Dynamic Surface Capture for Human Performance by Fusion of Silhouette and Multi-view Stereo

Zheng Zhang*
zhangzheng@nudt.edu.cn
College of System Engineering, National University of Defense Technology
Changsha, Hunan, China

You Li†
gogogadget@qq.com
National Key Laboratory of Human Factors Engineering, China Astronaut Research and Training Center
Beijing, China

Xiangrong Zeng
xiangrongzeng@nudt.edu.cn
College of System Engineering, National University of Defense Technology
Changsha, Hunan, China

Sheng Tan
3554984700@qq.com
4DCapture Group, FDVUST
Kuming, Yunan, China

Changhua Jiang
jiangch2022@qq.com
National Key Laboratory of Human Factors Engineering, China Astronaut Research and Training Center
Beijing, China

ABSTRACT

We present a multi-camera based 3D dynamic surface capture solution, which supports high-fidelity generation of 3D dynamic content for various performance scenes. Our system uses a set of RGB cameras to synchronously acquire scene image sequences and performs a processing pipeline to produce 4D videos. We propose a multi-view point cloud reconstruction method which integrates volumetric based guidence and contraint into a coarse-to-fine depth estimation framework. It gives accurate point cloud models and can handle well with the scenes where textureless objects and multiple subjects present. We also present new methods and introduce modifications for several other key computation modules of the processing pipeline, including foreground segmentation, multi-camera calibration, mesh surface reconstruction and registration, texturing, 4D video compression, *etc.* Experiments on captured scenes show that our system can produce highly accurate and realistic 4D models of the human performances. We develop a 4D video player and toolkit plugins, and demonstrate the use of integrating the 4D content in VR and AR applications.

CCS CONCEPTS

• **Computing methodologies**;

KEYWORDS

Surface Capture, Volumetric Capture, Free-viewpoint Video, 4D Reconstruction, 4D Video

*Corresponding author: Zheng Zhang, zhangzheng@nudt.edu.cn
†Corresponding author: You Li, gogogadget@qq.com

ACM Reference Format:
Zheng Zhang, You Li, Xiangrong Zeng, Sheng Tan, and Changhua Jiang. 2022. Dynamic Surface Capture for Human Performance by Fusion of Silhouette and Multi-view Stereo. In *The 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI '22), December 27–29, 2022, Guangzhou, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3574131.3574450

1 INTRODUCTION

High-fidelity 3D digital human has a wide range of applications such as immersive live-action performance, sports analysis, films and games. The traditional 3D animated content creation process uses 3D modeling or animation/game software tools to create virtual characters, and then uses captured motion data to drive a model to form character animation. These motion capture based technologies can only provide coarse motion while cannot capture small deformations or movements of the subject. Although there are other advanced technologies such as facial capture, realistic clothing modeling, *etc.*, which work together with motion capture can help in creation of realistic digital human. However, it often requires label-intensive and time-consuming work.

3D dynamic surface capture is to use multiple cameras to acquire image or video data of the human performance scene from multiple angles, and reconstruct space-time coherent 3D models of human performance. The space-time coherent 3D content can be viewed from any perspective in space and played and controlled in time, often referred to as 4D video, free-viewpoint video, or volumetric video [Carranza et al. 2003; Collet et al. 2015; Schreer et al. 2020]. Early work [de Aguiar et al. 2008; Gall et al. 2009; Starck and Hilton 2007; Vlasic et al. 2008] often uses a sparse set of cameras and relies on adjusting a template model to reconstructed meshes. The use of template has difficult in matching deformations in shape and clothing of complex scenes such as that having multiple subjects. Recent methods often adopt a per-frame independent reconstruction framework and employs more number of cameras to enhance reconstruction quality [Collet et al. 2015; Leroy et al. 2017; Liu et al. 2010; Schreer et al. 2020, 2019]. Since image data from many more perspectives are involved, these methods can produce highly accurate and realistic dynamic model of the scene. The disadvantage is that they require large number of cameras and often high computational cost to reconstruct precise surface details. Although advances have been achieved in recent years, creating highly realistic and accurate models for scenes with textureless objects, fast motions and multiple subjects is still challenging.

In this work, we present a surface capture solution to produce realistic and detailed 4D videos of human performances. The main contribution of this work including: (1) we propose a multi-view point cloud reconstruction method which fuses silhouettes and multi-view stereo to attain accurate reconstruction of the scene. This 3D reconstruction method handles well with the scenes where textureless objects and multiple characters present; (2) we develop a robust processing pipeline which involves of key computation modules such as foreground segmentation, multi-camera calibration, mesh surface reconstruction, mesh registration, texturing, 4D video compression, *etc.* We present new methods or introduce modifications for the computation modules to achieve fidelity and accuracy of 4D production. We also implement a 4D video player and toolkit plugins, and demonstrate the use of integrating the 4D content in VR and AR applications.

2 RELATED WORK

The method of 3D dynamic surface capture can be divided into two categories according to the type of data acquisition sensor used: multiple RGB camera based and depth sensor based. The depth sensor based methods [Dou et al. 2016; Du et al. 2019; Pandey et al. 2019] is easier to achieve real-time reconstruction, but the size of capture space is often limited by the effective imaging distance of the depth sensor. The RGB camera methods, which need to reconstruct 3D point cloud data from the captured images, can support large-scale spatial capture to meet potential wider application needs. This paper focuses on the multiple RGB camera based approaches.

One typical way of multi-camera surface capture is to first obtain a coarse mesh model by aligning a template model with the multi-view observations in a process of articulated body pose estimation [Gall et al. 2009; Vlasic et al. 2008], or by a coarse per-frame independent mesh reconstruction of the subject [de Aguiar et al. 2008; Starck and Hilton 2007], it then performs a surface refinement step to obtain a more detailed mesh for every frame. The pioneer work [Carranza et al. 2003] uses a simplified skeleton model and fits it to multiple silhouette images. Vlasic *et al.* [Vlasic et al. 2008] present a system to capture both the skeleton and the shape, by fitting a mesh based template to visual hull and then adjusting the deformed template to the silhouettes. Gall *et al.* [Gall et al. 2009] introduce a similar framework that first estimates skeleton body pose via optimization and then adjusts the mesh template to recover the shape model. To obtain more accurate shape reconstruction, Aguiar *et al.* present a performance capture solution that does not requires of skeleton driven model, while it deforms the reconstructed mesh in a coarse-to-fine manner to fit the silhouette and feature points. Due to the complex and diverse deformation of the human body and clothing, it is difficult for the template to accurately deform and match the shape and clothing appearance in certain poses. In addition, pre-defined template models have certain limitations on capturing content, such as dynamic scenes with multiple character interaction.

Another way of dynamic surface capture adopts a per-frame independent accurate reconstruction framework [Collet et al. 2015; Leroy et al. 2017; Liu et al. 2010; Schreer et al. 2019]. Since image data from many more perspectives are involved in the 3D reconstruction, these methods can produce highly accurate and realistic dynamic model of the scene. The core of these method is to performance accurate point cloud or mesh model reconstruction from multi-camera views. Multi-view reconstruction is a classic traditional problem in the field of computer vision. Shape-from-Silhouette is one typical multi-view reconstruction method, which is often used for dynamic shape reconstruction [N. et al. 2008; Starck and Hilton 2007]. Starck *et al.* [Starck and Hilton 2007] present a wide-baseline multi-camera studio for capturing dynamic surface. In their solution, visual hull is integrated with 2D image observations for refining dense reconstruction, where a graph-cut optimization is employed for a task of binary labeling. Ahmed *et al.* [N. et al. 2008] first construct a visual hull shape sequence, then use optical flow to help a dense matching. Multi-vew stereo methods are also often exploited to produce point clouds. Liu *et al.* [Liu et al. 2010] present a multi-view stereo algorithm for free-viewpoint video, where a visual hull is fused with the feature points obtained via a stereo matching. Vlasic *et al.* [Vlasic et al. 2009] present a high-resolution surface capture system which uses phtometric stereo for 3D reconstruction. It requires a large lighting dome to provide a series of novel spherical lighting configurations.

In recent years, several multi-camera 4D video capture solutions that use a dense set of cameras have been presented. Collet *et al.* [Collet et al. 2015] present an end-to-end solution to create high-quality free-view point video. Their system consists of 106 cameras including RGB and IR cameras. A multimodal multi-view stereo fusing RGB, IR and silhouette information is used for accurate point cloud reconstruction. Schreer *et al.* [Schreer et al. 2020, 2019] develop a system with 32 cameras for producing volumetric video. The point cloud is constructed by fusing stereo pairs using a patch sweeping stereo algorithm. Leroy *et al.* [Leroy et al. 2017] present a 4D shape reconstruction solution, of which 68 RGB cameras are used. Their reconstruction method also exploits the integration of volumetric visual hull and a depth-map fusion based stereo.

3 THE SYSTEM AND METHODS

3.1 Image Acquisition and Preprocessing

We present a multi-camera surface capture studio (Fig. 1) with the aim of producing highly realistic 4D video for various human performance scenes. A multi-camera system with 48 RGB cameras is setup in a green paint room with overhead lighting. A multi-camera sequence is acquired for each performance scene, and will be processed in a pipeline to generate a 4D video.

Our pipeline first performs foreground segmentation to obtain silhouettes of every camera. In our current studio we do not strictly adjust the uniformity of the ambient lighting, and there will be shadows, dark areas or highlights on the foreground regions in the views of some cameras at certain perspectives. This makes the use of green screen keying methods such as Chroma Key often wrongly segment some areas of the foreground subjects into

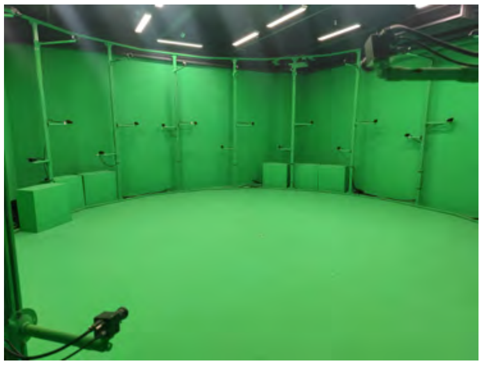

Figure 1: Our surface capture studio setup.

the background. For robust foreground segmentation, our method combines chroma key with a pixelwise statistic background model learned from a group of background frames. A reference background image is created from N background frames where no foreground subject appears. Each pixel i of the reference background image is given a vector $M(i) = [E(i), V(i), G(i), H(i)]$. Each component is defined as below: $E(i)$ and $V(i)$ represent the mean and standard deviation of i^{th} pixel's RGB values computed over the captured N background frames, where $E(i) = [u_r(i), u_g(i), u_b(i)]$, $V(i) = [\delta_r(i), \delta_g(i), \delta_b(i)]$; $G(i)$ denotes a variance of the difference between the green and the gray value, it is defined as:

$$G(i) = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(I_g^t(i) - I_{gray}^t(i)\right)^2}$$

where $I_k^t(k = r, g, b)$ is the t^{th} background image's RGB values, I_{gray}^t denotes the corresponding gray value; $H(i)$ denotes the mean of the HSV component: $H(i) = [u_h(i), u_s(i), u_v(i)]$. For each pixel i of one frame to be processed, the following two measures are computed:

$$\begin{cases} \Delta g(i) = \dfrac{I_g(i) - I_{gray}(i)}{G(i)} \\ \Delta b(i) = \sqrt{\displaystyle\sum_{r,g,b} w_k \left(\frac{I_k(i) - u_k(i)}{\delta_k(i)}\right)^2} \end{cases} \tag{1}$$

where w_k is weighting parameters for RGB color component, which we set $w_r = 0.3$, $w_g = 0.4$, and $w_b = 0.3$ in our experiments. Δg measures the difference between the green and gray value, which in fact is a typical Chroma key difference, and Δb measures the total difference of RGB colors. The potential foreground pixels are detected if the two measures such that: $\Delta g(i) < g_{th}$ and $\Delta b(i) > b_{th}$, where g_{th} and b_{th} are two predefined constant thresholds. In this way, we collect a set of candidate foreground pixels. To remove the noise effect, the average of the two values computed from the local neighbor pixels covered by a small window is taken as the observation measures.

The detected potential foreground pixels often contains shadow areas. The shadow pixels have the following characteristics: the brightness will be dark relative to the background pixels, but the relative changes in hue and saturation will be small. Based on this observation, we find a simple hard threshold method [Cucchiara et al. 2003] works well for our studio. Specifically, shadow pixels can be detected if the HSV values of the pixels such that:

$$\alpha_v \leq \frac{I_v(i)}{u_v(i)} \leq \beta_v \wedge |I_h(i) - u_h(i)| \leq \tau_h \wedge |I_s(i) - u_s(i)| \leq \tau_s \tag{2}$$

where α_v, β_v, τ_h, τ_s are constant thresholds.

3.2 Point Cloud Reconstruction

After foreground segmentation, a 3D point cloud of the captured performance is reconstructed. Inspired by the work [Collet et al. 2015; Leroy et al. 2017; Vogiatzis et al. 2007], we present a robust and accurate point cloud reconstruction for studio capture. The reconstruction mainly consists of three steps. The first step employs a volumetric Shape from Silhouette (SFS) method to obtain a base voxel surface S_b. The next step performs depth map refinement and fusion to obtain a refined shape $\mathcal{S}_{mvs}$. Finally, the point cloud model $\mathcal{S}_{mer}$ is generated by merging the two models $\mathcal{S}_b$ and $\mathcal{S}_{mvs}$.

3.2.1 Volumetric Reconstruction. The volumetric SFS starts by dividing a volume space V with predefined size and position into a voxel set. Each voxel is then tested whether it belongs to the foreground volumetric shape $\mathcal{S}_v$. This is a computational cost procedure [R.Szeliski 1993; S.M.Seitz and C.R.Dyer 1999]. For reducing computational complexity, our method approximates the projection of a voxel as a constant shape formed by pixels around the projected voxel center. Since the occupancy test for each voxel is independent of each other, our system implements the volumetric reconstruction on GPU and attains real-time reconstruction. We obtain the base surface shape $\mathcal{S}_b$ as a point cloud that consists of only surface voxels of the volumetric shape $\mathcal{S}_v$.

Before point cloud reconstruction starts, the following data computation is performed:

- For each surface voxel v_i on $\mathcal{S}_b$, its normal vector n_{v_i} and its visibility $C_{vis}(v_i, \mathcal{S}_b)$ are computed. The visibility $C_{vis}(v_i, \mathcal{S}_b)$ denotes a set of camera views that v_i is visible, and it can be constructed if the corresponding depth maps are available.
- We reconstruct the depth map D_{sil}^k for each kth view by first reconstruction of a triangle mesh using Marching Cubes on the base surface and then projecting the mesh to D_{sil}^k using a depth buffer rasterization.
- 3D distance transform map (DT) and closest feature transform (CFT) map of the volume V is also created, of which the surface voxels are taken as the feature voxels. The 3D DT map is an assignment to each voxel $v_i \in V$ of the distance value between v_i and the closest feature (surface) voxel of v_i. The CFT map is an assignment to each voxel $v_i \in V$ of the identity of the closest feature voxel in the volume. We adopt the method of [Calvin R.Maurer et al. 2003] to compute the exact 3D DT and CFT of the volume. This algorithm is time linear to the number of voxels and can be easily implemented on GPU.
- An inner boundary surface shape $\mathcal{S}_{in} = \{v_i | DT(v_i) = D_{in}, v_i \in \mathcal{S}_v\}$ is also constructed, where $DT(v_i)$ gives the distance value of v_i in the DT mapping, D_{in} is a predefined constant threshold. Our method assumes that the exact surface model lies within the base surface $\mathcal{S}_b$ and the inner boundary surface $\mathcal{S}_{in}$.

3.2.2 Depth Map Refinement. With the volumetric data ready, our method estimates a new depth for each pixel in D^k_{sil} to attain a more accurate depth map D^k_{mvs}. The estimation is carried out as a coarse search and refinement procedure. The search step is to find a coarse depth with maxima photometric consistency along the viewing line, and the refinement step is to locate an optimal position via a local optimization.

Let's assume the viewing line of a pixel with valid depth in D^k_{sil} intersects with the surface $\mathcal{S}_b$ and $\mathcal{S}_{in}$ at two points $p(d_b)$ and $p(d_{in})$ separately. The optimal depth is assumed within the interval $[d_b, d_{in}]$. To search a coarse depth efficiently, the depth interval is sampled uniformly with the voxel size as the step size, *i.e.*, the sampling depth value $d \in \{d_j | j = 0, 1, \cdots, J\}$, where $d_0 = d_b$ and $d_J = d_{in}$. A coarse depth value is then found as the sample point that has maxima photometric consistency.

The photoconsistency measure $\rho_k\left(p(d_j)\right)$ of a sample point $p(d_j)$ is evaluated by comparing its projections in the images where it is visible. The exact visibility information is not available without the true shape model. Using the base surface to provide visibility information for a surface within a small distance is acceptable [G.Vogiatzis et al. 2005]. Let v_c be the voxel on the base surface $\mathcal{S}_b$ closest to the sample point $p(d_j)$, where the closest voxel's identity c can be assigned via the CFT map. The visibility $C_{vis}\left(p(d_j)\right)$ is assigned as the union of the visibility of the voxel v_c and the point $p(d_0)$:

$$C_{vis}\left(p(d_j)\right) = C_{vis}\left(v_c, \mathcal{S}_b\right) \cup C_{vis}\left(p(d_0), \mathcal{S}_b\right) \tag{3}$$

The drawback of using small window-based correlation measure such as Normalized Cross Correlation (NCC) [Furukawa and Ponce 2010; G.Vogiatzis et al. 2005; M.Seitz et al. 2006; Schonberger and Zhang 2016] is that reliable image observation requires of image textures having sufficient quality. In addition, the computation cost of using NCC would be high in our surface capture framework. To increase robustness and efficiency, we use an efficient dense descriptor DAISY [Tola et al. 2010] for computing photoconsistency measure. DAISY is a very efficient local image descriptor that can be computed at every single image pixel. It has shown competitive performance in dense matching for wide-baseline multi-view stereo [Tola et al. 2010, 2011; Trulls et al. 2012]. Given an image pair $\Upsilon_i(I_k, I_l)$ which is taken from the visibility set $C_{vis}\left(p(d)\right)$, where k is the reference camera view of depth estimation, and l is arbitrary another camera view of which the sample point $p(d)$ is visible. Let $\mathbf{D}_k\left(p(d_j)\right)$ and $\mathbf{D}_l\left(p(d_j)\right)$ represent the DAISY descriptors at locations obtained by projecting the 3D sample point $p(d_j)$ onto image k and l. The photometric dissimilarity $D_{kl}\left(p(d_j)\right)$ between the two descriptors is computed as:

$$D_{kl}\left(p(d_j)\right) = \frac{1}{S}\sum_{s=1}^{S} \|\mathbf{D}^s_k - \mathbf{D}^s_l\|_2 \tag{4}$$

where $\mathbf{D}^s_k$ is the sth histogram in $\mathbf{D}_k\left(p(d_j)\right)$, and S is the number of histograms used in the descriptor. The photometric dissimilarity of all pairs $\Upsilon_i(I_k, I_l)$ are computed. Then $\rho_k(p(d_j))$ is obtained as:

$$\rho_k\left(p(d_j)\right) = exp\left(-\frac{u}{M-1}\sum_{\substack{l \in C_{vis}(p(d_j)) \\ l \neq k}} w_{kl} D_{kl}\left(p(d_j)\right)\right) \tag{5}$$

where M is the number of camera views in $C_{vis}\left(p(d_j)\right)$, and u is a constant parameter that controls the distribution. w_{kl} is a normalized weighting parameter given by the cosine angle between the optical axes of the camera view k and l. In order to reduce noise effect, the average measure of a adjacent sample points are taken as the valid measure:

$$\rho_k\left(p(d_j)\right) = \frac{1}{a}\sum_{j' \in \mathcal{N}(j,a)} \rho_k\left(p(d_{j'})\right) \tag{6}$$

where $\mathcal{N}(j, a)$ represents a neighborhood of size a around at the jth sample point, and through our experiments $a = 3$ is used.

Since the base surface is an approximate shape, it makes d_0 or its few neighbors often having higher photometric consistency than those penetrating deeper into the volumetric shape. To further reduce computational cost, the search follows the rules listed below:

- if $\rho_k\left(p(d_0)\right) > \rho_\alpha$ and $\forall j,\ j \in \{1, 2, \cdots, j_\alpha\}$, $\rho_k\left(p(d_0)\right) > \rho_k\left(p(d_j)\right)$, the search will stop at d_{j_α}.
- if $\rho_k\left(p(d_0)\right) < \rho_\beta$ and $\exists j,\ j \in \{1, 2, \cdots, j_\beta\}$, $\rho_k\left(p(d_j)\right) > \rho_\alpha$, the search will stop at d_{j_β} such that $\sum_{j'=j_\beta-3}^{j_\beta} \rho_k\left(p(d_{j'})\right) < \rho_\gamma$.
- If neither the above earlier stop condition is met, the search will exhaustively explore a fixed distance starting from d_0 along the viewing line until a maxima distance threshold Δd_{max} is reached.

In our setting, $\rho_\alpha = 0.75$, $j_\alpha = 5$,$\rho_\beta = 0.4$ and $\rho_\gamma = 1.5$, $\Delta d_{max} = 10cm$. To this end we obtain a coarse depth d_{ini} which corresponds to the sample point having maxima photometric consistency. Given the obtained point $p(d_{ini})$, assuming its closest voxel is v_j, and its normal is assigned the normal of v_j, *i.e.*, n_{v_j} which is already available (see Sect. 3.2.1), we refine its position and its surface normal by maximizing the photometric consistency.

To simplify computations, the position is constrained to lie on the viewing line and the normal vector is represented by yaw ad pitch angles. We use a conjugate gradient method to optimize the parameters, similar as the photoconsistency enforcing optimization adopted in [Furukawa and Ponce 2010]. With a refined depth map for every camera view, a 3D point cloud can be reconstructed by fusing all depth maps [Galliani et al. 2015; Schonberger and Zhang 2016]. A 3D point that is consistent with at least two camera views is kept as a reliable estimate. For all retained points, the 3D positions and normals are averaged over the consistent views to further suppress noise. We now obtain a refined point cloud model $\mathcal{S}_{mvs}$.

3.2.3 Point Clouds Merging. Although accurate reconstruction is obtained, it is experimentally found that $\mathcal{S}_{mvs}$ often has no or little points in the area of the subject surface with poor textures or the areas that no camera view is visible. For example, the hair often has small holes (See Fig. 2). We need to further fuse the two point cloud models to enhance reconstruction robustness. The following merging strategy is adopted: (1) for a voxel on $\mathcal{S}_b$, if its space contains any 3D point from $\mathcal{S}_{mvs}$, the voxel is removed; otherwise, it is

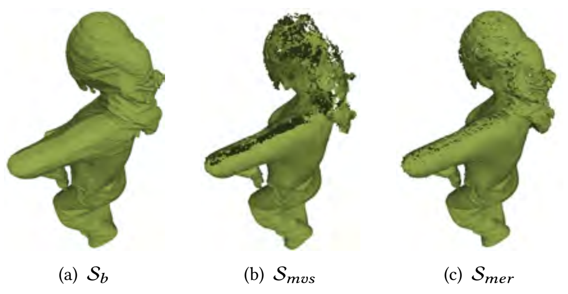

Figure 2: Example point cloud of the base surface $\mathcal{S}_b$, the refined surface $\mathcal{S}_{mvs}$ and the final merged surface $\mathcal{S}_{mer}$.

retained; (2) the reconstructed points in $\mathcal{S}_{mvs}$ whose photoconsistency measure exceeds a predefined threshold are considered to be reliable model points, and the visibility of such points is also considered to be accurate. Therefore, there should be no voxel points on the optical rays between such points and the optical center of the visible views, otherwise they will be occluded. Thereby, these voxels will be removed. After this simple merging step, we get a final point cloud model $\mathcal{S}_{mer}$.

3.3 Meshing and Texturing

For each captured frame, our method reconstructs a mesh model from the obtained point cloud $\mathcal{S}_{mer}$ using Poisson surface reconstruction algorithm [Kazhdan and Hoppe 2013]. The resulting mesh model is then simplified and cleaned such that its face number set to be about 65000 for desktop applications and about 25000 for mobile device use. Because each mesh is reconstructed individually, the mesh connectivity is changed every frame, which would cause face and texture jittering in 4D video. Keeping a stable mesh connectivity over a ground of frames not only can enhance fidelity but also is better for 4D video compression and editing. To attain temporal topology consistency, we manually divide the mesh sequence into a number of groups of frames, each of which will have the same connectivity. For each group, a key frame is chosen via a visual interaction tool and is then registered to each mesh in the group. Our system implements a non-rigid registration method similar as [Morgenstern et al. 2019] which uses a coarse-to-fine ICP for pairwise registration.

After all meshes are processed and temporal connectivity is maximally attained, meshes need texturing which is vitally important for creating realistic surface models. There is a lot of research work on texturing for multi-view based 3D reconstruction such as [Gal et al. 2010; Lempitsky and Ivanov 2007; Waechter et al. 2014]. A typical procedure for texturing a 3D model is to first find an optimal view for each face yielding a preliminary texture and then stitch and fuse the adjacent patches to achieve seamless textures. We use the texturing algorithm of [Waechter et al. 2014]. Our implementation is different from the original method in two ways. (1) since all cameras in our studio are about the same distance from the center, only the angle between viewing direction and face normal is taken into the data term used in optimization of finding optimal view(s); (2) the global adjustment is avoided and only the local Poisson editing algorithm [Perez et al. 2003] is used to blend the seams. Experiments show that due to accurate camera calibration and 3D reconstruction that can be obtained in our studio capture environment, this method can effectively obtain high quality seamless textures.

3.4 4D Video Compression

The obtained mesh model sequence with textures cannot be directly used for applications since it contains a huge amount of data. It needs to be compressed into a single 4D video file in a specific format. The 4D video can be viewed in arbitrary perspective and should support real-time decoding, rendering as well as playback controls. This is similar to generating video files from image sequences, but currently there is no unified file format standard for 4D video [Collet et al. 2015; Schreer et al. 2020]. We design a 4D video file format referred as FDV and develop an offline encoding tool for generating FDV 4D video from the reconstructed mesh sequences. A FDV file contains compressed data of three sources, including mesh, texture, and audio. The meshes has been splitted into groups of frames where each group has the same geometry connectivity, and hence for one group only the vertex position data needs to be compressed at every frame. A popular mesh compression algorithm Draco [Google 2017] is used. To better accommodate GPU rendering, our method provides multiple encoding types for texture compression where typical GPU compressed texture formats for different platforms such as DXT, ASTC, ETC, RGB24, *etc.* are included. This allows developers to use compact in-memory textures, with optimized memory access for faster and more efficient rendering.

4 IMPLEMENTATION AND RESULTS

Our surface capture studio (Fig. 1) is setup in a green paint room with size as $8m \times 7.5m \times 4.5m$, as shown in Fig. 1. The system consists of 48 RGB cameras of which every 6 cameras are connected to a workstation via USB3 connection. The cameras are fixed uniformly on a firm steel frame which is formed in a cylinder of height about $3m$ and radius $3.5m$. All cameras are controlled to work synchronously to capture scene image sequences with frame rate as 30 fps and image resolution as 2448×2048. The capture cylinder volumne where the subjects do performances is about $3m$ in diameter and $2.8m$ high. Fluorescent light tubes are placed on the room ceiling to make ambient lighting for the capture. Fig. 3 gives some example images of the captured sequences.

For a multi-camera based studio, accurate and fast calibration of the cameras is important. One typical method is to first design a specialize 3D calibration object with precisely known checkerboard patterns, such as the calibration object used in [Collet et al. 2015], and then capture images with the calibration object in several different positions for computing the intrinsic and extrinsic parameters of each camera. The difficult of making such accurate calibration object and tedious of calibration procedure urge us to use a more convenient and cheap way of calibration. We use a simple and efficient bundle adjustment based calibration method [Furukawa and Ponce 2009]. This method requires only a rough initial estimate of camera parameters and exploits a patch-based multi-view stereo algorithm PMVS [Furukawa and Ponce 2010] to reconstruct a rough surface model. The estimated rough surface geometry and visibility information help matching image features, which are used as

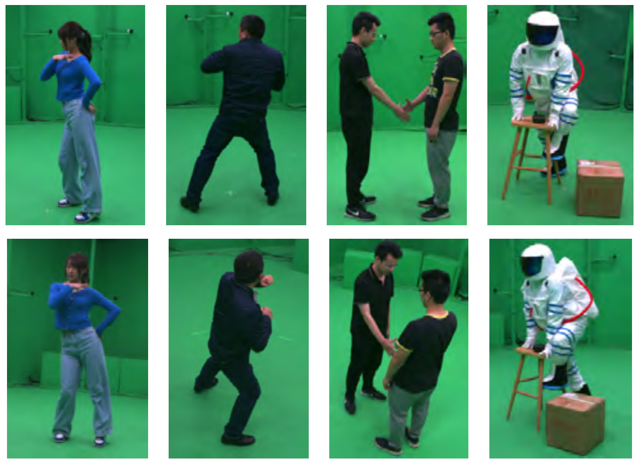
Figure 3: Example multi-camera images of captured scenes, where only two camera views for each sequence are shown.

the input to a bundle adjustment algorithm for tighten up camera parameters. In our implementation, the initial estimate is attained by a simple two-step calibration method [Zhang et al. 2011] which does not require of precisely made calibration object. Through our experiments, we find this initial estimate step only needs to be performed one time, and accurate calibration can be achieved with the estimate even when some cameras are slightly disturbed.

In our method, the segmented silhouettes are intersected to form a voxel-based visual hull which provides constraints as well as an initial estimate for accurate point cloud reconstruction. Hence accurate and robust segmentation of foreground is helpful for producing high quality 4D videos. We need to set appropriate values for these thresholds including g_{th}, b_{th}, α_v, β_v, τ_h and τ_s for each camera. Since our cameras are fixed, these thresholds only need to be set once before capturing scenes for the same camera. The setting can be done conveniently via GUI adjustment. Fig. 4 gives example results of foreground segmentation. Experiments demonstrate that our method can effectively detect shadow pixels and provide accurate silhouette images.

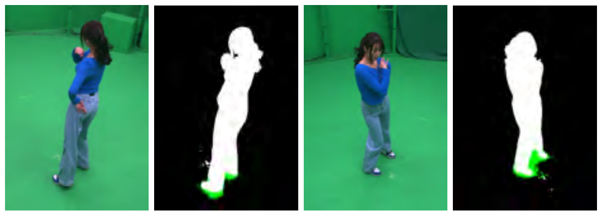
Figure 4: Example foreground segmentation result.

The accuracy of point cloud reconstruction significantly affects the quality of 4D video. To be practical and robustness, the capture system should support a wide range of performance scenes, for example including single or multiple subject interaction scenes. There should be no special requirements of the appearance of the subjects

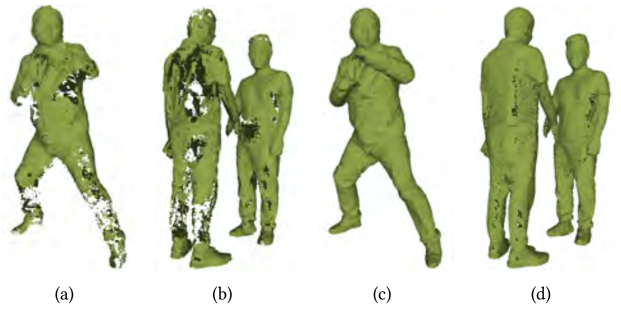

Figure 5: (a-b) Point cloud models obtained by Colmap [Schonberger and Zhang 2016]; (c-d) The corresponding point cloud models reconstructed by our method.

to be captured, *e.g.*, the reconstruction can deal with the objects with solid color or weak textures. To demonstrate the accuracy of our reconstruction, we captures groups of sequences of different scenes with textureless objects and multiple subject present (See Fig. 3). Due to lack of sufficient texture observations, the photometric consistency based MVS methods often cannot deal with regions of textureless, resulting holes or erroneous reconstruction. Colmap is a classic MVS algorithm based on manual feature 3D reconstruction. It can be adapted to various scene reconstruction, and its high-precision performance has also been widely verified. Fig. 5 gives example result of using Colmap algorithm [Schonberger and Zhang 2016] on three captured sequences. It is shown that in our studio capture Colmap does not deal well with the scenes with textureless subjects. Our point cloud reconstruction method effectively solves the problem of holes, and shows good performance in the consistency and accuracy of the reconstructed models.

Texture mapping quality will greatly affect the final model fidelity. We validate texture fidelity by comparing rendering views against the original images. Fig. 6 compares one example of ground-truth view with its enlarge region and the corresponding rendered views. The ground-truth view is the original captured image masking using the segmented silhouette. It is seen that good texture fidelity is achieved. We also implement the texturing method used in [Collet et al. 2015] which simply uses normal-weighted blending of non-occluded images. We find that this method would often yield seams or erroneous textures in our studio (see Fig. 6(b) and Fig. 6(e)). This may due to the less cameras used in our system where many faces may have not sufficient visible views that can be used for the simple blending.

Based on the 4D video codec (see Sect. 3.4), we developed a 4D video player to browse and play the FDV 4D video files. The player supports real-time rendering and playback controls of the 4D video, and supports mouse interaction to view from any perspective angle. Plugins of Unity/Unreal are also developed for easy use and integration of FDV assets into XR applications. Fig. 7 gives illustration of our 4D video player and the Unity plugin.

The 4D content of human performances has wide range of potential applications in various fields. We have applied our 4D video data in several applications. For example, as shown in Fig. 8, the

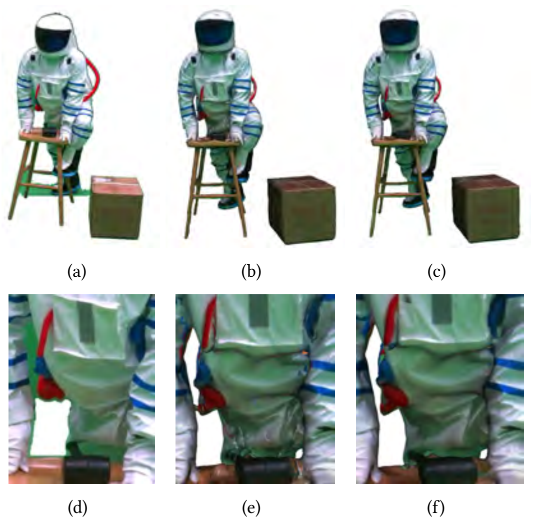

Figure 6: The ground-truth view (a) and its enlarge region (d), where the view is obtained by masking the original image using the segmented silhouette; (b) and (e): the result of using the normal-weighted blending based texturing that is used in [Collet et al. 2015]; (c) and (f): the result of the texturing method implemented in our system.

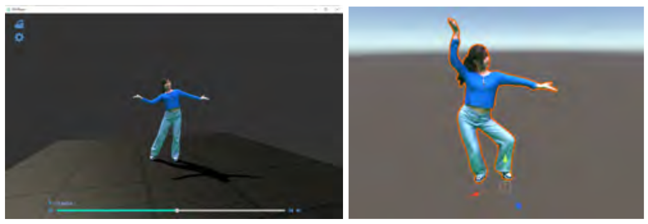

Figure 7: Illustration of the 4D video player (above) and the Unity plugin (below).

4D videos can be easily integrated in AR applications with the developed 4D video plugins. Fig. 9 shows another mobile application where the user can use fingers to rotate and view the 4D characters from any perspective. Fig. 10 gives more example frames of our 4D reconstruction sequences.

Figure 8: Example integration of our 4D video data for a AR application running on Apple phone

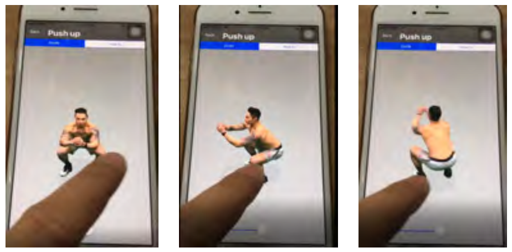

Figure 9: Workouts application using our 4D video data with finger-touching control.

5 CONCLUSION

In this work, a surface capture system for capturing highly realistic and detailed 4D content of human performances is introduced. The system implements a robust processing pipeline to produce high quality 4D videos. A new multi-view point cloud reconstruction method is presented, which fuses silhouettes and multi-view stereo to attain high accurate reconstruction of the scene. It can be used for challenging reconstruction scenes such as the performances with textureless objects or interaction of multiple subjects. The reconstructed mesh sequence is then under mesh post-processing and texture mapping to produce a final mesh sequence with consistent geometry connectivity and realistic appearance. Experiments show that our surface capture system can support a wide range of performance scenes, and can produce high realistic 4D videos. For future work, the automatic key frame selection and non-rigid mesh registration methods will be further studied.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. The work is partially supported by the National Natural Science Foundation of China under Grant No.: 62071478, the Foundation of Key Laboratory of National Defense Science and Technology of Human Factors Engineering (614222210401), and the Foundation of China Astronaut Research and Training Center (2022SY54B0605).

REFERENCES

Jr. Calvin R.Maurer, Rensheng Qi, and Vijay Raghavan. 2003. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 2 (2003), 265–270.

Joel Carranza, Christian Theobalt, Marcus A.Magnor, and Hans-Peter Seidel. 2003. Free-viewpoint video of human actors. *ACM Transactions on Graphics* 22, 3 (2003), 569–577.

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics* 34, 4 (2015), 1–13.

Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. 2003. Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on pattern analysis and machine intelligence* 25, 10 (2003), 1337–1342.

Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, and Hans-Peter Seidel. 2008. Performance Capture from Sparse Multi-view Video. *ACM Transactions on Graphics* 27, 3 (2008), 1–10.

M. Dou, S. Khamis, Y. Degtyarev, S. R. Fanello P. Davidson, S. O. Escolano A. Kowdle, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. 2016. Fusion4D: real-time performance capture of challenging scenes. *ACM Transactions on Graphics* 35, 4 (2016), 1–13.

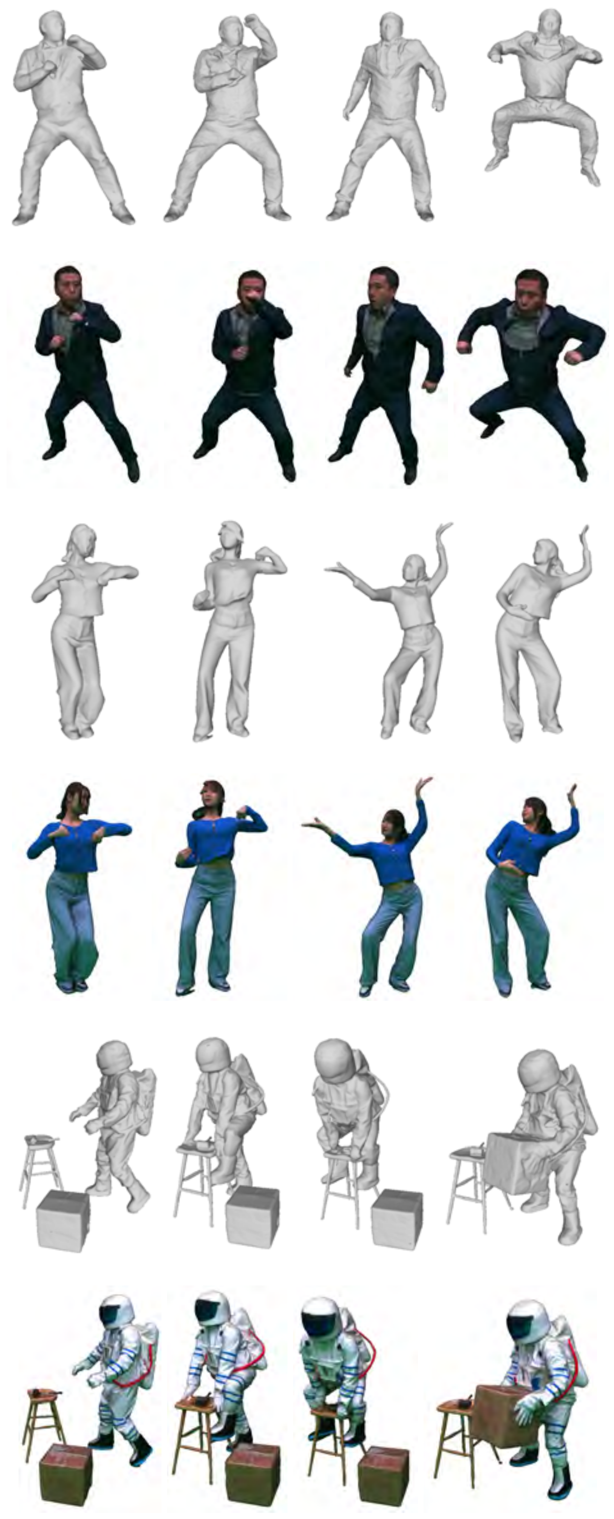
Figure 10: Example frames of 4D reconstruction sequences.

R. Du, M. Chuang, W. Chang, H. Hoppe, , and A. Varshney. 2019. Montage4D: Real-time Seamless Fusion and Stylization of Multiview Video Textures. *Journal of Computer Graphics Techniques* 8, 1 (2019).

Yasutaka Furukawa and Jean Ponce. 2009. Accurate Camera Calibration from Multi-view Stereo and Bundle Ajustment. *International Journal of Computer Vision* 84 (2009), 257–268.

Yasutaka Furukawa and Jean Ponce. 2010. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 8 (2010), 1362–1376.

Ran Gal, Yonatan Wexler, Eyal Ofek, Hugues Hoppe, and Daniel Cohen-Or. 2010. Seamless Montage for Texturing Models. *Eurographics* 29, 2 (2010), 1–7.

Juergen Gall, Carsten Stoll, Edilson de Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. 2009. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*.

Silvano Galliani, Katrin Lasinger, and Konrad Schindler. 2015. Massively Parallel Multiview Steropsis by Surface Normal Diffusion. In *ICCV*.

Google. 2017. Introducing Draco: compression for 3D graphics. https://opensource.googleblog.com/2017/01/introducing-draco-compression-for-3d.html. Accessed: 2022-10-06.

G.Vogiatzis, P.H.S Torr, and R.Cipolla. 2005. Multi-view Stereo via Volumetric Graph-cuts. In *CVPR*.

Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics* 32, 3 (2013), 1–13.

Victor Lempitsky and Denis Ivanov. 2007. Seamless Mosaicing of Image-based Texture Maps. In *CVPR*.

Vincent Leroy, Jean-Sebastien Franco, and Edmond Boyer. 2017. Multi-view Dynamic Shape Refinement using Local Temporal Integration. In *ICCV*.

Yebin Liu, Qionghai Dai, and Wenli Xu. 2010. A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE Transactions on visualization and computer graphics* 16 (2010), 407–418.

Wieland Morgenstern, Anna Hilsmann, and Peter Eisert. 2019. Progressive Non-rigid Registration of Temporal Mesh Sequences. In *CVMP*.

Steven M.Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. 2006. A Comparison and Evaluation of Multi-view Stereo Reconstruction Algorithms. In *CVPR*.

Ahmed N., Theobalt Ch., Rossl Ch., Thrun S., and Seidel H-P. 2008. Dense correspondence finding for parametrization-free animation reconstruction from video. In *CVPR*.

Rohit Pandey, Anastasia Tkach, and Shuoran Yang. 2019. Volumetric capture of humans with a single RGBD camera via semi-parametric learning. In *CVPR*.

Patrick Perez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Transactions on Graphics* 22, 3 (2003), 313–318.

R.Szeliski. 1993. Rapid octree construction from image sequences. *Computer vision, graphics and image processing* 58, 1 (1993).

Johannes L. Schonberger and Enliang Zhang. 2016. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*.

Oliver Schreer, Ingo Feldmann, Peter Kauff, Peter Eisert, Danny Tatzelt, and Cornelius Hell. 2020. Lessons learned during one year of commercial volumetric video production. *SMPTE Motion Imaging Journal* 129, 9 (2020).

Oliver Schreer, Ingo Feldmann, Sylvain Renault, and Marcus Zepp. 2019. Capture and 3D Video Processing of Volumetric Video. In *ICIP*.

S.M.Seitz and C.R.Dyer. 1999. Photorealistic Scene Reconstruction by Voxel Coloring. *International Journal of Computer Vision* 35, 2 (1999), 151–173.

Jonathan Starck and Adrian Hilton. 2007. Surface Capture for Performance based Animation. *IEEE Computer Graphics and Applications* 27, 3 (2007), 21–31.

Engin Tola, Vincent Lepetit, and Pascal Fua. 2010. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 5 (2010), 815–830.

Engin Tola, Christoph Strecha, and Pascal Fua. 2011. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine vision and applications* 23 (2011), 903–920.

Eduard Trulls, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2012. Spatiotemporal Descriptor for wide-baseline stereo reconstruction of non-rigid and ambiguous scenes. In *ECCV*.

Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic. 2008. Articulated Mesh Animation from Multi-view Silhouette. *ACM Transactions on Graphics* 27, 3 (2008), 1–9.

Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. 2009. Dynamic Shape Capture using Multi-View Photometric Stereo. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 28, 5 (2009).

George Vogiatzis, Carlos Hernandez Esteban, Philip H.S.Torr, and Roberto Cipolla. 2007. Multi-view Stereo via Volumetric Graph-cuts and Occlusion Robust Photo-Consistency. *IEEE PAMI* 29, 12 (2007), 2241–2246.

Michael Waechter, Nils Moehrle, and Michael Goesele. 2014. Let there be color! Large-scale texturing of 3D reconstructions. In *ECCV*.

Zheng Zhang, Hock Soon Seah, Chee Kwang Quah, Alex Ong, and Khalid Jabbar. 2011. A multiple camera system with real-time volumne reconstruction for articulated skeleton pose tracking. In *Advances in Multimedia Modeling*.