EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)

Volume 33 (2014), Number 2

4D Video Textures for Interactive Character Appearance

Dan Casas, Marco Volino, John Collomosse and Adrian Hilton

Centre for Vision, Speech & Signal Processing, University of Surrey, United Kingdom

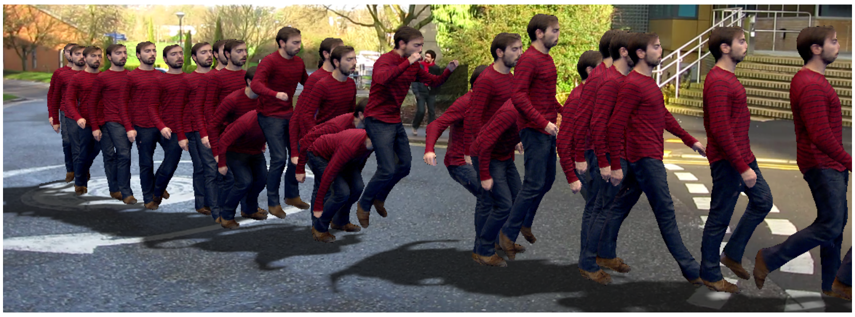


Figure 1: Interactive animation of a character with *4D Video Texture* combining multiple 4D videos for different motions

Abstract

4D Video Textures (4DVT) introduce a novel representation for rendering video-realistic interactive character animation from a database of 4D actor performance captured in a multiple camera studio. 4D performance capture reconstructs dynamic shape and appearance over time but is limited to free-viewpoint video replay of the same motion. Interactive animation from 4D performance capture has so far been limited to surface shape only. 4DVT is the final piece in the puzzle enabling video-realistic interactive animation through two contributions: a layered view-dependent texture map representation which supports efficient storage, transmission and rendering from multiple view video capture; and a rendering approach that combines multiple 4DVT sequences in a parametric motion space, maintaining video quality rendering of dynamic surface appearance whilst allowing high-level interactive control of character motion and viewpoint. 4DVT is demonstrated for multiple characters and evaluated both quantitatively and through a user-study which confirms that the visual quality of captured video is maintained. The 4DVT representation achieves >90% reduction in size and halves the rendering cost.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

1. Introduction

Visual realism remains a challenging goal in the production of character animation. In film visual-effects, realism is achieved through highly skilled modelling and animation, commonly using captured video as a reference. Static texture maps extracted from captured imagery are used to provide realistic visual detail, however, this results in a loss of visual realism compared to the captured dynamics appearance.

This limitation has motivated interest in video-based rendering to reproduce the detailed dynamic appearance of real-world scenes. Free-viewpoint video (FVVR) [ZKU*04] enables video-realistic rendering of novel views from multi-view footage, but is limited to replay captured performances.

The inference from multiple view video of structured 4D video models, representing the dynamic 3D surface shape and view-dependent appearance over time, allows greater reuse of the captured performance [dST*, VBMP08, XLS*]. The motion of the 4D models may be modified (via skeletal rigging, or direct surface manipulation), and rendered with the captured video, reproducing a detailed dynamic appearance that conveys a high degree of visual realism.

Character motion may be readily manipulated in such pipelines by blending, concatenating or editing captured sequences. However the *dynamic appearance* of the character remains that of the original motion, baked into the captured video. This limits the extent to which the character motion can be modified whilst maintaining visual realism. Consequently there remains a gap between existing video-based characters in interactive applications and full video-realism.

In this paper we present 4D Video Textures (4DVT), which allow interactive control of motion and dynamic appearance, whilst maintaining the visual realism of the source video. The approach is based on the parametrization of a set of 4D video examples for an actor performing multiple related motions, for example a short and long jump (Fig. 3). A video-realistic intermediate motion is produced by aligning and combining the multiple-view video from the input examples. As the character motion changes so does the dynamic appearance of the rendered video, reflecting the change in motion. Key technical contributions of this work are:

1. Efficient representation of captured multi-view video sequences as 4D video texture maps, achieving >90% compression whilst maintaining the visual quality of FVVR.
2. Real-time video-realistic rendering of novel motions combining multiple 4D video textures to synthesise plausible dynamic surface appearance.

We demonstrate video-realistic character animation using 4DVT for a variety of actors each performing a range of motions. A user-study is performed to verify that 4DVT produces plausible dynamics without loss of visual quality relative to the captured video or previous FVVR approaches.

2. Related Work

4D Performance Capture. There has been substantial research over the past decade exploiting multiple view video acquisition of actor performance for video-based rendering. The Virtualized Reality system [KR97] using a 51 camera dome with multi-view stereo reconstruction, first demonstrated the potential for free-viewpoint replay of dynamic scenes. Subsequent research focused on achieving video-realistic free-viewpoint replay by exploiting advanced multi-view depth estimation and view-dependent rendering [ZKU*04, EDDM*]. Parallel research proposed solutions to the problem of obtaining structured temporally coherent *4D video* models from multiple view video, representing dynamic surface shape and appearance over time [dST*, BHKH13, CBI10, VBMP08]. The resulting 4D video representation supports modification of the surface shape with free-viewpoint rendering using the captured multi-view video or derived texture maps to render realistic appearance. *MovieReshape* [JTST10] used a morphable model-based single-view reconstruction to modify actor shape and motion in movie sequences achieving photorealism. These approaches are limited to replaying similar motions to the captured actor performance as video-based rendering maps the captured videos onto the surface without modification of the dynamic appearance according to changes in the motion.

Current multiple view reconstruction techniques using stereo correspondence are limited to acquisition of mid-resolution surface shape. Research on acquisition of fine-scale dynamic detail includes techniques such as shape from shading and photometric stereo [BHV*11, LWS*, WVL*11]. Refined surface shape and normal acquisition, together with illumination estimation, have been employed to estimate surface reflectance from multiple view video [TAL*07, LWS*]. By contrast, we achieve video-realistic dynamic appearance that matches the user controlled surface motion by combining captured 4D video textures.

Eisemann *et al.* [EDDM*] proposed a view-dependent optic flow alignment to refine the integration of appearance information from multiple view video. This approach achieved high-quality real-time free-viewpoint rendering, avoiding ghosting and blur artefacts which occur due to errors in the shape proxy or camera calibration. However, their approach is limited to extrapolation from a single geometric proxy and corresponding set of multiple view images. Gal *et al.* [GWO*10] use a MRF optimization to recover high-resolution textures of static objects from multiple images. Our representation is temporally optimized across multiple frames, and maintains the view-dependant information.

Video-based Animation. *VideoRewrite* [BCS97] pioneered in video resampling to synthesize novel video sequences for facial animations. Subsequent research [SSE00] demonstrated animation of a variety of dynamic scenes by concatenating segments of the source video with transitions between frames with similar appearance and motion. *Human Video Textures* [FNZ*09] added 3D motion-capture markers in the sequence acquisition stage to identify suitable transitions combining 2D appearance and 3D pose information. These methods achieve 2D video-realistic animation, but are limited to replaying captured data. Recently, photo-realistic animation of clothing has been demonstrated from a set of 2D images augmented with 3D shape information [AHE13].

Character animation by resampling of 4D capture has also been investigated [SMH05, HHS09], achieving visual quality similar to FVVR but limited to replaying segments of captured motions. Xu *et al.* [XLS*] introduced a more flexible approach based on pose matching between a query motion and a dataset of 4D performance capture. Image warping is used to adapt retrieved images to query pose and viewpoint, enabling photorealistic rendering of novel sequences. However this method is not amenable to interactive character animation, which is a unique contribution of our method.

The 4DVT approach introduced in this work combines multiple 4D videos to interpolate the detailed video dynamics. This approaches maintains the dynamic appearance of the video as the motion changes allowing real-time interactive animation with free-viewpoint video-realistic rendering.

3. Overview

In this paper we introduce 4D Video Textures (4DVT), enabling the efficient representation of dynamic appearance in animations generated from 4D performance capture. 4D performance capture reconstructs a set of 4D video sequence $F(t) = \{M(t), V(t)\}$, where $M(t)$ is a mesh sequence where

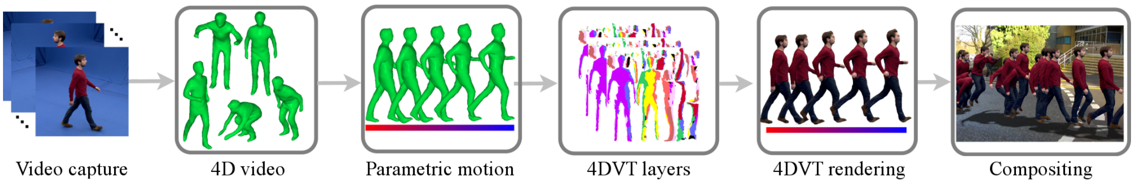


Figure 2: Overview of the proposed framework for *4D Video Textures*

vertices correspond to the same surface point over time, and $V(t) = \{I_j(t)\}_{j=1}^{C}$ is a set of video frames captured from C camera views [dST*, VBMP08]. 4D video is rendered via an image-based process synthesising a free-viewpoint video sequence $I(t, v(t))$ for novel viewpoints $v(t)$ from the set of input videos $V(t)$ using the mesh sequence $M(t)$ as a geometric proxy [BBM*01]. This 'free-viewpoint video rendering' (FVVR) process preserves the visual realism of the source video but it is limited to replay.

4DVT enables interactive control of the dynamic surface appearance together with the viewpoint, whilst maintaining video quality rendering. Combined with previous work in mesh animation from captured surfaces [CTGH13], this enables interactive animation of video-based characters with dynamic appearance that changes accordingly to the motion. 4DVT comprises the following stages, illustrated in Fig. 2:

1. **Multi-camera capture:** Actor performance is captured in a chroma-key studio to ease foreground segmentation.
2. **4D video reconstruction:** Temporally consistent mesh sequences are reconstructed [dST*, VBMP08, BHKH13] to create a set of 4D videos $F_i(t) = \{M_i(t), V_i(t)\}$.
3. **4D mesh animation:** Multiple mesh sequences $\{M_i(t)\}_{i=1}^{N}$ are interactively parametrized to synthesise a novel mesh sequence $M(t, w(t))$ with high-level motion control parameters $w(t)$ [CTGH13].
4. **4DVT representation:** Captured multiple view video $V_i(t)$ is represented in layered texture maps $L_i(t)$ providing a temporally coherent representation of appearance with >90% compression and negligible loss of FVVR quality (Sec. 4). Thus the set of 4D videos is represented efficiently as $F_i'(t) = \{M_i(t), L_i(t)\}$.
5. **4DVT rendering:** Dynamic appearance of multiple captured motions $\{L_i(t)\}_{i=1}^{L}$ are aligned at render time according to the user-controlled viewpoint $v(t)$, motion $w(t)$ and mesh sequence proxy $M(t, w(t))$ to render a novel video sequence $I_{new}(t, v(t), w(t))$ (Sec. 5).
6. **Compositing:** Rendered video sequences are composited with a video-background scene using the associated depth buffer from $M(t, w(t))$ for viewpoint $v(t)$.

Contributions of this work are steps 4 and 5, which allow photo-realistic free-viewpoint rendering with interactive control of motion $w(t)$ and viewpoint $v(t)$. Fig. 3 shows an example of a 4DVT animation of a novel motion with rendered dynamic appearance. A short and a long jump are combined to interactively animate a jump of intermediate distance. The close-up views show that the synthesised novel cloth wrinkling detail maintains the source video-quality.

4. 4D Video Texture Representation

Current free-viewpoint video rendering (FVVR) resamples the captured multiple-view video directly to render replays from novel view points. Real-time blending of multiple views allows photorealistic rendering in the presence of non-Lambertian reflectance and errors in geometry or calibration [ZKU*04]. However, this requires online streaming of the full set of videos from disk to GPU or preloading of the multiple view sequences to texture memory. For a typical 8 HD camera capture, FVVR requires streaming data-transfer rates up to 800MB/s. Foreground masking reduces the bandwidth by 50-80% but the data size limits FVVR to local rendering on high-performance machines.

This section introduces a novel 4DVT representation which resamples the captured multiple view video into a set of texture map layers ordered according to surface visibility. Resampling into the layered texture domain is optimized to preserve spatial and temporal coherence. The resulting 4DVT representation removes the redundancy in the captured video achieving >90% compression over the foreground only masking and reduces rendering to a simple lookup without a perceivable loss of photorealistic rendering quality (Sec. 6.2). The reduced data size and precomputation enabled by the layered representation allows free-viewpoint rendering on relatively low-performance mobile devices.

4.1. 4DVT layered representation

4DVT resamples each multi-view video frame into a set of texture map layers L. Each layer stores the appearance information for each mesh facet in order of camera visibility. For example, the first layer samples texture from the best (most visible) camera view for each facet with subsequent layers sampling texture in order of camera visibility. Multiple view texture is stored in L layers where $L \leq C$ the number of cameras C. In practice we show that for a typical multi-camera setup $L << C$ reducing storage by two order of magnitude with negligible reduction in rendering quality (for example $L = 3$ for $C = 8$).

For each multi-view video frame we generate a layered set of two-dimensional texture maps $L(t) = \{l_k(t)\}_{k=1}^{L}$ at time t. Each layer $l_k(t)$ is contributed to by up to C camera views according to the mesh geometry $M(t)$. To construct layer $l_k(t)$ for a given frame t we first consider the visibility of the mesh geometry, computing a set of depth maps with respect to C camera viewpoints, $\mathcal{C}$. Considering each viewpoint in turn, for each facet of the mesh $f \in M_i(t)$ we compute the angle

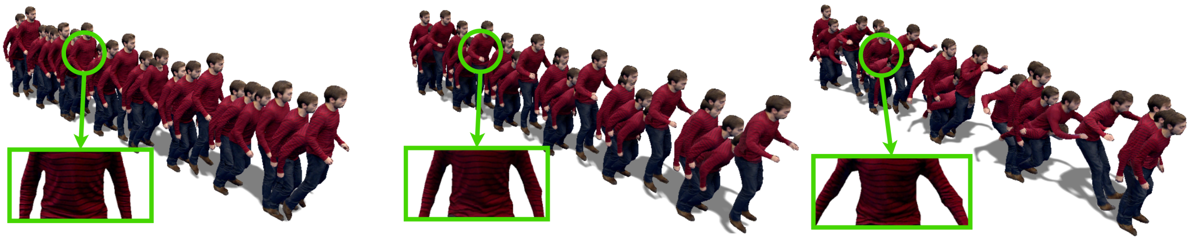


Figure 3: 4DVT: Original 4D videos for short (left) and long (right) jumps. Center: medium jump with synthesized appearance.

between the normal to f (written $\hat{n}(f)$), and the optical axis $\hat{o}(c)$ of the camera $c \in \mathcal{C}$. The visibility of face f for each camera c is ordered based on the penalty score:

$$p(c,f) = \begin{cases} 1-(\hat{n}(f)\cdot\hat{o}(c))^2 & \text{if } f \text{ visible from } c, \\ \infty & \text{if } f \text{ not visible.} \end{cases} \quad (1)$$

Texture map layers l_k for face f are ordered based on increasing penalty score $p(c,f)$ such that the first layer l_1 samples from the camera viewpoint $\hat{o}(c)$ most directly aligned with the face normal $\hat{n}(f)$. Successive layers l_k sample from the next best camera viewpoint. Thus the layered texture map represents the L best views for each face f which are subsequently used for view-dependent rendering. Camera best view selection based on orientation of the viewpoint to the face normal has been widely used as a criteria in free-viewpoint rendering, alternative criteria could also take into account camera sampling resolution for each face.

4.1.1. Optimization of Spatial Coherence

In practice, this naïve approach leads to artefacts due to spatial incoherence in viewpoint selection for neighboring facets (arising due to lighting variation, non-Lambertian reflectance or minor errors in camera calibration or surface reconstruction). Fig. 4a shows assignment of views to the first layer, notice the fragmentation.

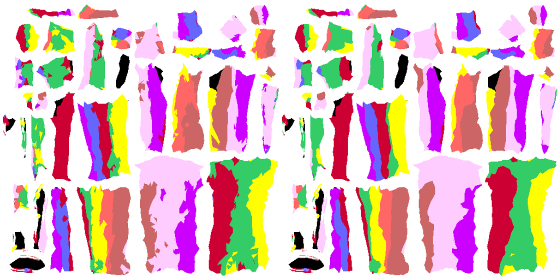
Spatial coherence of the layered texture is achieved by formulating the problem of labelling of facets to viewpoints as a Markov Random Field (MRF) optimisation. An undirected graph $G_{ci} = \{M_i(t), K\}$ is constructed where each facet f is a node, and connecting edges K are non-zero where vertex connectivity exists in $M_i(t)$. We seek a label assignment, mapping $f \in M_i(t)$ to the finite set of viewpoints $c \in \mathcal{C}$. We write this camera label assignment for each face as $Z = \{z_1 ... z_{|M_i(t)|}\}$ with $z_f \in \mathcal{C}$. The suitability of a given mapping is found by minimizing the energy function

$$E(Z) = \sum_{f \in M_i(t)} \Phi_f(z_f) + \sum_{f \in M_i(t)} \frac{1}{|N_f|} \sum_{j \in N_f} \Phi_{fj}(z_f, z_j). \quad (2)$$

where N_f indicates the 1-neighborhood of node $f \in G_{ci}$. The unary term Φ_f indicates the penalty attributed to assigning label z_f to facet f:

$$\Phi_f(z_f) = p(z_f, f). \quad (3)$$

The pairwise term $\Phi_{fj}(z_f, z_j)$ penalizes dis-similarity between f and its immediate neighbor j in $M_i(t)$:



(a) Non-optimized (b) Spatially optimized

Figure 4: Camera maps of an arbitrary frame of JP dataset (for layer 1) indicating the source of video texture for model parts. The MRF optimization yields improved spatio-temporal coherence in the map.

$$\Phi_{fj}(z_f, z_j) = \begin{cases} 0 & \text{if } z_f = z_j, \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

As this pairwise potential is in the form of a Potts model it can be minimized using α-expansion and αβ-swap algorithm [AKT08]. Fig. 4(b) shows a spatially smoothed label assignment, resulting in improved spatial coherence.

4.1.2. Optimization of Spatio-temporal Coherence

A further term may be added to Equation 2 to encourage temporal coherence in the label assignment. In this case optimization is performed over all facets over all time, *i.e.* $G_{ci} = \{M_i(\forall t), E\}$ using a revised energy score $E'(Z)$.

$$E'(Z) = \sum_{\forall t} \left[E(Z) + \frac{1}{|M_i|} \sum_{f \in M_i} \Phi_f(z_{f(t)}, z_{f(t-1)}) \right]. \quad (5)$$

Solution via α-expansion has low-polynomial complexity as the number of nodes in the graph scales, and so increases quickly with both mesh complexity and sequence duration. Sequences with an 8k facet count are processed at 1-2 seconds per viewpoint per frame using a commodity quad-core PC. Results presented in Sec. 6.2 demonstrate that the optimisation of spatio-temporal coherence produces 4DVT with improved compression whilst maintaining rendering quality.

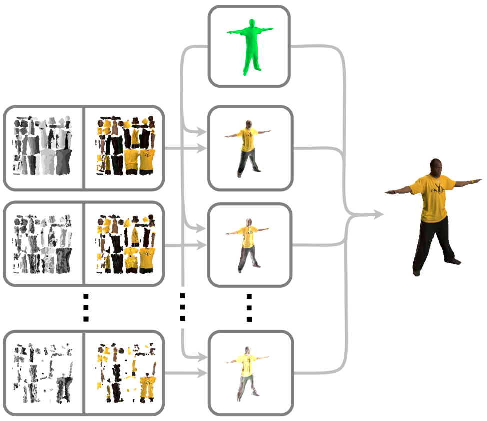


Figure 5: Overview of the view-dependent layered texture map representation: a set of layered textures (left); layer contribution based upon the user's viewpoint (middle); the final view-dependent rendering (right). Using dataset 'JP'.

4.2. Free-Viewpoint Rendering

Spatio-temporal optimization gives a 4D video $F'(t) = \{M(t), L(t)\}$ free-viewpoint rendering for an interactively controlled viewpoint $v(t)$ is reduced to a simple lookup and blending of the texture layers using view-dependent weights. Camera-surface visibility is pre-computed and represented in the layered texture map $L(t)$, eliminating the requirement for evaluation of visibility at render time. Fig. 5 depicts an overview of the pipeline. We show in Sec. 6 that this leads to a halving of computation time and reduction of two orders of magnitude in storage, enabling streaming playback of free-viewpoint 4D video.

5. 4D Video Texture Rendering

The 4DVT representations enable 4D performance capture to be used for video realistic interactive character animation. In this section we introduce a novel rendering approach which combines multiple 4DVTs to change the dynamic appearance in real-time to match the character motion.

The problem is addressed by combining multiple 4DVT sequences $\{F'_i(t)\}_{i=1}^N$ at render time to synthesise a novel video sequence $I(t, w(t), v(t))$ with interactive control of motion parameters $w(t)$ and viewpoint $v(t)$:

$$I(t, w(t), v(t)) = f(F'_1(t), ..., F'_N(t), w(t), v(t)) \qquad (6)$$

where $f(.)$ is a function which combines the source 4D videos according to the specified motion parameters $w(t)$ and viewpoint $v(t)$. The rendered video $I(t, w(t), v(t))$ should preserve the visual quality of both the scene appearance and motion. The 4D mesh sequence shape proxy $M(t, w(t))$ is synthesised by non-linear combination of the set of input mesh sequences $M_i(t)$ [CTGH13]. Motion parameters $w(t)$ allow high-level interactive control of the motion, for example captured input sequences of walking slow, fast and turning would allow walking motion parameters for speed and direction. View-dependent rendering of the output video $I(t, w(t), v(t))$ is based on real-time alignment of the rendered 4D video textures $L_i(t)$ using the shape proxy $M(t, w(t))$ for viewpoint $v(t)$. Fig. 6 presents an overview of this 4DVT rendering process.

5.1. View-dependent 4D Video Rendering

The critical challenge for video quality rendering with control of motion and viewpoint is the combination of multiple 4D videos $\{F'_i(t)\}$ to render a novel output video $I(t, w(t), v(t))$. A naïve approach to render the output 4DVT video $I(t, w(t), v(t))$ for a given set of motion parameters $w(t)$ and viewpoint $v(t)$ is to use the known mesh correspondence to directly transfer the multiple view video for each input mesh sequence $M_i(t)$ to the interpolated proxy shape $M(t, w(t))$ and blend the resulting textures from input multiple motions. However, any misalignment in the underlying geometric correspondence or change in appearance due to the motion will produce blurring and ghosting artefacts. This is shown in Fig. 7(c), where the appearance of a walk and run (Fig. 7(a,b)) have been transferred and blended into a synthesised geometry proxy $M(t, w(t))$ (notice the change in shape). Even with accurate surface reconstruction and alignment the 4D video appearance changes for different motions due to the dynamics of the skin, clothing and hair. A method to accurately align and interpolate the dynamic appearance of the input motions is required. Our online view-dependent alignment and rendering approach, Fig. 7(e), results in a sharp image with interpolated dynamic appearance detail.

Our approach mitigates visual artefacts by inferring an aligned video from the rendered 4D video textures $L_i(t)$ for viewpoint $v(t)$. Unfortunately it is not possible to precompute the alignment between the input multiple view videos $V_i(t)$ or 4D video textures $L_i(t)$ as the surface visibility depends on the pose relative to the camera which will change significantly for different motions. Alignment in the texture domain is non-trivial due to the distortion of the original surface appearance and seams in the texture map.

The set of 4D videos $F'_i(t)$, with corresponding motion parameters $w_i(t)$, for a particular motion class are embedded in the motion parameter space $w(t)$. Video of novel motions is then rendered by interpolating between the set of videos according to the motion parameters $w(t)$ and viewpoint $v(t)$. A view-dependent approach is introduced to align the appearance for video sequences with different surface dynamics. This alignment allows realistic interpolation of dynamic appearance to synthesize novel motions whilst maintaining the visual quality of the output video sequence.

View-dependent rendering with parametrized motion control requires online combination of the view-dependent renders for individual 4D videos $F_i(t, w_i(t))$. We achieve this by online alignment using optic flow and blending according to the motion parameters of view-dependent appearance. A proxy mesh sequence $M(t, w(t))$ for the given motion parameters $w(t)$ is synthesized [CTGH13]. As the same mesh $M(t, w(t))$ and viewpoint $v(t)$ are used to render each input motion the rendered views share the same geometry and

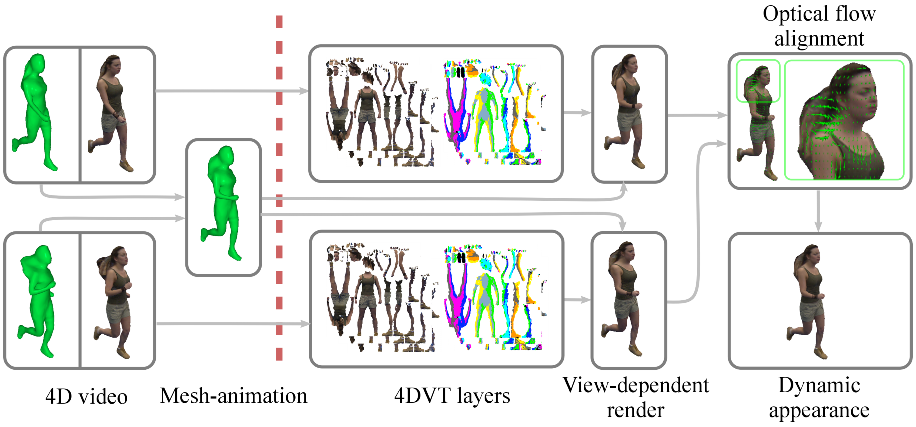


Figure 6: Overview of the proposed 4DVT rendering pipeline. Novel contribution on right of the red line.

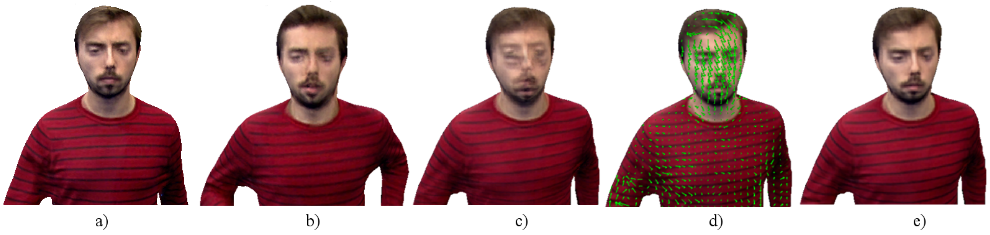


Figure 7: 4DVT rendering: (a),(b) original rendered 4D video for a walk and run sequence for view $v(t)$. Note the difference in appearance as well as in shape; (c) direct blend of textures to a geometric proxy $M(t,w(t)), w(t) = 0.5$; (d) optical flow between the two textures projected onto the geometric proxy; (e) result following proposed online alignment $I(t,w(t),v(t))$.

have similar appearance. Differences in the rendered appearances are due to the different dynamics of the input motions as well as mesh reconstruction and alignment residual errors. Our approach to 4DVT alignment and rendering based on the mesh sequence $M(t,w(t))$ comprises the following steps:

Input: multiple view video capture of an actor performing multiple related motions $V_i(t)$, together with the parametrized mesh sequence $M(t,w(t))$ for user specified parameter $w = w(t)$ and rendering viewpoint $v = v(t)$.

1. For each input sequence $V_i(t)$ we perform view-dependent rendering of view v using the parametrization mesh $M(t,w)$ giving a set of image sequences $I_i(t,w,v)$. This leads to unaligned set of textures, Fig. 7 (c).
2. The alignment between each pair of input video sequences is then evaluated using view-dependant optic flow: $a_{ij}(t,w,v) = g(I_i(t,w,v), I_j(t,w,v))$. Fig. 7 (d).
3. The combined rendered view is produced by blending of the warped texture view: $I_{out}(t,w,v) = b(a_{ij}(t,w,v), I_i(t,w,v), I_j(t,w,v), w)$. Fig. 7(e).

Output: Parametric video texture $I_{out}(t,w(t),v(t))$

Interactive control requires the image alignment $g(.)$ to be performed at run-time. We have found the GPU implementation of Farneback [Far03] (OpenCV 2.4) achieves the best results in terms of alignment error with favourable computational performance. Function $b()$, which runs on the GPU, is a linear interpolation in the 2D screen space of the warped images $I'(a,w)$ according to motion parameters w:

$$b(a, I_i, I_j, w) = w \cdot I'_i(a,w) + (1-w) \cdot I'_j(a,(1-w)), \quad (7)$$

where $I'(a,w)$ is the warped image $I()$ according to the flow field a such that for each pixel p given by $I'(a,w,p) = I(p + w \cdot a(p))$. The linear interpolation combines the forward warp from image $I_i()$ with the backward warp from image $I_j()$ to form the output rendered image $I_{out}(t,w,v)$. Fig. 8 presents 4DVT alignment for two characters at intermediate parameters at intervals $\Delta w = 0.2$ between the two source frames. This illustrates synthesis of detail such as wrinkles for intermediate meshes without blurring and ghosting.

6. Results and Evaluation

Evaluation is performed using 4D video datasets[†] for five people wearing a variety of clothing and performing a range of motions. Datasets JP, Roxanne and Dan are captured using an 8 camera studio [SH07]. Datasets Infantry and Knight were captured using a 10 camera studio with a larger $5m^2$ capture space. 25 sequences between 50-300 frames

[†] Data available in http://cvssp.org/projects/4d/4DVT/

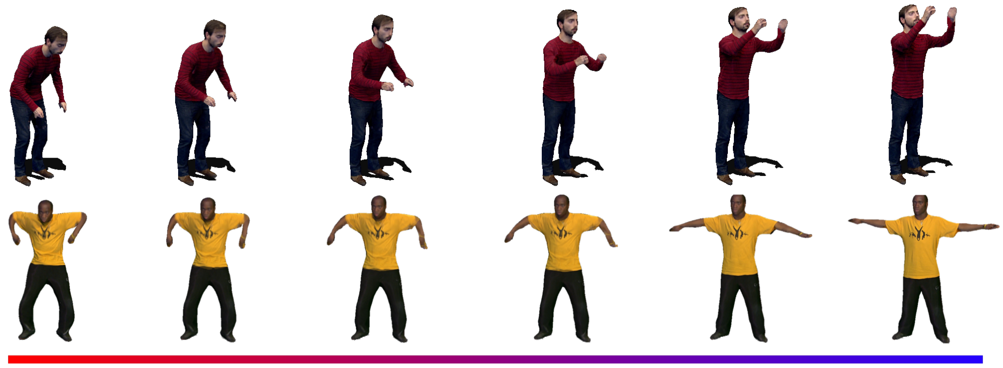


Figure 8: 4DVT rendering results from Sec. 5. Left and right columns are the input free-view point renders of the poses to be interpolated. Columns 2, 3, 4 and 5 are the 4DVT synthesized results $I_{out}(t,w,v)$ for different weight values of vector w. Note how the synthesized novel poses maintain the visual quality of the input data $V_i(t)$. Datasets Dan (top),JP (bottom).

each were used. Interactive animations ran at $\sim$ 18fps on a GeForce GTX 680 GPU, Pentium 2.5Ghz Quad Core.

6.1. Interactive Character Animation

The approach introduced in Sec. 5 enables video-realistic rendering of novel motions from a set of 4D videos of related motions. This allows the construction of a motion class $R(w_R)$ parametrized by a set of high-level motion parameters w_R to control the motion. Character animation within a 3D scene requires the combination of multiple motion classes R_j in a *Motion Graph* [KGP02], which represents the potential transitions between motion sequences, and have previously been extended to represent transitions between parametrized motion classes for skeletal [HG07] and surface motion sequences [CTGH13]. Here we use a parametric motion graph to represent potential transitions between 4D Video motion classes. A cost function that optimizes transition latency and both shape and appearance similarity is used. The use of 4DVT with motion graphs allows synthesis of video-realistic animation that combines a range of parametric motions.

All five datasets are used to construct interactive characters rendered using the 4DVT approach, with $L = 3$. Figs. 1 and 13 show results of 4DVT character animation with interactive movement control (please refer to supplementary video). This demonstrates that view-dependent rendering of dynamic appearance using 4DVT maintains the visual-quality while the character is interactively controlled. Seamless transitions across 4DVT sequences are achieved with continuous control of movement and plausible change in the appearance dynamics of hair and clothing.

Object interaction is illustrated with character Dan, Fig.13(a,d), for a variety of user-controlled motions including picking up objects, and jumping. A variety of environments including a composite over a real video with a moving camera, Fig. 1, are shown. Both Knight and Infantry have relatively loose garments resulting in increased surface dynamics. Results demonstrate that 4DVT visual quality is comparable with the captured video.

6.2. 4DVT Representation

We evaluated the 4DVT representation for view-dependent rendering. Fig. 9 shows the effect of the number of layers on the final view-dependent rendering compared with conventional FVVR, using structural similarity index measure SSIM [WBSS04] as a rendering quality metric, which has shown good correlation with perceived quality. The graph shows that rendering quality increases between 1 to 3 layers before it plateaus. This occurs because information in the layers becomes sparse for $L > 3$ due to the wide spacing between cameras. Note in Fig. 9 that with a single texture layer there is a significant loss of quality due to discontinuities in sampling from different cameras, resulting is visible seams, and loss of view-dependant appearance. Using 2-3 layers the textures are blended across seams eliminating visible discontinuities and achieving high-quality view-dependant results. Further quantitative evaluation is presented in Table 1, including loading time, storage requirements relative to foreground only in conventional FVVR and final rendering quality measured power signal-to-noise ratio PSNR. 4DVT with temporal optimization and 512x512 resolution MPEG video compression achieves $> 90\%$ reduction in size relative to the foreground only multi-view video in the FVVR column, Fig. 10. The latter is verified through a user study in Sec. 6.3. Loading time is also significantly reduced (98%) when storing the 4D video as 4DVT video layers. Spatio-temporal optimization of the layer further reduces the storage costs by allowing the compression algorithms to take advantage of increased coherence within the layer sequences.

6.3. User Aesthetic Preference

51 non-expert participants undertook a web-based survey to evaluate aesthetic preference for videos and stills generated by the 4DVT framework. Participants were asked to compare 30 randomized separate rendering pairs, each pair produced under two different algorithm configurations, and rate them from 1 (preference for the left) to 5 (preference for the right). 3 for no preference. Identical pairs were also inserted as a control, and we observed preferences of 3.15±0.67 indicating natural variations in response.

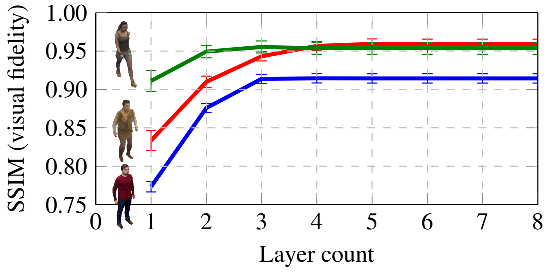


Figure 9: Evaluation of number of layers vs. 4DVT quality.

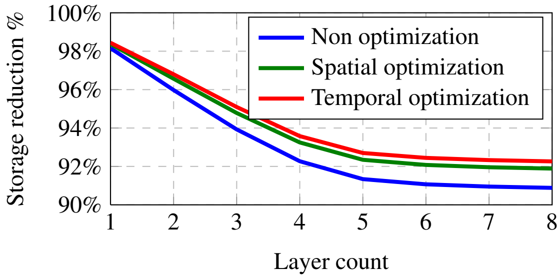


Figure 10: Evaluation of number of layers vs. storage reduction. Computed using dataset JP.

	FVVR	4DVT		
		NO	SO	TO
Dan (29 frames)				
Loading (s)	51.9	0.9 (98%)	0.9 (98%)	0.9 (98%)
Storage(MB)	61.4	4.2 (93%)	3.9 (93%)	3.7 (94%)
SSIM[-1,1]	-	0.81	0.82	0.80
PSNR (dB)	-	44.1	44.1	44.1
JP (250 frames)				
Loading (s)	405.5	6.8 (98%)	6.6 (98%)	4.0 (99%)
Storage(MB)	387.6	23.5 (93%)	20.2 (94%)	19,0 (95%)
SSIM[-1,1]	-	0.80	0.81	0.81
PSNR (dB)	-	42.4	42.5	42.6
Infantry (76 frames)				
Loading (s)	130.5	2.2 (98%)	2.1 (98%)	2.1 (98%)
Storage(MB)	137.7	11.3 (91%)	9.9 (93%)	9.2 (93%)
SSIM[-1,1]	-	0.87	0.88	0.88
PSNR (dB)	-	45.2	45.3	45.2

Table 1: Quantitative evaluation of storage and compression for FVVR vs. 4DVT ($L = 3$, 512x512 resolution) for three different characters with no optimization (NO), spatial optimization (SO) and temporal optimization (TO).

Perceptual impact of any deterioration in rendering quality induced by 4DVT layered representation is evaluated in Fig. 11. Overall slight preference for FVVR (3.41±0.94) was observed in still imagery. This may be explained by a slight drop in visual fidelity caused by 4DVT in still imagery. However this difference becomes less perceptible in video (3.25±0.75), see supplementary video. We next evaluated 15 pairs of real 4D captured character (rendered using FVVR) alongside a synthesized 4DVT character, Fig 12. Overall the output is judged to be approximately equal (3.30±1.10) indicating the plausibility of the synthetic character. Interestingly, datasets captured in studio B, which have lower resolution than studio A, present greater preference for real imagery (3.61±1.03). This is caused by the increased noise in the optical flow due to low-res textures, resulting in poorer 4DVT alignment. Nevertheless, the overall mean is approximately neutral (3.30) with a high standard deviation (1.10), indicating little perceivable difference between synthetic output and real imagery. Finally, a synthetic 4DVT animation was shown to the participants, which were asked to rate it from 1 (very artificial) to 5 (very realistic). The overall score of 3.97±0.69, indicates good level of acceptance.

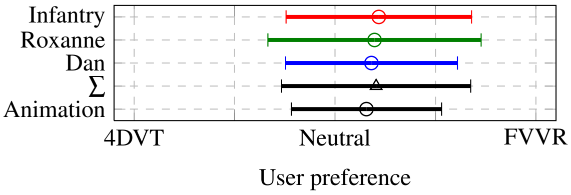


Figure 11: Preference for 4DVT compression vs. FVVR.

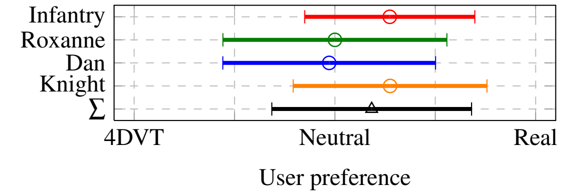


Figure 12: Preference for 4DVT synthesis vs. Real Capture.

6.4. Discussion

Results and evaluation presented show that the proposed 4DVT representation and rendering achieves visual quality of dynamic appearance similar to the captured video. Quality of the 4DVT renderings is dependent on a number of factors: high-resolution multiple view video capture without motion blur; 4D video reconstruction and robust temporal alignment; and accurate optical flow alignment in 4DVT rendering. Visual artefacts in either the captured video or 4D reconstruction will degrade the quality of the final rendering. Small residual errors from incorrect geometric reconstruction and alignment can be observed in the Knight and Infantry due to the relatively complex clothing deformations and lower resolution capture. Errors in the online optical flow alignment will result in blur or ghosting artefacts.

The main limitation of our approach is the computational overhead of the online optical flow. Although this is a performed in the GPU in real-time this limits the rendering to a single character. Pre-computation of the alignment is desirable, however, this is non-trivial due to the difference in character pose and camera views in each motion. Alignment computation in the 4DVT space is also non-trivial due to the discontinuities in texture. A possible solution is prealignment in a set of canonical rendered views, however this may result in a loss of quality.

Finally, our pipeline does not include any color correction or relighting step to adjust the illumination properties. Recent research on relightable performance [LWS*] could be used to increase the realism of the final video composite.

7. Conclusions

We have presented 4D Video Textures (4DVT), a framework for 4D video representation and rendering with two main contributions: an efficient representation of captured multiple video sequences as 4D video texture maps; and real-time rendering of video-realistic novel motions with view-point control that maintain the detailed dynamics present in the source videos. A layered view-dependent texture map representation is introduced for efficient storage and rendering of 4D video, achieving two orders of magnitude reduction in size and rendering load time without loss of visual quality. Multiple 4DVTs are combined to render interactive characters with video-realistic dynamic appearance which changes with the motion. Our approach renders multiple-view video using a proxy geometry with the desired character motion. Optic flow alignment is performed between the rendered images from different 4D videos to obtain a texture which preserves the appearance detail from the requested view-point. The resulting 4DVT maintains the appearance quality of the source video, including plausible dynamics correlated with the character motion. A user-study of the rendering quality shows that there is no significant difference between the synthesized 4DVT motions and original 4D video sequences.

Acknowledgments. Work partially supported by RE@CT EU FP7 Project. We thank Margara Tejera for helping during the capture of dataset *Dan*, Peng Huang for extracting background camera features for Fig 1, and RE@CT partners for their help in acquiring datasets Infantry and Knight.

References

[AHE13] A. Hilsmann P. F., Eisert P.: Pose space image based rendering. *Computer Graphics Forum 32*, 2 (2013). 2

[AKT08] Alahari K., Kohli P., Torr P. H. S.: Reduce, reuse, recycle: Efficiently solving multi-label MRFs. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2008). 4

[BBM*01] Buehler C., Bosse M., McMillan L., Gortler S., Cohen M.: Unstructured lumigraph rendering. In *Proc. of SIGGRAPH* (2001), pp. 425–432. 3

[BCS97] Bregler C., Covell M., Slaney M.: Video rewrite: Driving visual speech with audio. In *Proc. ACM SIGGRAPH* (1997), pp. 1—8. 2

[BHKH13] Budd C., Huang P., Klaudiny M., Hilton A.: Global non-rigid alignment of surface sequences. *International Journal of Computer Vision 102*, 1-3 (2013), 256–270. 2, 3

[BHV*11] Brostow G., Hernandez C., Vogiatzis G., Stenger B., Cipolla R.: Video Normals from Colored Lights. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 10 (2011), 2104—2114. 2

[CBI10] Cagniart C., Boyer E., Ilic S.: Free-Form Mesh Tracking: a Patch-Based Approach. In *CVPR* (2010), pp. 1339–1346. 2

[CTGH13] Casas D., Tejera M., Guillemaut J.-Y., Hilton A.: Interactive Animation of 4D Performance Capture. *IEEE Transactions on Visualization and Computer Graphics 19*, 5 (2013), 762–773. 3, 5, 7

[dST*] de Aguiar E., Stoll C., Theobalt C., Ahmed N., Seidel H.-P., Thrun S.: Performance Capture from Sparse Multi-view Video. *Proc. ACM SIGGRAPH 2008 27*, 3. 1, 2, 3

[EDDM*] Eisemann M., De Decker B., Magnor M., Bekaert P., de Aguiar E., Ahmed N., Theobalt C., Sellent A.: Floating textures. *Eurographics 2008 27*, 2. 2

[Far03] Farnebäck G.: Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis* (2003), pp. 363–370. 6

[FNZ*09] Flagg M., Nakazawa A., Zhang Q., Kang S.-B., Ryu Y., Essa I., Rehg J.: Human Video Textures. In *ACM Symposium on Interactive 3D Graphics* (2009). 2

[GWO*10] Gal R., Wexler Y., Ofek E., Hoppe H., Cohen-Or D.: Seamless montage for texturing models. *Computer Graphics Forum (Proc. Eurographics) 29*, 2 (2010). 2

[HG07] Heck R., Gleicher M.: Parametric Motion Graphs. In *ACM Symposium on Interactive 3D Graphics* (2007). 7

[HHS09] Huang P., Hilton A., Starck J.: Human Motion Synthesis from 3D Video. In *CVPR* (2009). 2

[JTST10] Jain A., Thormählen T., Seidel H.-P., Theobalt C.: Moviereshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. 29*, 5 (2010). 2

[KGP02] Kovar L., Gleicher M., Pighin F.: Motion graphs. In *Proc. ACM SIGGRAPH* (2002), pp. 473–482. 7

[KR97] Kanade T., Rander P.: Virtualized Reality: Constructing Virtual Worlds from Real Scenes. *IEEE MultiMedia 4*, 2 (1997), 34—47. 2

[LWS*] Li G., Wu C., Stoll C., Liu Y., Varanasi K., Dai Q., Theobalt C.: Capturing Relightable Human Performances under General Uncontrolled Illumination. *Computer Graphics Forum (Proc. of Eurographics) 22*, 2, 275–284. 2, 9

[SH07] Starck J., Hilton A.: Surface capture for performance-based animation. *IEEE Computer Graphics and Application 27* (2007), 21–31. 6

[SMH05] Starck J., Miller G., Hilton A.: Video-based character animation. In *ACM Symp. on Comp. Anim.* (2005). 2

[SSE00] Schodl A., Szeliski R. amd Salesin D., Essa I.: Video textures. In *Proc. ACM SIGGRAPH* (2000). 2

[TAL*07] Theobalt C., Ahmed N., Lensch H., Magnor M., Seidel H.-P.: Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 663–674. 2

[VBMP08] Vlasic D., Baran I., Matusik W., Popović J.: Articulated mesh animation from multi-view silhouettes. In *Proc. ACM SIGGRAPH* (2008). 1, 2, 3

[WBSS04] Wang Z., Bovik A., Sheikh H., Simoncelli E.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Proc. 13*, 4 (2004), 600–612. 7

[WVL*11] Wu C., Varanasi K., Liu Y., Siedel H.-P., Theobalt C.: Shading-based dynamic shape refinement from multi-view video under general illumination. In *Proceedings of International Conference on Computer Vision* (2011). 2

[XLS*] Xu F., Liu Y., Stoll C., Tompkin J., Bharaj G., Dai Q., Seidel H.-P., Kautz J., Theobalt C.: Video-based Characters - Creating New Human Performances from a Multi-view Video Database. *Proc. ACM SIGGRAPH 2011*. 1, 2

[ZKU*04] Zitnick C., Kang S., Uyttendaele M., Winder S., Szeliski R.: High-quality video view interpolation using a layered representation. In *Proc. ACM SIGGRAPH* (2004). 1, 2, 3

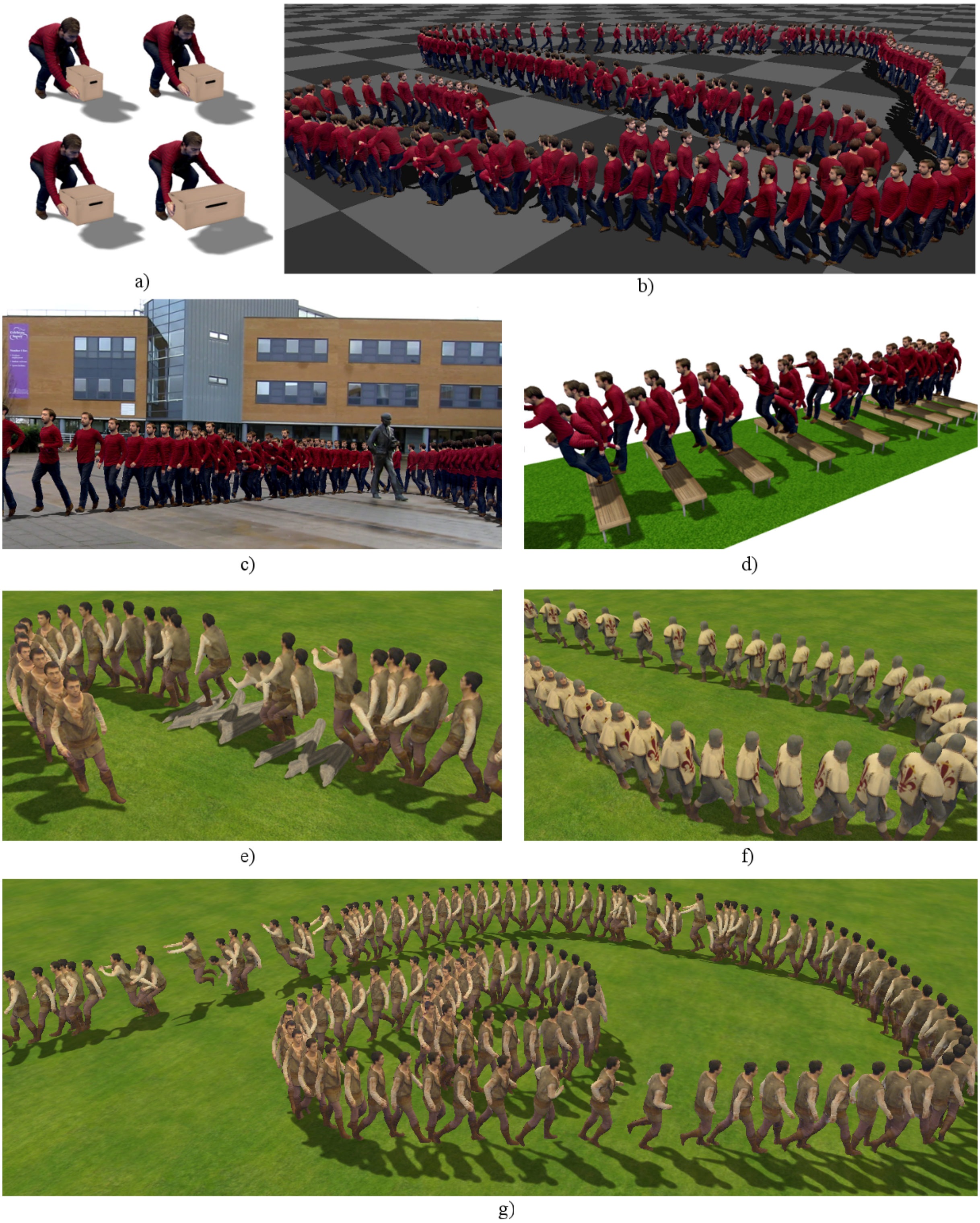


Figure 13: 4D video character animation using 4DVT. (a) Character Dan picking up boxes of different sizes. (b,c,d) Character Dan animation: virtual scenario; hand-held recorded camera footage composition; jumping benches with variable spacing. (e) Infantry character jumping over obstacles. (f,g) Character animation of Knight and Infantry, respectively.