Optimal Representation of Multi-View Video

Marco Volino
m.volino@surrey.ac.uk
Dan Casas
d.casasguix@surrey.ac.uk
John Collomosse
j.collomosse@surrey.ac.uk
Adrian Hilton
a.hilton@surrey.ac.uk

Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, UK

Abstract

Multi-view video acquisition is widely used for reconstruction and free-viewpoint rendering of dynamic scenes by directly resampling from the captured images. This paper addresses the problem of optimally resampling and representing multi-view video to obtain a compact representation without loss of the view-dependent dynamic surface appearance. Spatio-temporal optimisation of the multi-view resampling is introduced to extract a coherent multi-layer texture map video. This resampling is combined with a surface-based optical flow alignment between views to correct for errors in geometric reconstruction and camera calibration which result in blurring and ghosting artefacts. The multi-view alignment and optimised resampling results in a compact representation with minimal loss of information allowing high-quality free-viewpoint rendering. Evaluation is performed on multi-view datasets for dynamic sequences of cloth, faces and people. The representation achieves >90% compression without significant loss of visual quality.

1 Introduction

Image-based modelling from multi-view video acquisition enables photo-realistic rendering of dynamic real-world scenes and actor performances from arbitrary viewpoints, referred to as free-viewpoint rendering (FVR) [24]. Research has focused on multi-view reconstruction to obtain a 3D proxy of the dynamic scene enabling FVR by resampling from the captured images. Recent research has also addressed the problem of estimating temporally coherent geometry by non-rigid alignment of the reconstructed surface sequence. However, only limited attention has been paid to the representation of multi-view appearance.

Current approaches to FVR resample directly from the captured multi-view images at each time frame, achieving a high level of photo-realism but requiring storage and transmission of multi-video sequences. This is prohibitively expensive in both storage and bandwidth required for multiple video streams limiting applications to local rendering on high-performance hardware. To reduce the storage cost a single, static texture map per frame is commonly extracted by resampling the captured multi-view images to a 2D texture domain. This achieves a compact representation but results in loss of any view-dependent appearance detail, such as specularities, and introduces visual artefacts of ghosting and blurring if

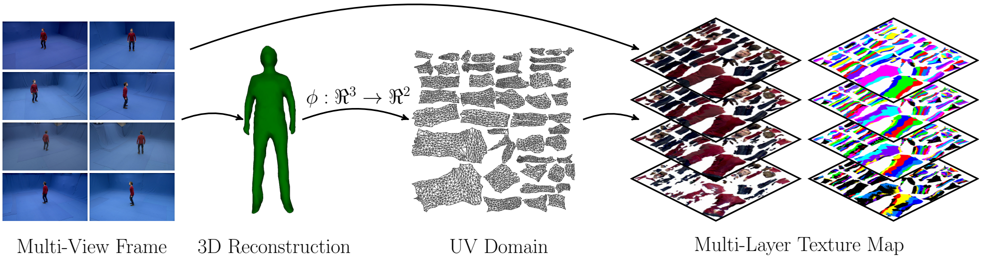


Figure 1: Overview of the resampling of multi-view video to a multi-layer texture video

there are errors in surface reconstruction or camera calibration. In this paper we address the problem of optimal representation of appearance for dynamic objects, such as people, from multi-view video acquisition. Figure 1 presents an overview of our approach mapping the captured multi-view video to a multi-layer texture map. Our primary contributions are:

- Multi-layer texture map representation of view-dependent appearance to maintain FVR quality in the presence of errors in geometry and calibration.
- Alignment of multi-view appearance to refine spatial coherence for resampling.
- Optimal sampling from multi-view video to maximise spatio-temporal coherence.
- Quantitative evaluation of rendering and storage for multi-layer texture map representation verses FVR from the captured images.

Optimal resampling and multi-layer texture map representation of multi-view video is evaluated on reconstructions from dynamic sequences of cloth, faces and people wearing a variety of clothing. Results demonstrate that the approach achieves a comparable visual quality to direct FVR from the captured multi-view video with >90% reduction in storage/transmission costs and improvements in rendering efficiency.

2 Related Work

Image and video-based modelling uses multi-view reconstruction to capture detailed object geometry and render novel views by resampling from the captured images. Extraction of a single surface texture map per frame combining the observed appearance from multi-view images has been widely used to provide a compact representation.

The simplest approach is to blend overlapping image regions weighted according to surface visibility for each camera [18]. This approach assumes accurate geometric reconstruction and camera calibration. In practice these errors commonly result in degradation of the visual quality producing blurring and ghosting artefacts due to misalignment between camera views projected onto the reconstructed surface. Correction of misalignment between multi-view images, due to errors in geometry and camera calibration, is addressed for image-based rendering in *Floating Textures* [10]. The approach performs online optical flow alignment in the rendered image for a specific viewpoint to reduce visual artefacts. Recent work has extended this approach to interactive video-based character animation using video-textures with similar online alignment [7]. These approaches achieve a high-level of visual realism but require expensive real-time optic flow computation using a high-performance GPU which is impractical for many applications. In this paper a surface-based optical flow approach is introduced to pre-compute and store the alignment between views. This achieves a high-visual quality and removes the requirement for online optic flow alignment in the rendered view [7, 10].

Offline approaches to align multi-view images of static objects have also been proposed using sparse feature matching to reduce visual artefacts[1]. A number of approaches have been proposed to generate spatially coherent texture maps of models reconstructed from multi-view images [3, 11, 13, 15] which minimise visual artefacts due to view transitions. These approaches cast the problem as a Markov Random Field (MRF) spatial optimisation to find the labelling between mesh polygons and cameras images that minimise an energy function based on camera visibility and transitions in camera views between adjacent faces. The energy function comprises two terms: a unary term defining a quality measure for assigning a camera image to a polygon based on angle between the normal and camera direction [15] and the projected area of the polygon [3, 13], maximising the sampling resolution; and a pairwise term defining the cost of assigning camera images to adjacent polygons, commonly defined as the colour distance at the edge of adjacent polygons [3, 11, 13, 15] to minimise visible seams. This approach still results in some visible seams due to non-Lambertian reflectance and misalignment of appearance between views resulting from errors in the underlying surface geometry and calibration. To reduce visible artefacts, Gal *et al.* [11] propose an extension[15] to allow alignment in the image domain to compensate for inaccuracies in camera calibration and geometry and reduce the visibility of seams.

Goldlucke and Cremers [12] introduced a variational formulation for extract super-resolution texture maps from multi-view images which achieves high-quality alignment and rendering even from low-quality images. Recent research [21] extends texture super-resolution to consider both spatial integration across multiple views and temporal integration over a short time window.

Whilst these methods give impressive results they are limited to static scenes. Janko *et al.* [13] extend MRF-based texture map extraction to dynamic scenes proposing two approaches: a spatially optimised texture map per frame; and a single texture map across all frames. Independent spatial optimisation of multi-view sampling does not ensure temporal coherence resulting in flicker artefacts due to camera labels switching over time. Extraction of a single texture map for the entire sequence results in loss of dynamic appearance details which are not modelled geometrically such as cloth or skin wrinkling. Existing approaches to resampling from multi-view image capture result in a significant loss of visual quality. Consequently, state-of-the-art approaches for video-based rendering of dynamic scenes resample directly from the captured images to maintain visual quality.

In this paper we address the problem of optimisation of both spatial and temporal resampling from multi-view video. Spatio-temporal optimisation of resampling is combined with a dense surface based optical flow approach to correct for misalignment due to geometric and calibration errors. This enables sampling of multi-view video into texture space to produce a compact representation of the dynamic surface appearance which maintains visual quality for free-viewpoint video rendering of captured scenes.

3 Problem Statement: Optimal Multi-View Resampling

Multi-view video of a dynamic scene, such as an actor's performance, is highly redundant due to both the spatial overlap of camera views on the surface and the inherent temporal coherence of surface appearance. However, both spatial and temporal changes in appearance are important to maintain the realism of the captured video. Spatial changes in appearance with viewpoint occur due to non-Lambertian reflectance together with any errors in reconstructed surface shape and camera calibration. Temporal changes in appearance reproduce the detailed surface dynamics, such as cloth wrinkling, hair and facial expression, which are often not represented accurately in the reconstructed surface shape. Preserving the observed spatio-temporal changes in surface appearance is required to maintain the realism in image-based FVR. The problem of representation of dynamic surface appearance from multi-view video addressed in this paper is defined as:

Optimal resampling of the captured views to obtain a compact representation without loss of view-dependent dynamic surface appearance information.

Resampling and representation should satisfy the following competing requirements:

1. Minimal loss of information due to image resampling;
2. Efficient representation of information across multiple views to minimise data size for storage, transmission and rendering;
3. Spatial coherence of view sampling in the texture domain such that adjacent texels are sampled from the same view to minimise switches in viewpoint;
4. Temporally coherent representation such that the texel corresponding to the same surface point is sampled from the same camera over time to minimise switches in camera;
5. Ordered sampling of multiple view observations of the same surface point according to visibility to allow efficient rendering (angle to the camera view and surface-image sampling resolution);
6. Texture layer alignment such that multiple observations of the same surface point at a time-instant have the same texel location.

In order to efficiently represent the appearance information from multiple views whilst minimising information loss, we introduce a layered texture map representation such that the observations are resampled to a hierarchy of 2D texture layers according to surface visibility. Note that with the layered texture representation, if the number of layers is equal to the number of views the observed information content in the multi-view video is preserved up to the resampling from the captured 2D image domain to the 2D texture domain. This representation allows a significant reduction in size by limiting the number of layers and reducing the spatial and temporal redundancy in the captured camera images. If the representation is reduced to a single layer at each frame this results in a conventional 2D texture map per frame with optimal resampling of the multi-view video but loss of any view-dependent appearance. In the following section, we introduce the layered representation, spatio-temporal optimisation for multi-view resampling and alignment to obtain an optimised representation according to the requirements stated above.

4 Optimal Multi-View Resampling and Representation

The input to our approach is a multi-view video sequence comprising a set of N_C camera views, $\{I_j(t)\}_{j=1}^{N_C}$, together with a reconstructed and temporally aligned mesh sequence $M(t) = (T, V(t))$ where T is the constant mesh topology and $V(t)$ are the vertex positions which change over time $t = 1...N_T$. Temporally consistent mesh reconstruction can be performed using a variety of multi-view reconstruction and surface alignment techniques [5, 6, 8, 22]. The reconstructed mesh sequence $M(t)$ provides a geometric proxy for FVR from the captured multi-view video. In this paper, we address optimal resampling of multi-view video to a layered texture map representation.

4.1 Layered Texture Representation

A reconstructed mesh surface manifold $M \in \Re^3$ can be mapped to a 2D domain to obtain a texture map image $U \in \Re^2$. A number of approaches have been introduced to define texture coordinates by either projection of the surface as a set of charts or continuous unwrapping (pelting) of the surface [16, 17]. Surface projection is used throughout this work as this minimises the distortion in the mapping between the 3D and 2D domains. Given video from a set of N_C camera views $\{I_j(t)\}_{j=1}^{N_C}$ together with a mapping $\phi : \Re^3 \rightarrow \Re^2$ from the mesh surface M to the texture domain U, which remains constant in time due to the constant mesh topology, we can generate a set of $N_L \leq N_C$ texture layers $\{U_p(t)\}_{p=1}^{N_L}$ for each time-frame t. A straightforward approach would be to map each camera to a separate texture map $N_L = N_C$, however this would not address the issue of spatial or temporal redundancy in the input multi-view video. Instead, we use a hierarchy of N_L texture maps ordered according to the visibility of each mesh facet such that the first texture layer, U_1, resamples from the camera view from which each facet is most visible. Facet visibility is evaluated based on the angle between the camera view direction and surface normal [9], and any inter-facet occlusion. Other criteria such as camera sampling resolution could also be incorporated but are not required for uniform camera spacing as used in this work.

This results in a layered hierarchy of texture maps U_p with the N_C input camera views resampled to a set of N_L texture maps. Due to the ordering of the texture map layers based on surface visibility, the view-dependent appearance information required to maintain FVR quality can be represented with $N_L \ll N_C$. For convenience, the layered texture representation also stores the pre-computed surface visibility for each facet from each camera in the texture map alpha channel for subsequent rendering by weighted blending of the layers according to surface visibility. Section 5 quantitatively evaluates the performance of the layered representation for FVR rendering demonstrating that for a typical $N_C = 8$ multi-camera setup, $N_L = 3$ results in minimal loss compared to direct rendering from the original images.

The simple independent per facet mapping to the texture domain results in both spatial and temporal fragmentation of the resampling from different camera views producing a sub-optimal representation. This may result in visual artefacts due to adjacent mesh facets sampling from different camera views resulting in discontinuities in appearance due to non-Lambertian reflectance, incorrect geometry and inexact camera calibration [10]. Temporal fragmentation will introduce high-frequency switching of camera views over time, resulting in flicker artefacts. In this paper, we optimise the resampling from the original camera views to maximise spatial and temporal coherence.

4.2 Optimal Multiple View Video Resampling

Optimal resampling from multiple views requires spatial and temporal coherence of the representation as set out in Section 3. The optimisation of spatio-temporal coherence is formulated as a labelling problem of mesh facets to cameras. Formally, the problem can be cast as a labelling problem where we seek the mapping $L : F \rightarrow C$ from the set of mesh facets F to the set of cameras $C = \{1...N_C\}$ which assigns a camera label $l_f \in C$ to each facet $f \in F$. We formulate the computation of the optimal labelling $L(t)$ as an energy minimisation of cost:

$$E(L(t)) = \sum_{\forall t} (E_v(L(t)) + \lambda_s E_s(L(t)) + \lambda_t E_t(L(t), L(t+1)). \tag{1}$$

where $E_v(L(t))$ is the unary visibility cost for all faces F to be assigned camera labels $L(t)$ at time t, $E_s()$ is the spatial coherence cost which enforces consistent camera labelling between adjacent mesh facets, and $E_t()$ is the temporal coherence cost which enforces temporal coherence of the camera labelling. In practice, the spatial and temporal coherence weight terms are set to unity $\lambda_s = \lambda_t = 1$ as the costs are balanced. The unary visibility cost is given by:

$$E_v(L(t)) = \sum_{f \in F} e_v(l_f(t)) \tag{2}$$

where $e_v(l_f(t))$ is the visibility cost associated with facet f being assigned camera label l_f at time t. This cost is given by the angle between the facet normal $n_f(t)$ and the camera view direction $v(l_f(t))$ if the facet is visible:

$$e_v(l_f(t)) = 1 - (n_f(t) \cdot v(l_f(t)))^2 \tag{3}$$

or infinity if the facet is not visible. This visibility penalty is widely used in FVR for the weighted combination of views [9]. Penalty terms which take into account the image sampling resolution for each facet could also be used for non-uniform camera setups. Spatial coherence of camera sampling is enforced by penalising different camera assignments for adjacent faces:

$$E_s(L(t)) = \sum_{f \in F} \left(\frac{1}{|N_f|} \sum_{r \in N_f} e_s(l_f(t), l_r(t)) \right) \tag{4}$$

where N_f is the 1-neighbourhood of facet f, and $e_s(l_f(t), l_r(t)) = 1 - v(l_f(t)) \cdot v(l_r(t))$ if both facets are visible for assigned camera labels, otherwise infinity.

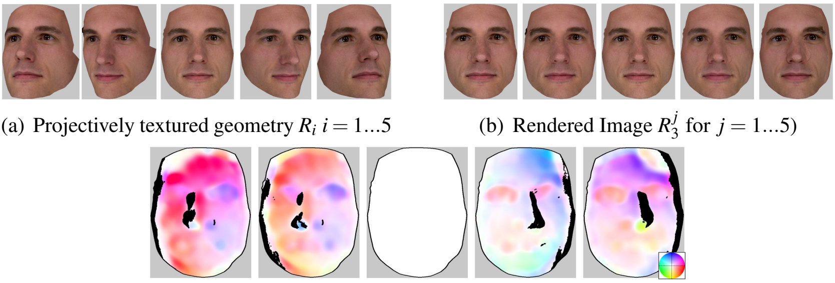
Similarly, temporal coherence is enforced by penalising different camera assignments for the same facet at consecutive frames:

$$E_t(L(t), L(t+1)) = \sum_{f \in F} e_t(l_f(t), l_f(t+1)) \tag{5}$$

where $e_t(l_f(t), l_f(t+1)) = 0 : l_f(t) = l_f(t+1) \vee 1 : l_f(t) \neq l_f(t+1)$. Optimisation of equation 1 for multi-view sequences is performed efficiently using graph cut $\alpha\beta$-swap [2, 4]. Label assignment is represented as a graph with mesh faces as nodes and edges representing facet adjacency. Optimisation of a mesh with 5K faces for 8 camera views takes approximately 1-2 seconds per frame for typical free-viewpoint video sequences of 15-20 seconds.

4.3 Multiple View Texture Alignment

Simple projection and blending of camera views using the approximate reconstructed mesh geometry leads to blurring and ghosting artefacts. These artefacts are caused by misalignment between overlapping camera images projected onto the mesh surface from inaccurate geometry and camera calibration. In order to minimise these artefacts, we use optical flow based image warping to correct misalignments before sampling into the texture domain. Previous approaches [7, 10] have used online optic flow computation in the rendered view. In this work we introduce a pre-computation of the alignment using a surface based optic flow.



(a) Projectively textured geometry R_i $i = 1...5$

(b) Rendered Image R_3^j for $j = 1...5$)

(c) Optical flow $O_{i \to j}$ between rendered image R_3^3 and R_3^j for $j = 1..5$. Optical flow colour mapped to direction

Figure 2: Optical flow correspondence between camera views

To establish optical flow between camera views, we first render the geometry from the viewpoint of camera C_i and projectively texture using the image of camera C_j for all N_C cameras. This results in N_C^2 rendered images, R_i^j, which denotes the image rendered from the i^{th} camera viewpoint using the j^{th} camera image. An optical flow correspondence field, $O_{i \to j}$, is computed between the rendered image $R_i = R_i^i$ and R_i^j where $i \neq j$. Optical flow is known to give unreliable flow vectors in the presence of occlusions where it is undefined. To mitigate such errors, a binary confidence score is assigned to each flow vector based on depth discontinuities (taken from a rendered depth map of the geometry) and occlusions (computed by identifying vertices visible in C_i but occluded in C_j). Fig. 2 shows such artefacts in black, indicating zero confidence scores S_i^j. A correction vector is applied to the projected point in the camera domain to take into account the projection error. The magnitude of the correction vector is given by a weighted average of all visible and high-confidence flow vectors:

$$V_i = \sum_{j=1}^{N_C} \omega_j S_i^j O_{i \to j} \tag{6}$$

where multiplication of fields occurs on a per-pixel basis, and V_i is the field of correction vectors for the i^{th} camera, ω_j is a scalar weight such that $\sum_{j=1}^{N_C} \omega_j = 1$. In our experiments we use uniform weighting, however this could be varied to prioritise particular cameras. Optimal resampling of the captured multi-view images as a layered texture map representation (section 4.1) is achieved by combining the optical flow alignment of the captured images on the reconstructed surface with the spatio-temporal optimisation of camera label assignments for each mesh facet (section 4.2). This ensures that: (1) rendering artefacts due to incorrect geometry are minimised and texels in different layers correspond to observations of the same surface point; (2) multi-view images are sampled to optimise the spatial coherence between views and minimise changes in camera view; and (3) the representation is temporally coherent such that switches in camera viewpoint over time are minimised. This leads to an efficient view-dependent representation of the multi-view video with minimal loss of information.

5 Evaluation

Evaluation is performed on four different multi-view video datasets shown in Figure 3. *Character 1* and *Dan* were captured using eight cameras in a circle of 8 metre diameter giving full 360 degree coverage of the subject with reconstruction [19] and temporal alignment [5]. *Face* and *Cloth* were captured using five cameras in a frontal configuration with reconstruction and alignment [14]. Dataset and storage requirements are summarised in Table 1, all video was captured at 1920x1080 HD-SDI at 25P. Datasets available for research: cvssp.org/cvssp3d.

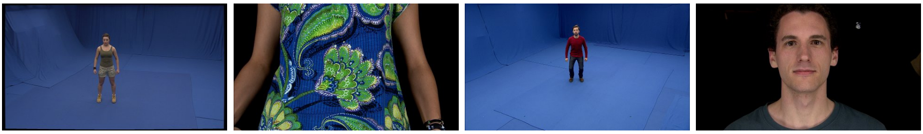
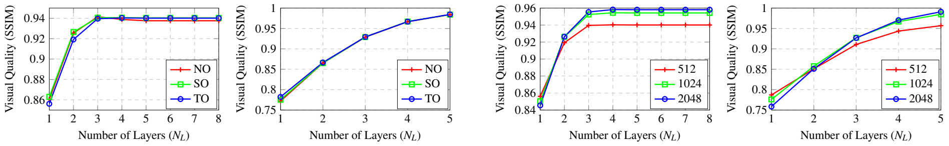


Figure 3: Single camera image from evaluated datasets: *Character 1; Cloth; Dan; Face*.

Texture Alignment: Results of the proposed surface-based optical flow alignment (section 4.3) are presented in Figure 6 for the Dan, Face and Cloth examples. For Dan and Face examples U_1 is show before (left) and after (right) alignment with a heat map (centre) highlighting the difference. In the Cloth example, $N_L = 3$ are blended into a conventional texture map to highlight ghosting and blurring artefacts. Before alignment large misalignments exist between different views, visible in the close-ups, which produce ghosting and blurring artefacts during rendering. After alignment these errors are corrected resulting in a sharp texture as is visible in the close-ups for the cloth-example.

Rendering Quality: FVR quality with the proposed multi-layer texture vs. direct resampling of the captured multi-view video is evaluated using an open-source render [20] as a benchmark. *Structural Similarity Index Measure* (SSIM) [23] which has been shown to correlate with perceived image quality is used to evaluate the rendering quality. Evaluation is performed for views mid-way between the capture cameras to test the hardest FVR case. Figure 4 presents two evaluations of rendering quality for Dan/Cloth datasets: (a) with respect to resampling optimisation approach (no-optimisation (NO), spatial optimisation only (SO) and spatio-temporal optimisation (TO)); and (b) with respect to texture image resolution using TO. This demonstrates that the optimisation method has no effect on the rendering quality for $N_L > 2$, as it is essentially the same texture information just assigned to different layers. Secondly evaluation of texture resolution shows that rendering quality remains the same for $> 1024^2$. Importantly for the Dan character dataset captured, with surrounding cameras, rendering quality remains constant for $N_L \geq 3$ indicating only 3 layers are required.

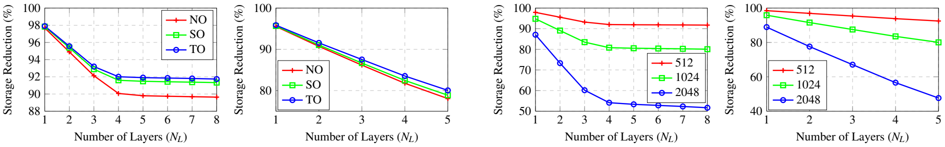


(a) Evaluation of optimisation: Dan at 512 resolution (left); Cloth at 1024 resolution (right)

(b) Evaluation of resolution: Dan using TO (left); Cloth using TO (right)

Figure 4: Free-viewpoint rendering quality for multi-layer texture vs. raw multi-view video

Representation Size: Evaluation of the multi-layer representation vs. captured video size is presented in Figure 5: (a) shows that spatio-temporal optimisation gives the best compression due to the increase in coherence in space and time; and (b) shows the size with increasing texture resolution. For the Cloth dataset, with N_L=N_C=5, using TO further reduces the overall storage by a further 2% compared to NO, this represents a 20MB reduction. Table 1 shows results for all datasets for 512 and 1024 texture sizes after MPEG video-compression compared to the captured data compressed using the same codec.



(a) Evaluation of optimisation: Dan at 512 resolution (left); Cloth at 1024 resolution (right)

(b) Evaluation of texture resolution: Dan using TO (left); Cloth using TO (right)

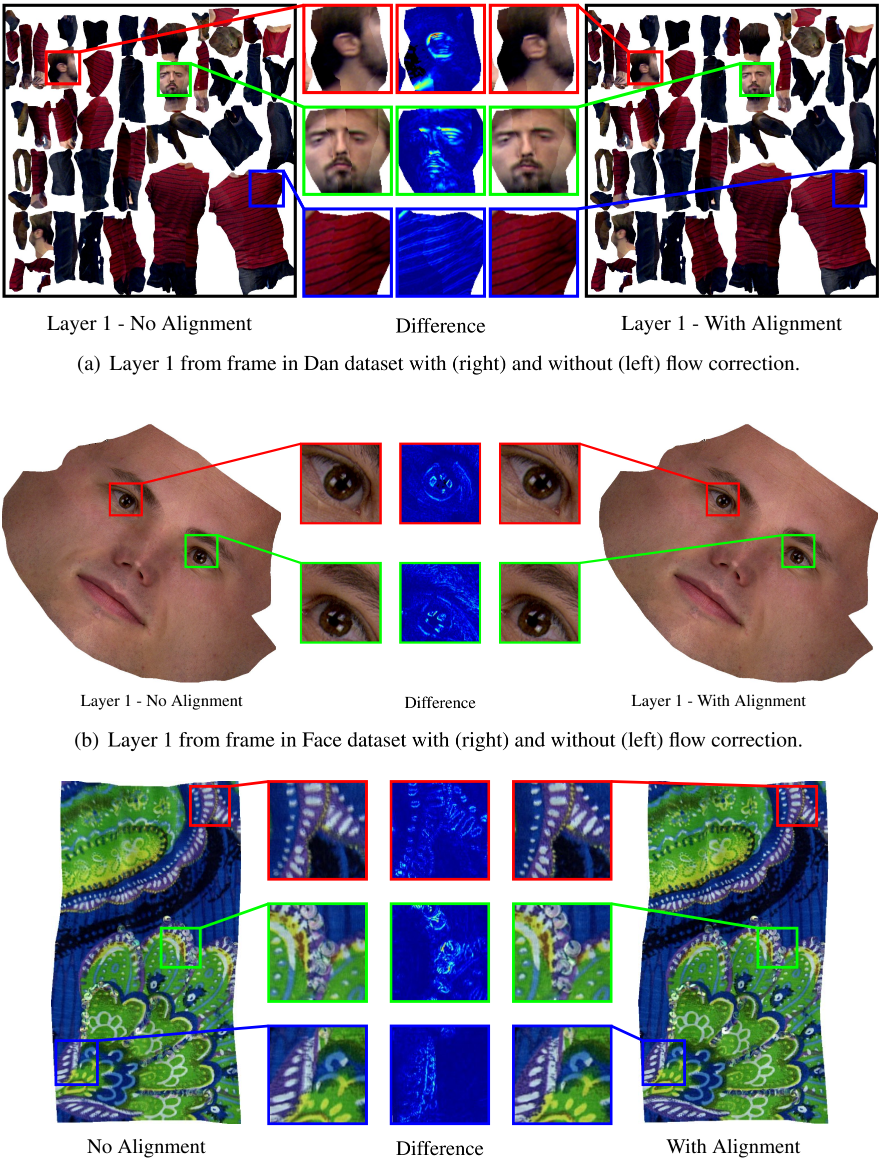
Figure 5: Representation size reduction for multi-layer texture vs. captured multi-view video

Dataset	N_C	N_T	Captured Video (MB)		Multi-layer Texture Video (MB)			
			Raw	Compressed	512		1024	
Character 1	8	31	1800	61	4.5	(93%)	10	(84%)
Cloth	5	310	11400	906	42	(95%)	112	(88%)
Dan	8	27	1600	57	3.8	(93%)	9	(84%)
Face	5	355	13100	386	26	(93%)	72	(81%)

Table 1: Dataset size for captured data and multi-layer texture video compression for resolution 512 and 1024 (Using TO and $N_L = 3$)

6 Conclusions

A method has been presented for optimisation of the resampling from multi-view video sequences of a reconstructed surface into a multi-layer 2D texture map representation to obtain a compact, spatially and temporal coherent representation that minimises the loss of information from the captured data to maintain FVR quality. Spatio-temporal optimisation is introduced to enforce consistency of camera sampling both spatially across the surface and temporally. This is combined with a surface-based optical flow alignment of the multiple view image projection on the reconstructed surface to minimise artefacts due to errors in geometry and camera calibration. This allows pre-computation and storage of the alignment for efficient high-quality rendering without blur and ghosting artefacts. In order to efficiently represent the appearance information from multiple views, a multi-layer texture map representation with layers ordered according to surface visibility is proposed. This represents view-dependent dynamic surface appearance detail for high-quality FVR. Typically 3-4 texture layers are required for capture systems with 8 cameras eliminating the inherent redundancy in multiple view capture without any significant loss of visual detail. Quantitative evaluation is performed on multiple view datasets for dynamic sequences of cloth, faces, and people. This demonstrates that the proposed approach results in an efficient representation that preserves the visual quality of the captured multi-view video for FVR whilst achieving approximately >90% reduction in size. The approach achieves compact representation of multiple view sequences to support video-based FVR without compromising visual quality or the requirement for large amounts of storage and bandwidth.



(a) Layer 1 from frame in Dan dataset with (right) and without (left) flow correction.

(b) Layer 1 from frame in Face dataset with (right) and without (left) flow correction.

(c) Single texture map from cloth dataset with (right) and without (left) flow correction

Figure 6: Results of surface-based optical flow alignment of appearance from multiple views

7 Acknowledgements

This research was supported by the EU-FP7 project RE@CT, BBC/EPSRC iCase Studentship and EPSRC Visual Media Platform Grant EP/F02827X. The authors would also like to thank Martin Klaudiny for providing the face and cloth datasets used in the evaluation.

References

[1] E. Aganj, P. Monasse, and R. Keriven. Multi-view texturing of imprecise mesh. In *Asian Conference on Computer Vision*, 2009.

[2] K. Alahari, P. Kohli, and P.H.S. Torr. Reduce, reuse & recycle: Efficiently solving multi-label MRFs. In *CVPR*, pages 1–8, 2008.

[3] C. Allene, J.P. Pons, and R. Keriven. Seamless image-based texture atlases using multi-band blending. In *ICPR*, pages 1–4, 2008.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans.PAMI*, 23(11):1222–1239, 2001.

[5] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global Non-rigid Alignment of Surface Sequences. *International Journal of Computer Vision*, 102(1-3):256–270, 2012.

[6] C Cagniart, E Boyer, and S Ilic. Probabilistic deformable surface tracking from multiple videos. *ECCV 2010*, pages 1–14, 2010.

[7] D. Casas, M. Volino, J. Collomosse, and A. Hilton. 4d video textures for interactive character appearance. *Computer Graphics Forum (Proc. EUROGRAPHICS 2014)*, 33 (2):371–380, 2014.

[8] E. de Aguiar, C. Stoll, and C. Theobalt. Performance capture from sparse multi-view video. In *ACM SIGGRAPH*, pages 1–10, 2008.

[9] P. Debevec, Y. Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *ACM SIGGRAPH*, 1998.

[10] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating Textures. *Computer Graphics Forum*, 27(2): 409–418, 2008.

[11] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. Seamless Montage for Texturing Models. *Computer Graphics Forum*, 29(2):479–486, 2010.

[12] B. Goldluecke and D. Cremers. Superresolution texture maps for multiview reconstruction. In *ICCV*, 2009.

[13] Z. Jankó and J.P. Pons. Spatio-temporal image-based texture atlases for dynamic 3-D models. In *ICCV Workshops*, pages 1646–1653, 2009.

[14] M. Klaudiny, C. Budd, and A. Hilton. Towards optimal non-rigid surface tracking. In *ECCV*, pages 743–756, 2012.

[15] V. Lempitsky and D. Ivanov. Seamless Mosaicing of Image-Based Texture Maps. In *CVPR*, pages 1–6, 2007.

[16] D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *ACM SIGGRAPH*, pages 471–478, 2000.

[17] Alex Rav-Acha, Pushmeet Kohli, Carsten Rother, and Andrew Fitzgibbon. Unwrap mosaics: A new representation for video editing. *ACM Trans. on Graphics*, 27(3): 17:1–17:11, 2008.

[18] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In *IEEE International Conference on Computer Vision*, pages 915–922, 2003.

[19] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, pages 21–31, 2007.

[20] J. Starck, J. Kilner, and A. Hilton. A Free-Viewpoint Video Renderer. *Journal of Graphics, GPU, and Game Tools*, 14(3):57–72, 2009.

[21] V. Tsiminaki, J.-S. Franco, and E. Boyer. High-resolution 3D Shape Texture from Multiple Videos. In *CVPR*, pages 1—8, 2014.

[22] D Vlasic, I Baran, W Matusik, and J Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH*, page 1, 2008.

[23] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600—612, 2004.

[24] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Proc. ACM SIGGRAPH*, 2004.