4D Video Textures for Interactive Character Appearance

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. Conse-
Finally there remains a gap between existing video-based characters in interactive application 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 multiple 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 and costumes 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 modifications to 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 cues from shading, self-shadowing 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 match 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.

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 not only based on 2D appearance but also on 3D pose information. These methods achieve 2D video-realistic animation, but are limited to replaying captured segments. Recently, photo-realistic animation of clothing has been demonstrated from a set of 2D images augmented with 3D shape information to achieve natural appearance transitions [AHE13].

Character animation by resampling of 4D performance 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 video 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 vertices correspond to the same surface point over time, and \( V(t) = \{I_i(t)\}^{I_i}_{i=1} \) 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*04]. This ‘free-viewpoint
video rendering’ (FVVR) process preserves the visual real-
ism 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 capture surface [CTGH13], this en-
ables 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 facilitate foreground segmen-
tation.

2. **4D video reconstruction:** Temporally consistent mesh
   sequences are reconstructed [dST*, VBMP08, BHKH13]
   to create a set of 4D videos $F(t) = \{M(t), V(t)\}$.

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 $\geq90\%$ compression and negligible loss of FVVR
   quality (Sec. 4). Thus the set of 4D videos is represented
   efficiently as $F(t) = \{M(t), L_i(t)\}$.

5. **4DVT rendering:** Dynamic appearance of multiple cap-
tured motions $\{L(t)\}_{t=1}^T$ are aligned at render time ac-
cording to the user-controlled viewpoint $v(t)$, motion
   $w(t)$ and mesh sequence proxy $M(t, w(t))$ to render a novel
   video sequence $I_{nov}(v(t), v(t), w(t))(t)$ (Sec. 5).

6. **Compositing:** Rendered video sequences are compos-
ited 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 al-
low 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 ren-
dered 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 views 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 calibra-
tion [ZKU04]. 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 $800\text{MB/s}$. Foreground masking reduces the band-
width by $50$-$80\%$ but the data size limits FVVR to local ren-
dering 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 vis-
ibility. Resampling into the layered texture domain is opti-
imized to preserve spatial and temporal coherence. This
resulting 4DVT representation removes the redundancy in
the captured video achieving $\geq90\%$ compression over the
foreground only masking and reduces rendering to a simple
lookup without a perceivable loss of photorealistic render-
ing quality (Sec. 6.2). The 4DVT representation allows free-
viewpoint rendering on relatively low-performance mobile
devices and streaming for web-based applications.

#### 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 in-
formation 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 cam-
ers $C$. In practice we show that for a typical multi-camera
setup $L \ll C$ 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 ac-
cording 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
genometry, computing a set of depth maps with respect to $C$
camera viewpoints, $\hat{C}$. Considering each viewpoint in turn,
for each facet of the mesh \( f \in M_i(t) \) we compute the angle between the normal to \( f \) (written \( \hat{n}(f) \)), and the optical axis \( \hat{o}(c) \) of the camera \( c \in C \). 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 \( c \in C \). We write this camera label assignment for each face as \( Z = \{z_f, \ldots, z_f(M_i(t))\} \) with \( z_f \in C \). The suitability of a given mapping is found by minimizing the energy function

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

where \( N_f \) indicates the 1-neighborhood of node \( f \in G_{ci} \). The unary term \( \Phi_f \) 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 \( \Phi_{f,j}(z_f, z_j) \) penalizes dis-similarity between \( f \) and its immediate neighbor \( j \) in \( M_i(t) \):

\[
\Phi_{f,j}(z_f, z_j) = \begin{cases} 0 & \text{if } z_f = z_j, \\ 1 & \text{otherwise.} \end{cases} \quad (4)
\]

As this pairwise potential is in the form of a Potts model it can be minimized using the \( \alpha \)-expansion and \( \alpha \)-swap algorithm [AKT08]. Fig. 4(b) shows a representative 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 \( E'(Z) \).

\[
E'(Z) = \sum_t E(Z) + \frac{1}{|M_i|} \sum_{f \in M_i} \frac{1}{|M_i|} \sum_{f \in M_i} \Phi_{f,j}(z_{f(t)}, z_{f(t-1)}). \quad (5)
\]

Solution via \( \alpha \)-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.

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.
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 halfing 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 \( v(t) \).

\[
I(t,w(t),v(t)) = f(F'_1(t), ..., F'_N(t), w(t), v(t)) \quad (6)
\]

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 \( L(t) \) 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 \( \{F'_i\} \) 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. We therefore require an approach to accurately align and interpolate the dynamic appearance of the input motions. Our online view-dependent alignment and rendering approach, shown in 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 pre-compute 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 synthesise 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 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-dependent 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. Texture blending and warping is also performed on the GPU. Fig. 8 presents 4DVT alignment for two characters rendered with intermediate parameters at intervals $\Delta w = 0.2$ between the two source frames. This illustrates reproduction 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 public datasets captured using a 8 camera studio [SH07]. Datasets Infantry and Knight were captured using a 10 camera studio with a larger $5m^2$ capture space. A total of 25 sequences
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 $I_{\text{out}}(t, w, v)$ 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).

between 50-300 frames each were used. Interactive animations ran at $\sim 18$fps on a GeForce GTX 680 GPU equipped Pentium 2.5Ghz quad core PC.

### 6.1. Interactive Character Animation

The approach introduced in Sec. 5 enables the 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 within class motion. Video-based character animation within a 3D scene requires the combination of multiple motion classes $R_j$ in a Motion Graph [KGP02]. The motion graph represents the potential transitions between motions sequences. Motion graphs have previously been extended to represent transitions between parametrized motion classes for skeletal [HG07] and surface [CTGH13] motion sequences. 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 to synthesize video-realistic animation that combines a range of parametric motions.

All five datasets are used to construct interactive characters rendered using the view-dependent 4DVT approach, with $L = 3$. Figs. 1 and 13 show results of 4DVT character animation with interactive movement control (please refer to supplementary video for full results). This demonstrates that view-dependent rendering of dynamic appearance using the proposed 4DVT approach maintains the visual-quality while the character movement is interactively controlled. Seamless transitions between different 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. The composite presents comparable visual fidelity to the captured video. 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 quality of the final render 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. Further quantitative evaluation is presented in Table 1, including also loading time, storage requirements relative to foreground only in conventional FVVR and final rendering quality measured power signal-to-noise ratio PSNR. This analysis shows that 4DVT with temporal optimization and 512x512 resolution MPEG video compression achieves $> 93\%$ reduction in size without significant reduction in visual quality, Fig. 10. The latter is verified through a user study in Sec. 6.3. The loading time is also significantly reduced (98%) when storing the 4D video as 4DVT video layers. Spatio-temporal optimization of the layer sampling further reduces the total 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 submitted to EUROGRAPHICS 2014.
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-res 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 character video 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.
In addition to improvements in the 4D video reconstruction pipeline, the main limitation of our approach is the computational overhead of the online optical flow. Although this is performed in real-time this limits the rendering to a single character and requires high-performance GPU. 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 pre-alignment in a set of canonical rendered views, however this may result in a loss of quality.

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 the 4D video. This representation is demonstrated to achieve 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 composite image 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 indicates that there is no significant difference between the synthesized 4DVT motions and original 4D video sequences.

References


Figure 13: (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.