u}^i, F_{t_v}^j)$, where $F_{t_u}^i$ is a frame t_u from the ith sequence $F_{t_u}^i = \{M_{t_u}^i, T_{t_u}^i\}$, comprising meshes $M_{t_u}^i$ and textures $T_{t_u}^i$, where $i = [1 \ldots N_S]$. For a given latent vector z_t^s the decoder $Q(z_t^s|\tilde{M}_t^s)$ reconstructs temporally consistent geometry, and the appearance generator $A(z_t^s|\tilde{T}_t^s)$ reconstructs the 2D texture appearance of generated frame. The shape, motion and appearance similarity is computed for every pair of source $F_{t_u}^i$ and target $F_{t_v}^j$ frames, having $t_u \in [1, N_T^s]$ and $t_v \in [1, N_T]$ frames for all sequences $i, j \in [1, N_S]$.

$$\boldsymbol{SIM_T}(F_{t_u}^i, F_{t_v}^j) = \theta \boldsymbol{SIM_M}(M_{t_u}^i, M_{t_v}^j) + (1-\theta)\boldsymbol{SIM_A}(T_{t_u}^i, T_{t_v}^j) \tag{6}$$

Where θ weights the relative importance of shape and appearance similarity, giving a complete similarity matrix $\boldsymbol{SIM_T}(F_{t_u}^i, F_{t_v}^j)$ for all frames generated by the learnt 4D volumetric video representation. To measure shape similarity we use the Euclidean distances and velocities between mesh vertices as illustrated in **Eq. 7**.

$$\boldsymbol{SIM_M}\left(M^i_{t_u}, M^j_{t_v}\right) = \frac{1}{N_V}\left(\|x^i_{t_u} - x^j_{t_v}\| + \|v^i_{t_u} - v^j_{t_v}\|\right) \tag{7}$$

Where vertex velocity $v^i_{t_u} = (x^i_{t_u} - x^i_{t_{u-1}})$, and N_V is the number of vertices. The appearance similarity uses the average absolute difference of the 2D texture appearance between two frames as illustrated in **Eq. 8**.

$$\boldsymbol{SIM_A}\left(T^i_{t_u}, T^j_{t_v}\right) = \frac{1}{N_X}\|T^i_{t_u} - T^j_{t_v}\| \tag{8}$$

Where N_X is the number of pixels. The similarities are normalised to the range (0,1) as follows:

$$\boldsymbol{SIM_Q}\left(F^i_{t_u}, F^j_{t_v}\right) = \frac{\boldsymbol{SIM_Q}\left(F^i_{t_u}, F^j_{t_v}\right) - min\left(\boldsymbol{SIM_Q}\left(F^i_{t_u}, F^j_{t_v}\right)\right)}{max\left(\boldsymbol{SIM_Q}\left(F^i_{t_u}, F^j_{t_v}\right)\right) - min\left(\boldsymbol{SIM_Q}\left(F^i_{t_u}, F^j_{t_v}\right)\right)} \tag{9}$$

Where $\boldsymbol{SIMQ}\,(\cdot)$ is either $\boldsymbol{SIM_M}\,(\cdot)$ or $\boldsymbol{SIM_A}\,(\cdot)$ similarity metrics for shape and appearance. The pre-computed similarity matrix $\boldsymbol{SIM_T}(F^i_{t_u}, F^j_{t_v})$ for all frames allows to evaluate in real-time the similarity cost between any source and target meshes.

4.4 Transition Edge Cost

An edge in a motion graph represents a transition between two frames, where for clarity frames will be described as nodes. For every edge, we associate a weight to represent the similarity of shape transitions between nodes quantitatively. Realistic transitions should require little change in shape and appearance corresponding to a small similarity score. Hence the metric used takes into account the optimal surface interpolation cost between any pair of nodes (Boukhayma and Boyer, 2017). The cost of transitioning is the sum of intermediate poses between source node u and destination node v weighted by the similarity score for each intermediate frame.

In order to smoothly blend source node u from a 3D mesh sequence to destination node v from another sequence, it is necessary to consider a blend window of length b. This window represents a successive number of nodes b_u, on the source sequence it begins at node u and ends at node $u + b_u - 1$, in the destination sequence a window b_v ending at node v and starting at node $v - b_v + 1$. Once, the window frame is initialised between source and destination sequence, it is necessary to extrapolate the nodes that gradually blend both sequences, generating smooth realistic transitions. To extract the optimal nodes from source and destination sequences we use a variant of dynamic time warping (DTW) (Muller, 2007; Witkin and Popovic, 1995 Wang and Bodenheimer, 2008; Casas et al., 2013) to estimate the best temporal warps w_u and w_v respectively with respect to the similarity metric defined in **Eq. (6)**. DTW was first introduced by Sakoe and Chiba (1990) for signal time alignment, it was used in conjunction with dynamic programming techniques for the recognition of isolated words and it had been widely used since then mainly for recognition tasks. The transition duration varies within a third of a second and 2 s (Wang and Bodenheimer, 2008), hence we allow the length b_u and b_v to vary between boundaries $b_{\min}$ and b_{max}. The optimal transitions with minimal total similarity cost $D(u, v)$ through the path generated from the DTW algorithm.

$$D(u,v) = \min_{b_u,b_v,w_u,w_v,D_l,D_l} \sum_{t\in[0,D_l]} \boldsymbol{SIM_T}\left(F^i_{t_u}, F^j_{t_v}\right) \tag{10}$$

where $\boldsymbol{SIM_T}\left(F^i_{t_u}, F^j_{t_v}\right)$ is the shape similarity cost defined in **Section 4.3**, and D_l is the length of the path found by the DTW algorithm considered as the transition duration, see supplementary material for illustration. The optimisation above finds the following optimal parameters (b_u, b_v, w_u, w_v, D_l, D_l), which are considered later for motions synthesis. Similar to **Section 4.3**, we define the edge weight between nodes to be the surface deformation cost $D\ (u,\ v)$ and its interpolated duration cost $D_l(u, v)$. **Eq. (11)** summarises the definition for the edge cost between nodes u and v.

$$D'(u,v) = \min\left[\alpha(v-u), D(u,v) + \alpha\ D_l(u,v)\right] \tag{11}$$

For the case nodes u and v are from the same sequence the surface deformation should be minimal. To control the tolerance between surface deformation and transition duration we add weight α.

This process will create a fully connected digraph where edges are weighted for the shape similarity and transition cost between nodes, in the following **Section 4.5** we will discuss how to prune and optimise the connectivity of the complete digraph.

4.5 Motion Graph Optimisation

The last stage in the framework aims to find a globally optimal solution to minimise the number of transitions between nodes. Plausible transitions can be achieved by selecting the minimum cost transition from the similarity matrix between sequences, to generate a motion graph. A fully connected digraph was generated from **Section 4.4**, which connects every pair of nodes for all existing motion sequences. Therefore selecting the minimum cost transition for every node would maintain dense connectivity in the graph.

We have implemented a globally optimal strategy that extracts and maintains only the best paths between every pair of nodes (Huang et al., 2009) Casas et al., 2011; Casas et al., 2013; Boukhayma and Boyer, 2015; Boukhayma and Boyer, 2017). This strategy corresponds to extracting the essential sub-graph from the complete digraph induced from the input sequences (Bordino et al., 2008). This method ensures the existence of at least one transition between any two nodes in the graph, which potentially yields a better use of the original data with less dead ends. Given the fully connected digraph, we use the Dijkstra algorithm on every pair of nodes to extract the shortest paths between source and target nodes. Once this process is completed, we remove all edges that do not belong to the new generated paths, giving a connected digraph that contains only the necessary least cost transitions. The resulting structure is also referred to as the union of shortest-path trees rooted at every graph node. This solution will guarantee the minimal difference when transitioning from frames of different sequences.

4.6 4D Volumetric Video Animation

This section demonstrates generation of 4D volumetric video using the Deep4D motion graphs. To generate a continuous stream of animation between motion sequences it is necessary to calculate the least costly transition path between a source frame $F^i_{t_u}$ and a target frame $F^j_{t_v}$ from different motion sequences. As discussed previously, the least costly transition should be a transition within the same motion sequence, consequently if the animation remains unchanged by the user the framework will play the same motion in a loop. If the user requests the character change to a new motion state, the animation framework computes the minimum transition cost $D(F^i_{t_u}, F^j_{t_v})$ from the current motion frame $F^i_{t_u}$ to the selected motion sequence $F^j_{t_v}$, and returns the following parameters $(b_u, b_v, w_u, w_v, D_l)$, **Section 4.4**. These parameters allow interpolation of the intermediate frames between frame u and v with a transition length of D_l, creating a seamless transition in real-time between different motion sequences. The approach presented in **Section 4.4** finds the corresponding pair of frames by computing the shortest path on the warps (w_u, w_v). The following sub-sections discuss how to synthesise 4D volumetric video and how intermediate frames are generated using generative networks.

4.6.1 4D Motion Synthesis

For every node in the motion graph we store the latent vector z^s_t that corresponds to a particular frame of a motion sequence. This allows for the pre-trained generator $Q(z^s_t|\tilde{M}^s_t)$ and $A(z^s_t|\tilde{T}^s_t)$ from the generative networks to reconstruct 3D mesh and 2D texture appearance for any given latent vector. At run-time the framework provides a latent vector z^s_t of the current frame and generates the corresponding dynamic mesh shape and texture appearance to synthesise the 4D volumetric video. **Figure 6** illustrates synthesised 4D volumetric video sequences. The motion graph representation generates seamless transitions to enable interactive character animation. The world coordinates of each frame are given by the root of the original 3D skeletal motion information which is used to transform the 3D mesh content given by the generators, allowing it to reproduce the original physical motion translations.

4.6.2 Motion Frames Interpolation

Edges in the motion graph represent transitions between frames take into account the shape, motion and appearance similarity. It is necessary to create intermediate blend frames to smoothly transition between different sequences. As seen in **Section 3.4**, the generative network allows synthesis of frames via interpolation of the latent vectors. Therefore, we perform a linear interpolation of the latent vectors for the given transition parameters, see **Section 4.4**, to create smooth human character animation. **Figure 7**, **Figure 8** illustrate interpolation between distinct body and face poses generating plausible intermediate mesh and texture.

5 RESULTS AND EVALUATION

This section presents results and evaluation for the proposed Pose2Shape network, the Pose2Appearance network from motion, and their applicability using Deep4D motion graphs to generated realistic animations, introduced in **Section 3.2** and **Section 3.3**. To evaluate the 4D animation framework we use publicly available volumetric video datasets for whole body and facial performance. The SurfCap dataset, JP and Roxanne characters, and Dan character (Casas et al., 2014) are reconstructed using multi-view stereo (Starck and Hilton,

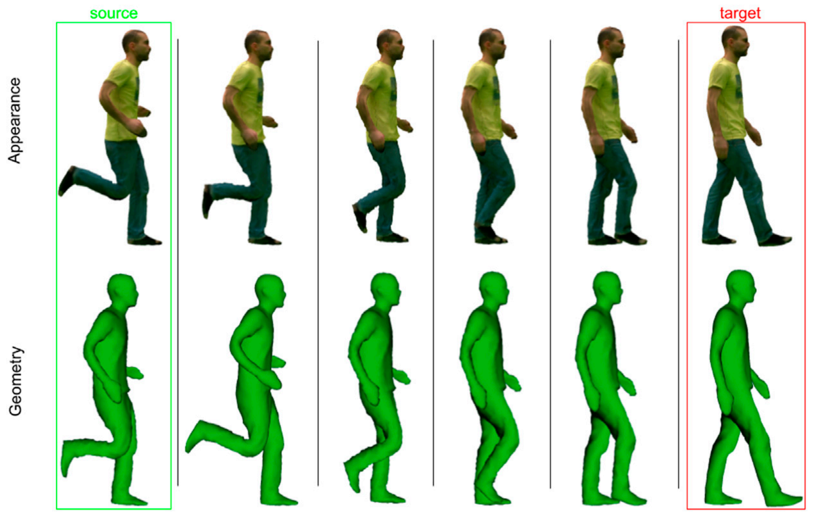

FIGURE 6 | Latent space interpolation, two frames with distinct motions are selected, on the left surrounded with a green box is the source, on the right surrounded with a red box is the target, in between is the interpolated results for 4D shape and appearance.

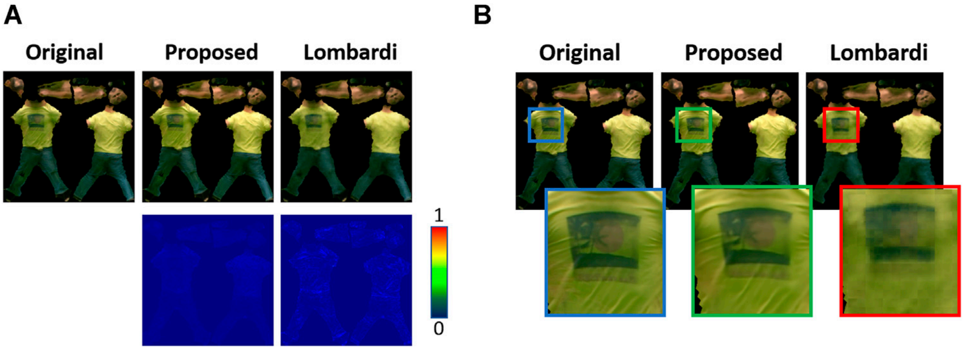

FIGURE 7 | Thomas appearance synthesis qualitative evaluation of Lombardi et al. (2018).

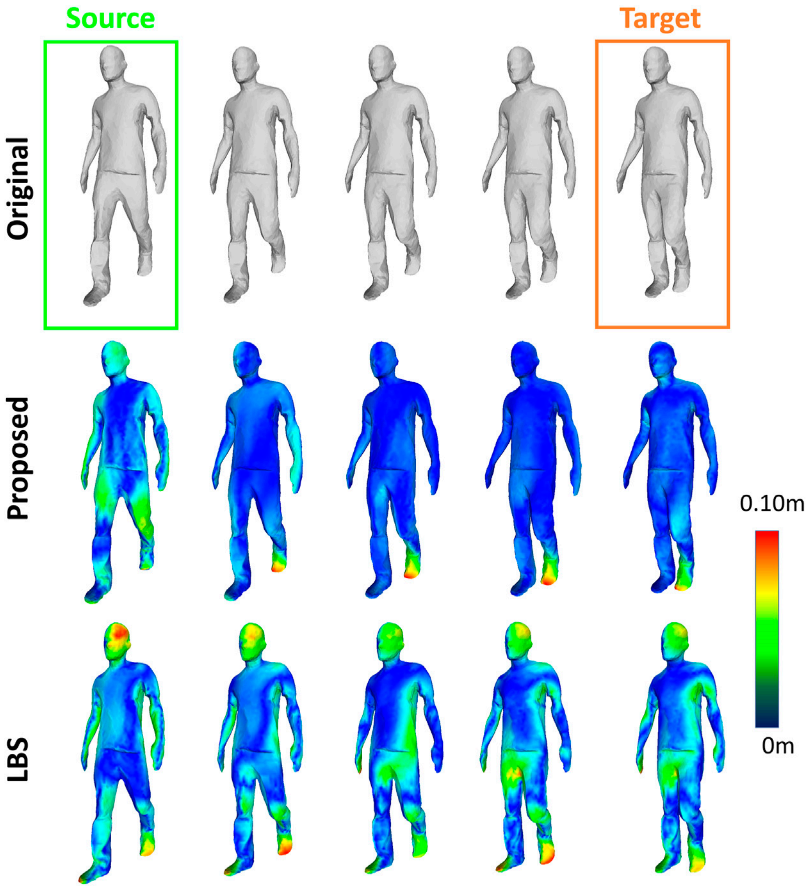

FIGURE 8 | Interpolation between two frames from the original motion sequence surrounded with a green and orange box. The top row represents the original sequence composed of five consequent frames. The middle row is the result of interpolating the learnt latent vector of the proposed network. The bottom row is the result of using linear blend skinning.

2007) and temporally aligned with SDSR (Regateiro et al., 2018) which allows for surface pose manipulation. Martin dataset (Klaudiny and Hilton, 2012) consists of one sequence of temporally aligned geometry and texture appearance of a human face, and 3D facial key-points given by OpenPose (Cao et al., 2021). Thomas dataset (Boukhayma and Boyer, 2015) consists of four sequences of temporally aligned meshes and texture appearance. An overview of dataset properties is shown in

TABLE 1 | Comparison of error metrics used for evaluation of 3D mesh and 2D texture appearance. The values represent the average error across the all motion sequence for different datasets.

Dataset	Mesh		Appearance		
	RMSE (m)	STDDV	MSE	SSIM	PSNR
Dan Casas et al. (2014)	0.0158	0.0156	0.0008	0.8417	30.7327
JP Starck and Hilton (2007)	0.0266	0.0257	0.0007	0.9610	31.1675
Martin Klaudiny and Hilton (2012)	0.0027	0.0015	0.0001	0.9813	38.6342
Roxanne Starck and Hilton (2007)	0.0166	0.0161	0.0002	0.9804	36.0430
Thomas Boukhayma and Boyer (2015)	0.0125	0.0122	0.0002	0.9889	35.9946

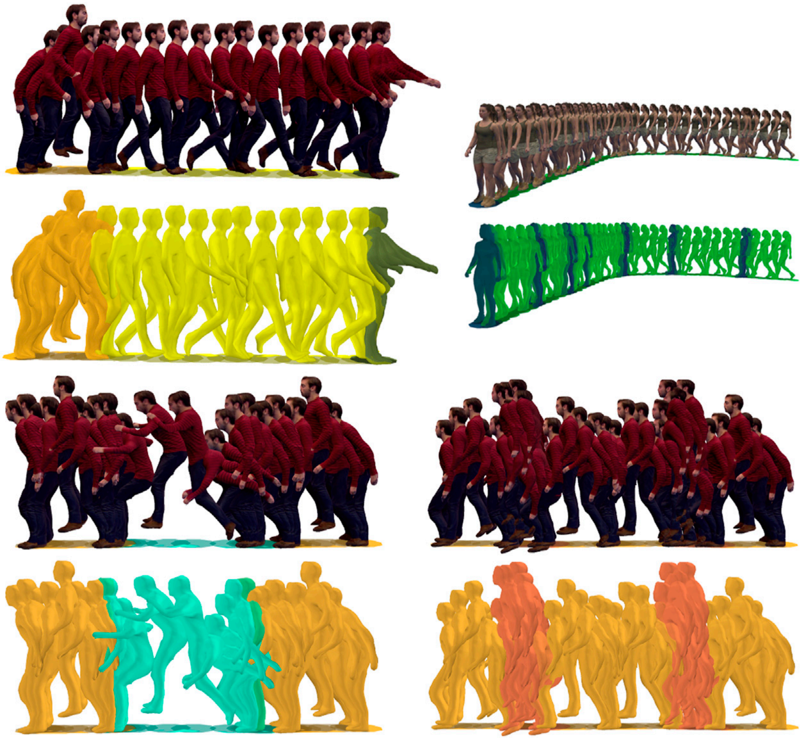

FIGURE 9 | Dan and Roxanne characters performing several animations, top-left: transition from jump to walk to reach; top-right: from walk to stand; bottom-left: from jump short to jump long; bottom-right: from jump short to jump high sequence. Mesh colours indicate motion sequences and the generated transitions.

Table 2. Examples of character animation using Deep4D motion graphs are shown in **Figures 9,6**. Results demonstrate that the proposed generative representation allows interactive character animation with seamless transitions between sequences based on interpolation of the latent space. The meshes are coloured to illustrate different motion sequences and interpolation between them when performing a blend transition. The learned generative model for shape and appearance synthesises animation with a quality similar to the input 4D video.

5.1 Quantitative Results

The variational encoder-decoder uses **Eq. (4)** as a metric to predict plausible shape reconstructions from skeletal pose. The Pose2Appearance network uses the mean squared error (MSE) as loss function between generated images and ground truth as a metric to predict plausible high resolution textures. The comparison was performed between the training data, to ensure minimum error when sampling the original sequences, and validation data to guarantee a plausible result when generating unseen mesh.

We compare generated 3D meshes with ground truth geometry acquired from multiple view stereo reconstruction (Starck and Hilton, 2007). 3D mesh evaluation is performed using Hausdorff distance defined as $d_H(A, B) = max\{\sup_{a\in A} d(a, B), \sup_{b\in B} d(b, A)\}$, where $d(a, B)$ and $d(b, A)$ is the distance from a point a to a set B and from a point b to a set A, which has been shown to be a good measurement between 3D meshes. The comparison contains training and validation data for all sequences, **Table 1**. The appearance is evaluated using three metrics that are commonly used to assess image quality: mean

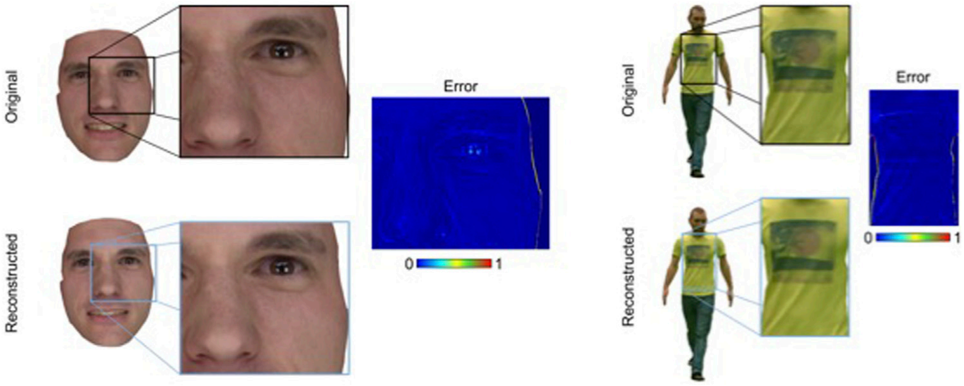

FIGURE 10 | The images represent the same motion frame, the top images were rendered using the original geometry and texture appearance, and the bottom images were rendered using the generative networks. The error image is an error histogram between the original and resulted images. It is visible from the reconstructed that it preserves original details such as, reflection on the eye ball, clothing patterns and wrinkles.

squared distance (MSE); multi-scaled structural similarity (MS-SSIM); peak signal to noise ratio (PSNR), **Table 1** for results.

5.2 Qualitative Evaluation

We compare our network generated results to rendered images of the original textured model and synthesised 4D volumetric content, **Figure 10** and supplementary material for more results. Our network is able to capture dynamic shape detail and high frequency appearance details such as wrinkles and hair movement, **Figure 10**. The network is also capable of interpolating the existing data to generate novel geometry and appearance within the learned space. To test the interpolation performance of the network, the mesh and appearance of two encoded frames were selected and intermediate frames synthesised. **Figures 7,8** shows a more challenging example for two randomly selected frames with large differences in shape and appearance, note that the method is able to produce a natural transition between frames.

The proposed generative network maps 3D skeletal pose to 4D volumetric video sequences consisting of shape and appearance. To evaluate this capability we use existing public skeletal motion capture sequences (CMU Graphics Lab, 2001) to synthesise novel 4D animations. To drive the generative network, we use the 3D skeletal joint positions p_t^s to obtain the encoded latent vectors z_t^s, sampling from the learnt distribution $P(p|z)$. **Figure 9** shows three characters driven using a novel motion capture sequence This demonstrates the potential to generate novel plausible 4D shape and appearance sequences from MoCap input for similar motions.

5.3 Appearance Synthesis Evaluation

In this section, we evaluate the performance of the progressive appearance generator against the state-of-the-art method proposed by Lombardi et al. (2018) for facial image synthesis. The variant network architecture was chosen to allow for appearance synthesis only, as we intend to evaluate the texture synthesis quality, see supplementary for network illustration. Therefore we have removed the mesh and view-point conditioning from the original network architecture. We trained this network on 2D textures from the Thomas

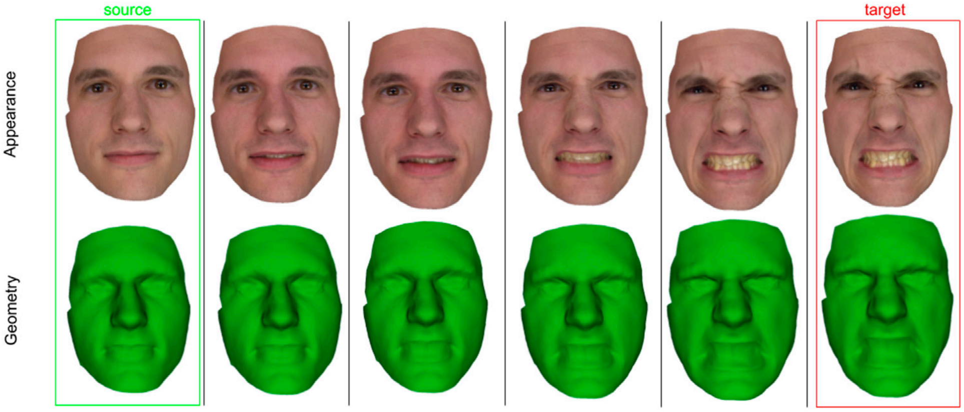

FIGURE 11 | Latent space interpolation, two frames with distinct 3D motion landmarks are selected, on the left surrounded with a green box is the source, on the right surrounded with a red box is the target, in between is the interpolated results for shape and appearance.

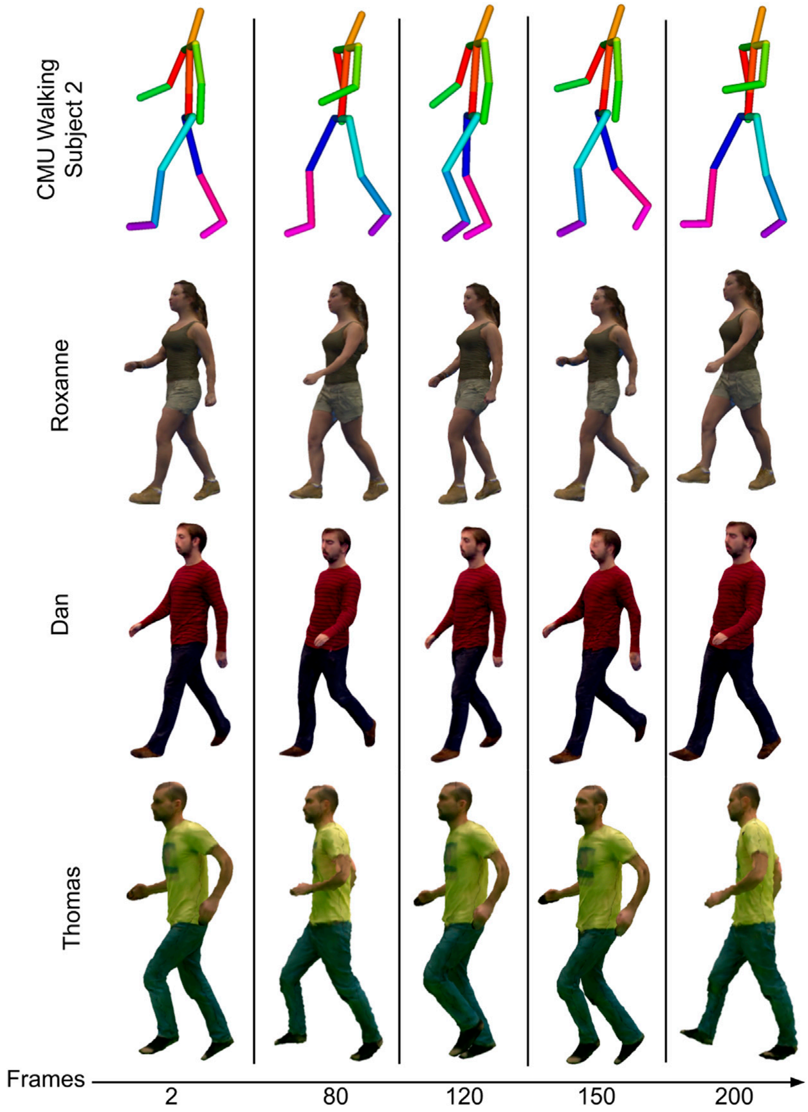

FIGURE 12 | The top row is a motion capture sequence which is used to synthesise the bottom rows. The bottom rows are 4D volumetric content of Roxanne. Dan and Thomas datasets. It is visible that is able to generate 4D volumetric video from skeletal motion for different subjects.

(Boukhayma and Boyer, 2015) and Martin (Klaudiny and Hilton, 2012 datasets, where the training took approximately 10 days for 10^4 training cycles, with a mini-batch size of 64. This network minimises the MSE error and the KL-divergence simultaneously, similar to the proposed approach.

Figure 11 illustrates qualitative evaluation for this experiment. We have chosen one random sample from the training dataset to evaluate the quality of the texture synthesis given a seen example. **Figure 11A** presents heat-map images to compare the synthesised result against the ground-truth for the proposed and Lombardi networks. It is visible that the proposed network outperforms the Lombardi et al., 2018 approach, this is more visible on the close-up **Figure 11B**, where the details on the t-shirt have been lost when using the Lombardi et al., 2018 network. The proposed network is capable of preserving the printed image on the t-shirt along with wrinkles present in the original image. The lack of detail and the presence of blurred results from state-of-the-art Lombardi et al., 2018 network has led to the network presented in **Section 3.3**. The proposed approach is a more sophisticated network, capable of preserving fine details and complex structures, and achieves faster training given limited computational hardware.

5.4 Linear Blend Skinning Comparison

This section includes a comparison of the proposed Pose2Shape network against linear blend skinning (LBS) techniques demonstrating the benefits of using the proposed network. LBS is a widely used approach in real-time character animation for deforming a surface mesh according to an underlying bone structure, where every bone contains a transformation matrix that affects a group of vertices. This relation is given by a weighting attribute that weights the

TABLE 2 | The table illustrates the total amount of disk space occupied in Megabytes (MB). The original column represents 3D mesh and 2D textures of the original dataset, and the latent space and decoder columns represent the required memory to synthesise 3D meshes and 2D texture appearance.

Dataset	Vertices	Frames	Original (MB)	Latent space (MB)	Decoder (MB)
Dan Casas et al. (2014)	2,667	1,447	768.2	2.6	104
JP Starck and Hilton (2007)	3,463	1788	1,272.7	4.7	106.8
Martin Klaudiny and Hilton (2012)	2,689	310	479.1	0.80	104
Roxanne Starck and Hilton (2007)	2,475	414	428.1	1.1	103.3
Thomas Boukhayma and Boyer (2015)	5,002	212	1,186.3	0.55	112.4

contribution of a bone transformation on a vertex. LBS is computationally efficient and commonly used in animation frameworks, allowing real-time character animation by manipulation of surface geometry using a low-dimensional skeletal structure. Although, it does not allow propagation of non-linear surface deformation, and it can cause artefacts on the mesh surface. To understand if the proposed Pose2Shape model is capable of learning non-linear attributes from the input data instead of only learning a linear mapping, we compare the results against LBS. For this comparison, we present two experiments; the first experiment evaluates the interpolation performance against LBS. The second experiment compares the synthesis of a mesh sequence against using LBS to animate the same motion sequence, please see supplementary material for second experiment. To compare the meshes, we use the Hausdorff distance metrics, discussed in **Section 5.1**.

Figure 12 illustrates the results for the first experiment using the Thomas (Boukhayma and Boyer, 2015) dataset. The top row represents the original sequence of walking motion, the source and target frames surround by green and orange boxes, respectively, represent the frames used for interpolation. The middle row shows the results of interpolating the latent vectors representative of the source and target frames. Latent vectors were generated by encoding the respective skeletons of the source and target frames. As a consequence, we can synthesise intermediate poses following **Eq. (5)**. The bottom row shows the LBS results for source and target frames. LBS is achieved using the animation capabilities of the SDSR framework (Regateiro et al., 2018), which allows mesh manipulation through skeletal animation. Therefore, given the original skeletal motion frames, we map the source frame onto the target frame whilst generating the intermediate frames, as illustrated in the bottom row.

This experiments demonstrates the ability to generate a more accurate reconstruction of the original mesh compared to LBS. To support these figures, **Table 1**, shows quantitative evaluation for all the datasets between LBS and the proposed results.

5.5 Compression

Table 2 demonstrates the proposed approach is capable of compressing 4D volumetric video through a deep learnt representation. The latent space representation achieves up to two orders of magnitude reduction in the size of the captured 4D volumetric video depending on sequence length. The decoders have an approximate size 105 *MB* with the texture encoder size constant, 94 *MB*, due to the fixed texture image resolution and the mesh encoder dependent on the mesh resolution, 10–18 *MB*.

5.6 Performance

Presented results were generated using a desktop PC with an Intel Core i7-6700K CPU, 64 GB of RAM and an Nvidia Geforce GTX 1080 GPU. Our training time is approximately 4 days on a single GPU. The non-optimised animation framework performance achieves ≈10 frames per second (fps) at full resolution. The performance bottleneck is in the Pose2Appearance network from **Section 3.3** as a result of the high number of convolutional layer and training parameters. The generative networks from **Section 3.2** is capable of achieving ≈35 frames per second (fps). The characteristic of the Pose2Appearance network allows for multi-scale texture resolution improving rendering performance and memory usage, see supplementary material for illustration. The generator is capable of reconstructing multiple resolutions of the appearance, increasing rendering performance and decreasing memory usage, allowing the possibility to use on platforms with memory constraints.

5.7 Limitations

Primary limitation is quality of the 4D volumetric video sequences for training. The synthesis will reproduce artefacts present in the input data such as shape error or appearance misalignment. Currently this is limited by the publicly available 4D video sequences but will improve as 4D volumetric video improves. The current implementation is not optimised for texture rendering due copy operations between CPU and GPU memory, with optimisation this could achieve >30 fps for shape and appearance synthesis. Motion capture data synthesis may create undesired artefacts on the appearance and shape if the skeletal motion is outside the space of observed 4D motions as this requires extrapolation in the latent space. Currently the network is only able to represent one character a time, an interesting extension for future work would be to encode multiple characters in a single space, or a single person wearing multiple types of clothing.

6 CONCLUSION

The proposed Deep4D representation enables interactive animation through motion graphs to generate dynamic shape and high-quality appearance. The 4D generative network supports interpolation in the latent space to synthesise novel intermediate motions allowing smooth transitions between captured sequences. The Pose2Appearance network synthesises high resolution textures for the learnt motion space, whilst preserving details of motion and realistic details. The proposed network is capable of a compact representation of multiple 4D volumetric video sequences achieving up-to two orders of magnitude compression compared to the captured 4D volumetric video. The generative network allows mapping of skeletal motion capture data to generate novel 4D volumetric video sequences with detailed dynamic shape and appearance. The approach achieves efficient representation and real-time rendering of 4D volumetric video in a motion graph for interactive animation. This overcomes the limitations of previous approaches to animate 4D volumetric video which require high storage and computational costs. Generative network usually suffer from discontinuities in areas where there is insufficient training data. This limitation is overcome by enforcing transitions through the motion graph which does not allow for extrapolation outside the space of observed 4D volumetric video. The proposed method is able to preserve shape details, motion and appearance as shown in the evaluation. We demonstrated the integration of the proposed generative network with traditional animation frameworks, improving on interpolation between different motions, and adding more information to the similarity metrics to improve the quality of motion transitions. The animation framework is independent of the network architecture, allowing for future improvements in either of the frameworks. For instance, the training performance of the neural network can be improved by reducing the number of convolutional layers, which consequently improves the run-time appearance rendering. The animation framework can be extended to parameterised motion, allowing increased interactivity and motion control.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in the CVSSP3D (https://cvssp.org/data/cvssp3d/) and INRIA (https://hal.inria.fr/hal-01348837/file/Data_EigenAppearance.zip) repositories.

ETHICS STATEMENT

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

JR is the first author and responsible for the implementation and also wrote the first draft of the manuscript. MV and AH supervised the implementation, manuscript generation and contributed to the final draft of the manuscript. All authors contributed to the manuscript revision, read, and approved the submitted version.

FUNDING

This research was supported by the EPSRC “Audio-Visual Media Research Platform Grant” (EP/P022529/1), “Polymersive: Immersive Video Production Tools for Studio and Live Events’” (InnovateUK 105168), and “AI4ME: AI for Personalised Media Experiences” UKRI EPSRC (EP/V038087/1).

ACKNOWLEDGMENTS

The authors would also like to thank Adnane Boukhayma for providing the “Thomas” dataset used for evaluation. The work presented was undertaken at CVSSP.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frvir.2021.739010/full#supplementary-material

REFERENCES

Arikan, O., Forsyth, D. A., O’Brien, J. F., and O’Brien, J. F. (2003). Motion Synthesis from Annotations. *ACM Trans. Graph.* 22, 402–408. doi:10.1145/882262.882284

Bordino, I., Donato, D., Gionis, A., and Leonardi, S. (2008). “Mining Large Networks with Subgraph Counting,” in 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008, (IEEE), 737–742. doi:10.1109/icdm.2008.109

Boukhayma, A., and Boyer, E. (2017). “Controllable Variation Synthesis for Surface Motion Capture,” in 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017, (IEEE), 309–317. doi:10.1109/3DV.2017.00043

Boukhayma, A., and Boyer, E. (2019). Surface Motion Capture Animation Synthesis. *IEEE Trans. Vis. Comput. Graphics* 25, 2270–2283. doi:10.1109/tvcg.2018.2831233

Boukhayma, A., and Boyer, E. (2015). “Video Based Animation Synthesis with the Essential Graph,” in 2015 International Conference on 3D Vision, Lyon, France, 19–22 October 2015, (IEEE), 478–486. doi:10.1109/3dv.2015.60

Brock, A., Donahue, J., and Simonyan, K. (2018). *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. New Orleans, LA, USA: International Conference on Learning Representations (ICLR).

Budd, C., Huang, P., Klaudiny, M., and Hilton, A. (2013). Global Non-rigid Alignment of Surface Sequences. *Int. J. Comput. Vis.* 102, 256–270. doi:10.1007/s11263-012-0553-4

Cagniart, C., Boyer, E., and Ilic, S. (2010). “Free-form Mesh Tracking: A Patch-Based Approach,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 13–18 June 2010, (IEEE), 1339–1346. doi:10.1109/CVPR.2010.5539814

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2021). Openpose: Realtime Multi-Person 2d Pose Estimation Using Part Affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 172–186. doi:10.1109/tpami.2019.2929257

Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H.-P. (2003). Free-viewpoint Video of Human Actors. *ACM Trans. Graph.* 22, 569–577. doi:10.1145/882262.882309

Casas, D., Tejera, M., Guillemaut, J.-Y., and Hilton, A. (2012). “4d Parametric Motion Graphs for Interactive Animation,” in Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (ACM), I3D '12, Costa Mesa, CA, 9–11 March 2012, (Association for Computing Machinery), 103–110. doi:10.1145/2159616.2159633

Casas, D., Tejera, M., Guillemaut, J.-Y., and Hilton, A. (2013). Interactive Animation of 4d Performance Capture. *IEEE Trans. Vis. Comput. Graphics* 19, 762–773. doi:10.1109/TVCG.2012.314

Casas, D., Tejera, M., Guillemaut, J.-Y., and Hilton, A. (2011). "Parametric Control of Captured Mesh Sequences for Real-Time Animation," in Proceedings of the 4th international conference on Motion in Games, Edinburgh, UK, 13–15/11/2011 (Berlin, Germany: Association for Computing Machinery), 242–253. doi:10.1007/978-3-642-25090-3_21

Casas, D., Volino, M., Collomosse, J., and Hilton, A. (2014). 4d Video Textures for Interactive Character Appearance. *Comput. Graphics Forum* 33, 371–380. doi:10.1111/cgf.12296

[Dataset] CMU Graphics Lab (2001). *Cmu Graphics Lab Motion Capture Database*. Pittsburgh, PA: Carnegie Mellon University.

Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., et al. (2015). High-quality Streamable Free-Viewpoint Video. *ACM Trans. Graph.* 34, 1–13. doi:10.1145/2766945

de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.-P., and Thrun, S. (2008). Performance Capture from Sparse Multi-View Video. *ACM Trans. Graph.* 27, 1–10. doi:10.1145/1360612.1360697

Dosovitskiy, A., Springenberg, J. T., and Brox, T. (2015). "Learning to Generate Chairs with Convolutional Neural Networks," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Boston, MA, USA: IEEE), 1538–1546. doi:10.1109/cvpr.2015.7298761

Ekman, P. (1980). *The Face of Man: Expressions of Universal Emotions in a New guinea Village*. Incorporated: Garland Publishing.

Esser, P., Haux, J., Milbich, T., and Ommer, B. (2019). *Towards Learning a Realistic Rendering of Human Behavior*. Cham: Springer, 409–425. doi:10.1007/978-3-030-11012-3_32

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). *Generative Adversarial Networks*. New York, NY, USA: ACM.

Hilsmann, A., Fechteler, P., Morgenstern, W., Paier, W., Feldmann, I., Schreer, O., et al. (2020). Going beyond Free Viewpoint: Creating Animatable Volumetric Video of Human Performances. *IET Comput. Vis.* 14, 350–358. doi:10.1049/iet-cvi.2019.0786

Holden, D., Komura, T., and Saito, J. (2017). Phase-functioned Neural Networks for Character Control. *ACM Trans. Graph.* 36, 1–13. doi:10.1145/3072959.3073663

Huang, P., Budd, C., and Hilton, A. (2011). "Global Temporal Registration of Multiple Non-rigid Surface Sequences," in CVPR 2011, Colorado Springs, CO, 20–25 June 2011, (IEEE), 3473–3480. doi:10.1109/cvpr.2011.5995438

Huang, P., Hilton, A., and Starck, J. (2009). "Human Motion Synthesis from 3D Video," in 2009 IEEE Conference on Computer Vision and Pattern Recognition (Miami, FL, USA: IEEE), 1478–1485. doi:10.1109/CVPR.2009.5206626

Huang, P., Tejera, M., Collomosse, J., and Hilton, A. (2015). Hybrid Skeletal-Surface Motion Graphs for Character Animation from 4d Performance Capture. *ACM Trans. Graph.* 34, 1–14. doi:10.1145/2699643

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). *Image-to-Image Translation with Conditional Adversarial Networks*. Honolulu, Hawaii: IEEE.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). *Perceptual Losses for Real-Time Style Transfer and Super-resolution*. Amsterdam, Netherlands: Springer International Publishing.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. Vancouver, Canada: International Conference on Learning Representations (ICLR).

Karras, T., Laine, S., and Aila, T. (2018). *A Style-Based Generator Architecture for Generative Adversarial Networks*. Salt lake city, Utah: IEEE.

Kingma, D. P., and Welling, M. (2013). *Auto-Encoding Variational Bayes*. Banff, AB, Canada: International Conference on Learning Representations (ICLR).

Klaudiny, M., and Hilton, A. (2012). "High-detail 3d Capture and Non-sequential Alignment of Facial Performance," in 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission, Zurich, Switzerland, 13–15 October 2012, (IEEE), 17–24. doi:10.1109/3dimpvt.2012.67

Kovar, L., Gleicher, M., and Pighin, F. (2002). Motion Graphs. *ACM Trans. Graph.* 21, 473–482. doi:10.1145/566654.566605

[Dataset] Laine, S. (2018). *Feature-based Metrics for Exploring the Latent Space of Generative Models*. Vancouver, Canada: International Conference on Learning Representations (ICLR).

Lombardi, S., Saragih, J., Simon, T., and Sheikh, Y. (2018). Deep Appearance Models for Face Rendering. *ACM Trans. Graph.* 37, 1–13. doi:10.1145/3197517.3201401

Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., and Fritz, M. (2017). *Disentangled Person Image Generation*. Salt Lake City, Utah: IEEE.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). *Spectral Normalization for Generative Adversarial Networks*. Vancouver, Canada: International Conference on Learning Representations (ICLR).

Muller, M. (2007). *Information Retrieval for Music and Motion*. Berlin, Germany: Springer-Verlag.

Paier, W., Hilsmann, A., and Eisert, P. (2020). "Neural Face Models for Example-Based Visual Speech Synthesis," in CVMP '20: European Conference on Visual Media Production (New York, NY, USA: Association for Computing Machinery). doi:10.1145/3429341.3429356

Prada, F., Kazhdan, M., Chuang, M., Collet, A., and Hoppe, H. (2016). Motion Graphs for Unstructured Textured Meshes. *ACM Trans. Graph.* 35, 1–14. doi:10.1145/2897824.2925967

Regateiro, J., Hilton, A., and Volino, M. (2019). "Dynamic Surface Animation Using Generative Networks," in International Conference on 3D Vision (3DV), Quebec, Canada, 16–19 September 2019, (IEEE). doi:10.1109/3dv.2019.00049

Regateiro, J., Volino, M., and Hilton, A. (2018). "Hybrid Skeleton Driven Surface Registration for Temporally Consistent Volumetric Video," in 2018 International Conference on 3D Vision (3DV) (Verona, Italy: IEEE), 514–522. doi:10.1109/3DV.2018.00065

Sainburg, T., Thielk, M., Theilman, B., Migliori, B., and Gentner, T. (2018). *Generative Adversarial Interpolative Autoencoding: Adversarial Training on Latent Space Interpolations Encourage Convex Latent Distributions*. New Orleans, Louisiana: International Conference on Learning Representations (ICLR).

Sakoe, H., and Chiba, S. (1990). *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 159–165. doi:10.1016/b978-0-08-051584-7.50016-4

Siarohin, A., Sangineto, E., Lathuiliere, S., and Sebe, N. (2017). *Deformable GANs for Pose-Based Human Image Generation*. Salt Lake City, Utah: IEEE.

Starck, J., and Hilton, A. (2007). Surface Capture for Performance-Based Animation. *IEEE Comput. Grap. Appl.* 27, 21–31. doi:10.1109/MCG.2007.68

Starck, J., Miller, G., and Hilton, A. (2005). "Video-based Character Animation," in SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (New York, NY, USA: ACM), 49–58. doi:10.1145/1073368.1073375

Tan, Q., Gao, L., Lai, Y.-K., and Xia, S. (2018). "Variational Autoencoders for Deforming 3d Mesh Models," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, 18–22 June 2018, (IEEE), 5841–5850. doi:10.1109/CVPR.2018.00612

Tanco, L. M., and Hilton, A. (2000). "Realistic Synthesis of Novel Human Movements from a Database of Motion Capture Examples," in Proceedings Workshop on Human Motion, Austin, Texas, 7–8 Decembre 2000, (IEEE), 137–142. doi:10.1109/HUMO.2000.897383

Tejera, M., and Hilton, A. (2013). "Learning Part-Based Models for Animation from Surface Motion Capture," in 2013 International Conference on 3D Vision (Seattle, WA, USA: IEEE), 159–166. doi:10.1109/3DV.2013.29

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2017). *MoCoGAN: Decomposing Motion and Content for Video Generation*. Salt Lake City, Utah: IEEE.

Ulyanov, D., Lebedev, V., Vedaldi, A., and Lempitsky, V. (2016). *Texture Networks: Feed-Forward Synthesis of Textures and Stylized Images*. New York, USA: ACM.

Vlasic, D., Baran, I., Matusik, W., and Popovic, J. (2008). Articulated Mesh Animation from Multi-View Silhouettes. *ACM Trans. Graph.* 27, 1–97. doi:10.1145/1360612.1360696

Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). *Generating Videos with Scene Dynamics*. Barcelona, Spain: ACM.

Wang, J., and Bodenheimer, B. (2008). Synthesis and Evaluation of Linear Motion Transitions. *ACM Trans. Graph.* 27 (1), 1–15. doi:10.1145/1330511.1330512

Witkin, A., and Popovic, Z. (1995). "Motion Warping," in SIGGRAPH '95: Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques (ACM), Los Angeles, CA, 6–11 August 1995, (Association for Computing Machinery), 105–108. doi:10.1145/218380.218422

Zhu, J.-Y., Krahenbuhl, P., Shechtman, E., and Efros, A. A. (2016). *Generative Visual Manipulation on the Natural Image Manifold*. Amsterdam, The Netherlands: Springer.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). *Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks*. Venice, Italy: IEEE.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.