# Volumetric performance capture from minimal camera viewpoints

Andrew Gilbert[1],Marco Volino[1], John Collomosse[1,2],Adrian Hilton[1]

[1]Centre for Vision Speech and Signal Processing,
University of Surrey
[2]Creative Intelligence Lab,
Adobe Research

**Abstract.** We present a convolutional autoencoder that enables high fidelity volumetric reconstructions of human performance to be captured from multi-view video comprising only a small set of camera views. Our method yields similar end-to-end reconstruction error to that of a probabilistic visual hull computed using significantly more (double or more) viewpoints. We use a deep prior implicitly learned by the autoencoder trained over a dataset of view-ablated multi-view video footage of a wide range of subjects and actions. This opens up the possibility of high-end volumetric performance capture in on-set and prosumer scenarios where time or cost prohibit a high witness camera count.

**Fig. 1.** Two high fidelity character models (JP, Magician) where 3D geometry was fully reconstructed using only two wide-baseline camera views via our proposed method.

## 1 Introduction

Image based model reconstruction from multi-view video acquisition is enabling new forms of content production across the creative industries. In particular, the capture of human performance in three dimensions (3D) enables rendering from an arbitrary viewpoint (free-viewpoint video rendering - FVVR) [1–3] and photo-realistic replay within immersive VR/AR experiences. Commercial studios now operate for the capture of volumetric ("holographic") performance capture e. g. implementations of at Mixed Reality Capture Studios (San Francisco, London) [4] and Intel Studios (Los Angeles) both utilising over 100 camera views of a $\sim 2.5\text{m}^3$ capture volume. Whilst able to reconstruct detailed 3D models of performance, such configurations do not scale to on-set deployments where practical constraints limit the number of deployable witness cameras (e. g. due to cost or rigging overheads). The contribution of this paper is to explore whether

a deeply learned prior can be incorporated into volumetric reconstruction to minimise the number of views required at acquisition. Specifically, we investigate for the first time whether convolutional autoencoder architectures, commonly applied to visual content for de-noising and up-scaling (super-resolution), may be adapted to enhance the fidelity of volumetric reconstructions derived from just a few wide-baseline camera viewpoints. We describe a symmetric autoencoder with 3D convolutional stages capable of refining a probabilistic visual hull (PVH) [5] i. e. voxel occupancy data derived from a small set of views. Hallucinating a PVH of approximately equal fidelity to that obtainable from the same performance captured with significantly greater (double or more) camera viewpoints (Fig. 1). This extends the space of use scenarios for volumetric capture to stages with low camera counts, prosumer scenarios where cost similarly limits the number of available camera views, or settings where volumetric capture is not possible due to restrictions on camera placement and cost such as sports events [6].

## 2 Related Work

Volumetric performance capture pipelines typically fuse imagery from multiple wide baseline viewpoints [1, 7] equispaced around the capture volume. Initially, an estimate of volume occupancy is obtained by fusing silhouettes across views to yield a volumetric [8] or polyhedral [9] "visual hull" of the performer. Stereo-matching and volume optimisation subsequently fuse appearance data to refine the volume estimate ultimately yielding a textured mesh model [3, 10]. In the case of video, a 4D alignment step is applied to conform 3D mesh topology over time [11]. Reconstruction error can be mitigated by temporally propagating error through a soft i. e. probabilistic visual hull (PVH) [5] estimate. Or where practical by increasing the number of camera views since view sparsity limits the ability to resolve fine volume detail leading to the introduction of phantom volumes. Shape refinement and hole filling has been explored with a LSTM and 3D convolutional model [12] for objects. 3D ShapeNets by Wu [13], learnt the distribution of 3D objects across arbitrary poses and was able to discover hierarchical compositional part representation automatically for object recognition and shape completion while Sharma learnt the shape distribution of objects to enhance corrupted 3D shapes [14]

Our work is inspired by contemporary super-resolution (SR) algorithms that apply learned priors to enhance visual detail in images. Classical approaches to image restoration and SR combine multiple data sources (e. g. multiple images obtained at sub-pixel misalignments [15], fusing these within a regularisation constraint e. g. total variation [16]. SR has been applied also to volumetric data in microscopy [17] via depth of field, and multi-spectral sensing data [18] via sparse coding. Most recently, deep learning has been applied in the form of convolutional neural network (CNN) autoencoders for image [19, 20] and video-upscaling [21]. Symmetric autoencoders effectively learn an image transformation between clean and synthetically noisy images [22] and are effective at noise reduction e. g. due

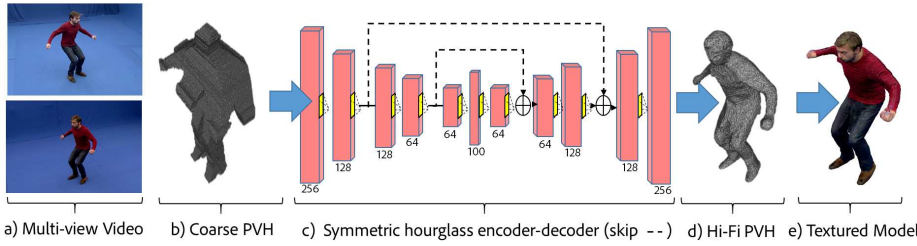


**Fig. 2.** Overview and autoencoder architecture. A coarse PVH (b) captured using minimal camera views (a) is encoded into a latent representation via 3D convolutional and full-connected layers (c). The decoder uses the latent representation to synthesise an output PVH of identical size but improved fidelity (d) which is subsequently meshed and textured to yield the performance capture model; meshing/texturing (e) is not a contribution of this paper. The encoder-decoder is optimised during training using exemplar PVH pairs of the coarse and Hi-Fi PVH volumes.

to image compression. Similarly, Dong [23] trained end-to-end networks to learn image up-scaling.

Whilst we share the high-level goal of learning deep models for detail enhancement, our work differs from prior work including deep autoencoders in several respects. We are dealing with volumetric (PVH) data and seek not to up-scale (increase resolution) as in SR, but instead, enhance detail within a constant-sized voxel grid to simulate the benefit of having additional viewpoints available during the formation of the PVH. This motivates the exploration of alternative (3D) convolutional architectures and training methodologies.

## 3 Minimal Camera Volumetric Reconstruction

The goal of our method is to learn a generative model for high fidelity 3D volume reconstruction given a low number of wide baseline camera views. We first describe the convolutional autoencoder architecture used to learn this model using a training set of sub-volume pairs sampled from full volumetric reconstructions (PVHs) of performance obtained using differing camera counts (Sec. 3.1). By using a PVH we are able to process wide baseline views, that would cause failure for a correspondence based method. Our process for refining the PVH echos the stages employed in traditional image de-noising. First, a pre-processing step (adapted from [5]) reconstructs a coarse PVH using a limited number of cameras. This low quality result will contain phantom limbs and blocky false positive voxels (Fig. 2b). Next, a latent feature representation of the PVH (akin to the low-fidelity image in traditional pipelines) is deeply encoded via a series of convolution layers. We then perform non-linear mapping decoding the latent feature space to a high fidelity PVH (akin to the high-fidelity image). The reconstruction is performed in a piece-wise fashion using densely overlapping sub-volumes, This mitigates the instabilities and memory constraints of training and inference on a network with a large receptive (volumetric) field (Sec. 3.2). The high fidelity

PVH is then meshed and textured with appearance data from the camera views yielding a video-realistic character model (Sec. 3.3). Note that the final stage *is not a contribution of this paper*, rather we demonstrate the benefits of the PVH refinement using the method of Casas *et al.* [3] but any textured meshing pipeline could be substituted as a post-process.

### 3.1 Volumetric Autoencoder

We wish to learn a deep representation given input tensor $\mathbf{V_L} \in \mathbb{R}^{X \times Y \times Z \times 1}$, where the single channel encodes the probability of volume occupancy $p(X, Y, Z)$ derived from a PVH obtained using a low camera count (eq.5). We wish to train a deep representation to solve the prediction problem $\mathbf{V_H} = \mathcal{F}(\mathbf{V_L})$ for similarly encoded tensor $\mathbf{V_H} \in \mathbb{R}^{X \times Y \times Z \times 1}$ derived from a higher fidelity PVH of identical dimension obtained using a higher camera count. Function $\mathcal{F}$ is learned using a CNN specifically a convolutional autoencoder consisting of successive three-dimensional (3D) alternate convolutional filtering operations and down- or up-sampling with non linear activation layers. Fig. 2 illustrates our architecture which has symmetric structure with skip connections bridging hourglass encoder-decoder stages, the full network parameters are:

$n_e$ =[64,64,128,128,256]
$n_d$ = [256,128,128,64,64]
$k_e$ = [3,3,3,3,3]
$k_d$= [3,3,3,3,3]
$k_s$ = [0,1,0,1,0]
NumEpoch = 10

where $k[i]$ indicates the kernel size and $n[i]$ is the number of filters at layer $i$ for the encoder ($e$) and decoder ($d$) parameters respectively. The location of the two skip connections are indicated by $s$ and link two groups of convolutional layers to their corresponding mirrored up-convolutional layer. The passed convolutional feature maps are summed to the up-convolutional feature maps element-wise and passed to the next layer after rectification. The central fully-connected layer encodes the 100-D latent representation.

Learning the end-to-end mapping from blocky volumes generated from a small number of camera viewpoints to cleaner high fidelity volumes, as if made by a greater number of camera viewpoints, requires estimation of the weights $\phi$ in $\mathcal{F}$ represented by the convolutional and deconvolutional kernels. Specifically, given a collection of $N$ training sample pairs $x^i, z^i$, where $x^i \in \mathbf{V_L}$ is an instance of a low camera count volume and $z^i \in \mathbf{V_H}$ is the high camera count output volume provided as a groundtruth, we minimise the Mean Squared Error (MSE) at the output of the decoder across $N = X \times Y \times Z$ voxels:

$$\mathcal{L}(\phi) = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{F}(x^i : \phi) - z^i\|_2^2. \quad (1)$$

To train $\mathcal{F}$ we use Adadelta [24] an extension of Adagrad that seeks to reduce it's aggressive, radically diminishing learning rates, restricting the window of accumulated past gradients to some fixed size $w$. Given the amount of data and variation in it due to the use of patches the number of epochs required for the approach to converge is small at around 5 to 10 epochs.

**Skip Connections** Deeper networks in image restoration tasks can suffer from performance degradation. Given the increased number of convolutional layers, finer image details can be lost or corrupted, as given a compact latent feature abstraction, the recovery of all the image detail is an under-determined problem. This issue is exasperated by the need to reconstruct the additional dimension in volumetric data. Deeper networks also often suffer from vanishing gradients and become much harder to train. In the spirit of highway [25] and deep residual networks [26], we add skip connections between two corresponding convolutional and deconvolutional layers as shown in Fig. 2. These connections mitigate detail loss by feeding forward higher frequency content to enable up-convolutional stages to recover a sharper volume. Skip connections also benefit back-propagation to lower layers, enhancing the stability of training. Our proposed skip connections differ from that proposed in recent image restoration work [25, 26] which concern only smoother optimisation. Instead, we pass the feature activation's at intervals of every two convolutional layers to their mirrored up-convolutional layers to enhance reconstruction detail.

### 3.2 Volumetric Reconstruction and Sampling

The low-fidelity input PVH ($\mathbf{V_L}$) is reconstructed using a variant of [5]. We assume a capture volume observed by a limited number $C$ of camera views $c = [1, C]$ for which extrinsic parameters $\{R_c, COP_c\}$ (camera orientation and focal point) and intrinsic parameters $\{f_c, o_c^x, o_c^y\}$ (focal length, and 2D optical centre) are known, and for which soft foreground mattes are available from each camera image $I_c$ using background subtraction $\mathcal{BG}$.

The studio capture volume is finely decimated into voxels ${\mathbf{V_L}}^i = \left[\, v_x^i \; v_y^i \; v_z^i \,\right]$ for $i = [1, \ldots, |\mathbf{V_L}|]$; each voxel is approximately 5mm$^3$ in size. The point $(x_c, y_c)$ is the point within $I_c$ to which ${\mathbf{V_L}}^i$ projects in a given view:

$$x[{\mathbf{V_L}}^i] = \frac{f_c v_x^i}{v_z^i} + o_c^x \quad \text{and} \quad y[{\mathbf{V_L}}^i] = \frac{f_c v_y^i}{v_z^i} + o_c^y, \quad \text{where} \tag{2}$$

$$\left[\, v_x^i \; v_y^i \; v_z^i \,\right] = COP_c - R_c^{-1} V_L^i. \tag{3}$$

The probability of the voxel being part of the performer in a given view $c$ is:

$$p({\mathbf{V_L}}^i | c) = \mathcal{BG}(x[{\mathbf{V_L}}^i], y[{\mathbf{V_L}}^i]). \tag{4}$$

The overall likelihood of occupancy for a given voxel $p({\mathbf{V_L}}^i)$ is:

$$p({\mathbf{V_L}}^i) = \prod_{i=1}^{C} 1/(1 + e^{p({\mathbf{V_L}}^i | c)}). \tag{5}$$

We compute $p(\mathbf{V_L}^i)$ for all voxels to create the PVH for volume $\mathbf{V_L}$.

In practice, the extent of $\mathbf{V_L}$ is limited to a sub-volume (a 3D "patch") of the capture volume. Patches are densely sampled to cover the capture volume, each of which is processed through $\mathcal{F}$ independently at both training and inference time. Similar to prior image super-resolution and de-noising work [23] this makes tractable the processing of large capture volumes without requiring excessively large receptive fields or up-convolutional layer counts in the CNN. In Sec. 4.1 we evaluate the impact of differing degrees of patch overlap during dense sampling. For efficiency we ignore any patches where $\sum_i p(\mathbf{V_L}^i) = 0$.

### 3.3 Meshing and texturing

Given $\mathbf{V_H}$ inferred from the network we produce a "4D" (i. e. moving 3D) performance capture. To generate the mesh for a given frame, the PVH is converted to a vertex and face based mesh using the marching-cubes algorithm. The iterative process fits vertices to the PVH output by the CNN using the marching cubes algorithm [27] with a dynamically chosen threshold, thus producing a high-resolution triangle mesh, that is used as the geometric proxy for resampling of the scene appearance onto the texture. Without loss of generality, we texture the mesh using the approach of Casas *et al.* [3] where a virtual camera view $I_{c*}$ is synthesised in the renderer by compositing the appearance sampled from the camera views $I_{1,\ldots,C}$ closest to that virtual viewpoint.


**Fig. 3.** Samples of the multi-view video datasets used to evaluate our method.

## 4 Experiments and Discussion

We evaluate the quantitative improvement in reconstruction accuracy, as well as the qualitative improvement in visual fidelity, due to the proposed method. Reconstruction accuracy is evaluated using two public multi-view video datasets of human performance; *TotalCapture* [28] (8 camera dataset of 5 subjects performing 4 actions with 3 repetitions at 60Hz in 360° arrangement) and *Human3.6M* [29] (4 camera view dataset of 10 subjects performing 210 actions at 50Hz in a 360° arrangement). Perceptual quality of textured models is evaluated using the public 4D datasets *Dan:JumpLong* [3], *JP:Flashkick* [30], *JP:Lock2Pop* [30], and *Magician* [31][1] (see Fig. 3 for samples of each dataset).

[1] We use the datasets released publicly at http://cvssp.org/data/cvssp3d/

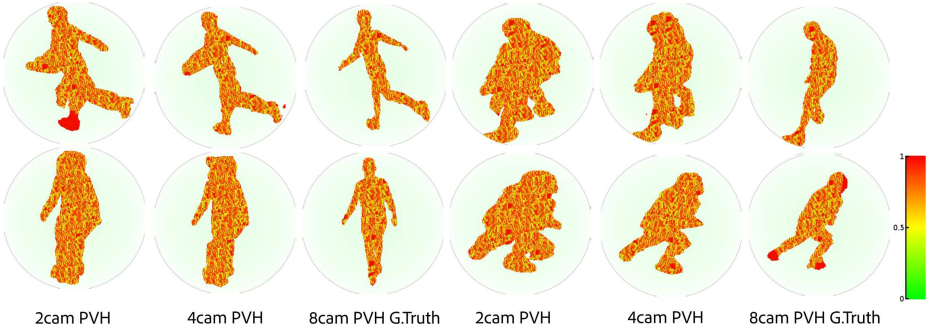


**Fig. 4.** Visualisation of raw PVH occupancy volumes estimated with C=2,4,8 views using standard method (i. e. without enhancement via our approach). PVH is a probability between 0 and 1 of the subject's occupancy. This data forms the input to our auto-encoder and illustrates the phantom volumes and artefacts to contend with at $C = \{2, 4\}$ versus the $C = 8$ ground-truth (GT) for this dataset (*TotalCapture).*

### 4.1 Evaluating Reconstruction Accuracy

We study the accuracy gain due to our method by ablating the set of camera views available on *TotalCapture*. The autoencoder model is trained using high fidelity PVHs obtained using all ($C = 8$) views of the dataset, and corresponding low fidelity PVHs obtained using fewer views (we train for $C = 2$ and $C = 4$ random neighbouring views). The model is then tested on held-out footage to determine the degree to which it can reconstruct a high fidelity PVH from the ablated set of camera views. The dataset consists of a total of four male and one female subjects each performing four diverse performances, repeated three times: *ROM, Walking, Acting and Freestyle*, and each sequence lasts around 3000-5000 frames. The train and test partitions are formed wrt. to the subjects and sequences, the training consists of *ROM1,2,3*; *Walking1,3*; *Freestyle1,2* and *Acting1,2* on subjects 1,2 and 3. The test set is the performances Freestyle3 (**FS3**), Acting (**A3**) and Walking2 (**W2**) on subjects 1,2,3,4 and 5. This split allows for separate evaluation on unseen and on seen subjects but always on unseen sequences.

The PVH is set to $z \in \mathbb{R}^{256\times256\times256}$. The sub-volume ('patch') size i. e. receptive field of the autoencoder ($\mathbf{V_L}$ and $\mathbf{V_H} \in \mathbb{R}^{n\times n\times n}$ is varied across $n = \{16, 32, 64\}$ the latter being a degenerate case where the entire volume is scaled and passed through the CNN in effect a global versus patch based filter of the volume. Patches are sampled with varying degrees of overlap; overlapping densely every 8, 16 or 32 voxels (Table 1). The PVH at $C = 8$ provides a ground-truth for comparison, whilst the $C = \{2, 4\}$ input covers at most a narrow 90° view of the scene. Prior to refinement via the autoencoder, the ablated view PVH data exhibits phantom extremities and lacks fine-grained detail, particularly at $C = 2$ (Fig. 4). These crude volumes would be unsuitable for reconstruction with texture as they do not reflect the true geometry and would cause severe visual misalignments when camera texture is projected onto the model. Applying

our autoencoder method to clean up and hallucinate a volume equivalent to one produced by the unabated $C = 8$ camera viewpoints solves this issue.

Table 1 quantifies error between the unablated ($C = 8$) and the reconstructed volumes for $C = \{2, 4\}$ view PVH data, baselining these against $C = \{2, 4\}$ PVH prior to enhancement via the auto-encoder (*input*). To measure the performance we compute the average per-frame MSE of the probability of occupancy across each sequence. The 2 and 4 camera PVH volume prior to enhancement is also shown and our results indicate a reduction in MSE of around 4 times through our approach when 2 cameras views are used for the input and a halving of MSE for a PVH formed from 4 cameras. We observe that $C = 4$ in a 180° arc around the subject perform slightly better than $C = 2$ neighbouring views in a 90° arc. However, the performance decrease is minimal for the greatly increased operational flexibility that a 2 camera deployment provides. In all cases, MSE is more than halved (up to 34% lower) using our refined PVH for a reduced number of views. Using only 2 cameras, a comparable volume to that reconstructed from a full 360° $C = 8$ setup can be produced. Qualitative results of using only 2 and 4 camera viewpoint to construct the volume are shown in Figure 5, where high quality reconstructions are possible despite the presence of phantom limbs and extensive false volumes in the input PVH. The bottom line includes results, from increasingly wide baseline cameras, separated by 45°, 90°, and 135°. Furthermore, the patch overlap is examined with the steps of 8,16 and 32. When sampled at 32 voxel increments i. e. without any overlap, performance is noticeably worse. This distinction between the patch overlap (16) and not (32) is visualised in Fig. 7. In all cases, performance is slightly better when testing on seen versus unseen subjects.

| Patch | NumCams | SeenSubjects(S1,2,3) | | | UnseenSubjects(S4,5) | | | Mean |
|---|---|---|---|---|---|---|---|---|
| Overlap | C | W2 | FS3 | A3 | W2 | FS3 | A3 | |
| Input | 2 | 19.1 | 28.5 | 23.9 | 23.4 | 27.5 | 25.2 | 24.6 |
| Input | 4 | 11.4 | 16.5 | 12.5 | 12.0 | 15.2 | 14.2 | 11.6 |
| 8 | 2 | 5.49 | 9.98 | 6.94 | 5.46 | 9.86 | 8.79 | 7.75 |
| 16 | 2 | 5.43 | 10.03 | 6.70 | 5.34 | 10.05 | 8.71 | 7.71 |
| 32 | 2 | 6.21 | 12.75 | 8.08 | 5.98 | 11.88 | 10.30 | 9.20 |
| 8 | 4 | 5.01 | 9.07 | 6.48 | 4.98 | 9.81 | 8.61 | 7.33 |
| 16 | 4 | 5.49 | 9.56 | 6.58 | 5.12 | 10.01 | 8.81 | 7.60 |
| 32 | 4 | 5.98 | 10.02 | 7.85 | 5.32 | 10.85 | 9.21 | 8.28 |

**Table 1.** Quantitative performance of volumetric reconstruction on the *TotalCapture* dataset using 2-4 cameras prior to our approach (Input) and after, versus unablated groundtruth using 8 cameras (error as MSE $\times 10^{-3}$). Patch size is $32^3$ voxels; patch overlap of 32 implies no overlap. Our method reduces reconstruction error to 34% of the baseline (Input) for 2 views.


**Fig. 5.** Qualitative visual comparison of a PVH before (left) and after (right) enhancement, showing detail improvement from $C = \{2, 4\}$ views *(TotalCapture)*. False colour volume occupancy (PVH) and groundtruth $C = 8$ PVH. Bottom line indicates performance for different pairs of cameras separated by increased amounts

**Cross-dataset generalisation** Given that the learned model on *TotalCapture* can improve the fidelity of a PVH acquired with 2-4 views to approximate a PVH reconstructed from 8 views, we explore the performance of the same model on a second dataset (*Human3.6M*) which only has on $C = 4$ views. The *Human3.6M* PVH models are poor quality as there are only 4 cameras at body height in four corners of a studio covering a relatively large capture area. This causes phantom parts and ghosting to occur. Examples of the PVH reconstructed using $C = \{2, 4\}$ views on *Human3.6M* are shown in Fig. 6 (red). These volumes are of poorer quality, even for 4 camera reconstructions, primarily due to the cameras being closer to the ground causing greater occlusion. However, we are able to transfer our trained CNN models for $2 \mapsto 8$ and $4 \mapsto 8$ views on *TotalCapture* without any further training, to hallucinate volumes as if 8 cameras were used at acquisition. Fig. 6 visualises the enhanced fidelity due to significantly reduced phantom volumes that would otherwise frustrate efforts to render the volume. $C = 4$ result provides a more complete volume but slightly enlarged. Quantitatively, the MSE of the input PVH with $C = 2$ against the *groundtruth* $C = 4$ PVH across the test datasets of S9 and S11 is $17.4 \times 10^{-3}$. However, after using our trained CNN model on the $C = 2$ input PVH this MSE is reduced to $12.3 \times 10^{-3}$, mirroring the qualitative improvement shown in Figure 6.

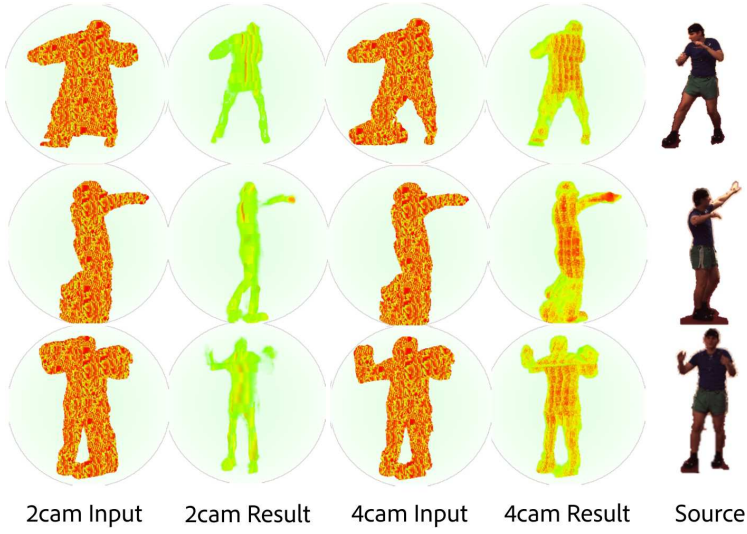


**Fig. 6.** Qualitative visual comparison of a PVH before (left) and after (right) enhancement, showing detail improvement from $C = \{2, 4\}$ views *(Human3.6M)*. False colour volume occupancy (PVH) and source footage.

**Receptive Field Size** The use of densely sampled sub-volumes (patches) rather than global processing of the PVH is necessary for computational tractability of volumes at $\mathbb{R}^{256\times256\times256}$, since the 3D convolutional stages greatly increase the number of network parameters and GPU memory footprint for batches during training. However, a hypothesis could be that the use of patches ignores the global context that the network could be learning about the subjects thus increasing error. Therefore we performed an experiment on the *TotalCapture* dataset using the network with a modified input vector of $z \in \mathbb{R}^{64\times64\times64}$, therefore making each voxel around $30\text{mm}^3$, against standard $p \in \mathbb{R}^{32\times32\times32}$, $p \in \mathbb{R}^{16\times16\times16}$ and $p \in \mathbb{R}^{8\times8\times8}$ patches sampled from the same $z \in \mathbb{R}^{64\times64\times64}$ vector, with a patch sampling overlap of 8, 16 and 32. Quantitative results of the average MSE against the groundtruth 8 camera reconstruction volume are shown in table 2 and qualitative results are shown in Figure 7.

Comparing the performance of the whole volume against patch based methods shows little change both quantitatively and qualitatively, providing that overlapping patches are utilised (therefore an overlap of 8 and 8 or 16 for $p \in \mathbb{R}^{16\times16\times16}$ and $p \in \mathbb{R}^{32\times32\times32}$ respectively. Therefore we can conclude that there is no requirement for global semantics to be learned as separate patches provide a measured compromise against the computational costs of training using a single global volume. However, the benefit of using patches is that much larger PVH can be processed, as in our experiments ($256^3$ voxels).

### 4.2 4D Character Reconstruction

We explore the efficacy of our approach as a pre-process to a state of the art 4D model reconstruction technique [3]. We use three popular 4D datasets (*J-*

| Patch Size | Patch Overlap | NumCams C | SeenSubjects(S1,2,3) W2 | FS3 | A3 | UnseenSubjects(S4,5) W2 | FS3 | A3 | Mean |
|---|---|---|---|---|---|---|---|---|---|
| Input | - | 2 | 20.1 | 24.2 | 22.3 | 23.5 | 25.7 | 26.8 | 23.8 |
| Input | - | 4 | 9.9 | 14.2 | 13.5 | 11.8 | 14.1 | 13.9 | 12.9 |
| 64 | - | 2 | 4.34 | 6.45 | 5.78 | 5.01 | 7.45 | 6.98 | 6.00 |
| 16 | 8 | 2 | 4.43 | 6.42 | 5.65 | 4.99 | 7.56 | 7.23 | 6.05 |
| 16 | 16 | 2 | 5.45 | 7.03 | 6.03 | 6.56 | 8.02 | 7.98 | 6.85 |
| 32 | 8 | 2 | 4.56 | 6.47 | 5.48 | 5.13 | 7.98 | 6.90 | 6.10 |
| 32 | 16 | 2 | 4.42 | 6.52 | 5.63 | 5.23 | 7.78 | 6.97 | 6.10 |
| 32 | 32 | 2 | 5.67 | 7.34 | 6.34 | 7.02 | 8.87 | 8.03 | 7.20 |

**Table 2.** Quantifying the effect of patch (sub-volume) size and patch overlap during dense sampling of the PVH; *TotalCapture* dataset (error as MSE $\times 10^{-3}$).

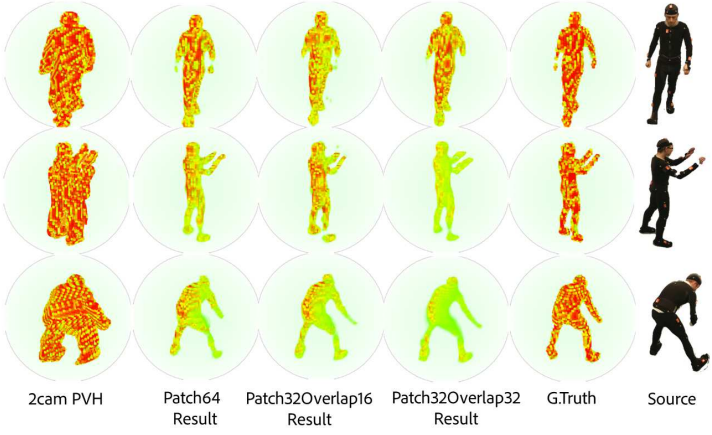


**Fig. 7.** Visual comparison accompanying quantitative data in Tbl. 2 comparing the efficacy of different patch sizes and overlaps (where a patch size of 64 implies whole volume processing).

*P*, *Dan*, *Magician*) intended to be reconstructed from a PVH derived from 8 cameras in a 360° configuration. We pick a subset of 2 neighbouring views at random from the set of 8, compute the low fidelity PVH from those views, and use our proposed method to enhance the fidelity of the PVH prior to running the reconstruction process [3] and obtaining model geometry (Sec. 3.3). The geometric proxy recovered via [3] is then textured using all views. The purpose of the test is to assess the impact of any incorrect geometry on texture alignment.

The datasets all comprise a single performer indoors in a $3m^2$ capture volume. The cameras are HD resolution running at 30Hz. Across all datasets, there are a total of 20 sequences of duration 80-3000 frames. We randomly select for test sequences: *Dan:JumpLong*, *JP:FlashKick*, *JP:Lock2Pop* and *Magician*; the remaining 16 sequences and a total of 5000 frames used as training. Given the lower number of frames available for training, the autoencoder is initially trained

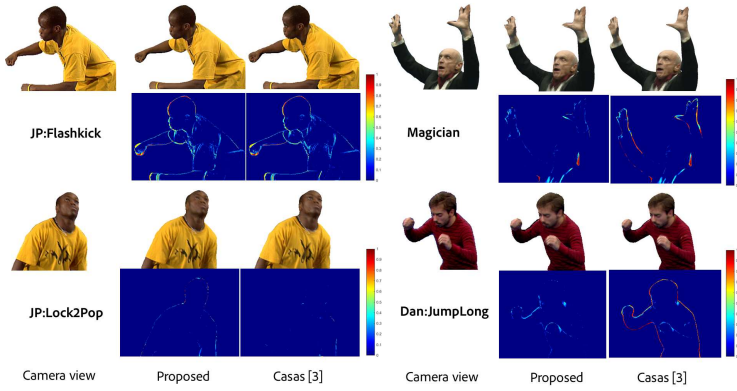


**Fig. 8.** Visual comparison of reconstructions from 2- (our) and 8-view (baseline) PVHs rendered from the viewpoint of an unused camera. The difference images (SSIM) show only minor differences relative to the real camera footage, with 2- and 8- reconstructions near identical. The error is quantified in Tbl. 3 and AMT user study (Tbl. 4).

on *TotalCapture* dataset per Sec. 4.1 then fine-tuned (with unfixed weights) using these 5000 frames. We quantify the visual fidelity of our output by rendering it from a virtual viewpoint coinciding with one of the 6 ablated viewpoints (picked randomly). This enables a direct pixel comparison between our rendering and the original camera data for that ablated view. As a baseline, we also compare our rendering against a baseline built using all 8 views using Casas [3] with identical parameters. Each frame of test data thus yields a triplet of results for comparison; 2-view PVH, 8-view PVH, and real footage from the viewpoint.

Fig. 8 presents a visual comparison for a representative triplet from each of the 4 test data. In particular, we are examining the differences in geometry which would manifest e. g. via texture misalignment or spurious mesh facets that would cause texture artefacts. The results are nearly indistinguishable with only minor texture artefacts present; a high quality result considering only 2 views are used for estimate the geometry. Table 3 quantifies performance using two metrics; PSNR and structural similarity (SSIM) [32], which closely correlates with perceptual quality. The metrics compare the 2-view and 8-view reconstructions to the camera footage which is considered to be the ground truth.

| Method | Dan JumpLong | | JP flashkick | | JP lock2pop | | Magic Magician | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Casas [3] | 38.0 | 0.903 | 31.8 | 0.893 | 32.4 | 0.893 | 38.1 | 90.4 | 35.1 | 0.898 |
| Proposed | 37.5 | 0.902 | 33.6 | 0.896 | 32.3 | 0.893 | 36.1 | 90.3 | 34.9 | 0.899 |

**Table 3.** Quantifying 4D reconstruction fidelity in terms of PSNR and SSIM averaged across frames of the sequence. We compare running [3] over our proposed output; a PVH recovered from 2 views via the autoencoder, against a baseline reconstructed directly from an 8 view PVH. The reconstruction errors are very similar, indicating our model correctly learns to hallucinate structure from the missing views.

The main sources of error between the rendered frames and the original images are found in the high frequency areas such as the face and hands, where additional vertices could provide greater detail. However, the overall reconstruction is impressive considering the poor quality of the input PVH due to the minimal camera view count.

**Perceptual User Study** We conducted a study via Amazon Mechanical Turk (AMT) to compare the performance of our rendering to the 8-view baseline. A total of 500 frames sampled from the four 4D test sequences is reconstructed as above, yielding 500 image triplets. The camera view was presented to the participant alongside the 2- and 8-view reconstructions in random order. Participants were asked to "identify the 3D model that is closest to the real camera image". Each result was presented 15 times, gathering in total 7763 annotations, from 343 unique users. Tbl. 4 reports the preferences expressed. It was our expectation that the preference would be around random at 50%, and over the 7.8K results, yet our approach was chosen as most similar to the real camera view 50.7% of the time. An unpaired t-test indicates the likelihood of identical preference is $p > 0.9984$. Given also the near-identical SSIM and PSNR scores we can conclude that despite only using 2 camera viewpoints our reconstructions are statistically indistinguishable from those sourced using the full 8 camera viewpoints.

| Sequence | Our Approach | Casas [3] |
|---|---|---|
| Dan:JumpLong | 43.5 % | 56.3% |
| JP:Flashkick | 53.2 % | 46.7% |
| JP:Lock2Pop | 57.7% | 42.2% |
| Magician | 48.2% | 51.7% |
| Mean | 50.7 | 49.2% |
| Standard Deviation | 6.15% | 6.11% |

**Table 4.** Perceptual user study (7.8k annotations). 334 AMT participants were asked to "identify the 3D model that is closest to the real camera image" and could not perceive a difference between the 2- an 8-view reconstructed models.

### 4.3 Failure cases

Despite the excellent performance of our approach at reconstructing view impoverished scenes, Fig. 9 highlights failure cases sometimes encountered by the proposed method. The use of the soft mattes from the 2D images to form the PVH can limit performance e. g. in Fig. 9(a) the initial coarse PVH input has a large horizontal hole and this isn't compensated for by the deeply learned prior; in general we find the prior learns to erode phantom volumes instead of dilating existing volumes. Fig. 9(b) illustrates that sometimes the extremities of the arms are missed, due to ambiguities in the input PVH. Finally, Fig. 9(c), indicates a

reconstruction failure due to incomplete removal of a phantom limb, caused by inaccurate geometry created from the PVH volume.

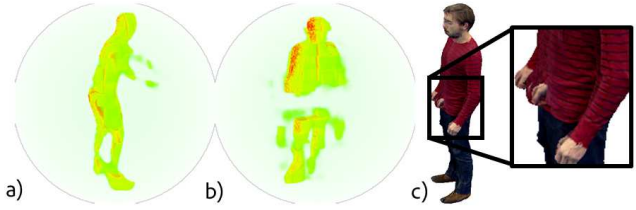


**Fig. 9.** Illustrative failure cases. Large holes due to errors in multiple 2D mattes can cause holes in the PVH that are non-recoverable. Texture misalignments can occur in areas of phantom geometry. Discussion in Sec. 4.3.

## 5 Conclusion

Volumetric performance capture from multi-view video is becoming increasingly popular in the creative industries, but reconstructing high fidelity models requires many wide-baseline views. We have shown that high fidelity 3D models can be built with as few as a couple of views, when accompanied by a deep representation prior learned via our novel autoencoder framework. We demonstrated that the models reconstructed via our method are quantitatively similar (Tables 1,2) and perceptually indistinguishable (AMT study, Table 4) from models reconstructed from considerably more camera views via existing volumetric reconstruction techniques. An additional feature of our approach is that we are able to greatly reduce the computational cost of 4D character reconstruction. Whilst training the autoencoder takes several hours, computing the PVH and passing it through the trained network for inference of a higher fidelity volume is comfortably achievable at 25 fps on commodity GPU hardware. Furthermore, the cross-data set performance of the autoencoder appears strong without (Sec. 4.1) or with minimal (Sec. 4.2) fine-tuning.

Future work could include exploring the efficacy of our deep prior beyond the domain of human performance capture, or inference of meshes directly from a coarse PVH. Nevertheless, we believe these findings are promising first steps toward the commoditisation of volumetric video, unlocking broader use cases for volumetric characters in immersive content.

## Acknowledgements

The work was supported by InnovateUK via the TotalCapture project, grant agreement 102685. The work was supported in part through the donation of GPU hardware by the NVidia corporation.

## References


1. Starck, J., Kilner, J., Hilton, A.: A free-viewpoint video renderer. Journal of Graphics, GPU, and Game Tools **14**(3) (2009) 57–72
2. Tsiminaki, V., Franco, J., Boyer, E.: High resolution 3d shape texture from multiple videos. In: Proc. Comp. Vision and Pattern Recognition (CVPR). (2014)
3. Volino, M., Casas, D., Collomosse, J., Hilton, A.: 4d for interactive character appearance. In: Computer Graphics Forum (Proceedings of Eurographics 2014). (2014)
4. Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., Sullivan, S.: High-quality streamable free-viewpoint video. ACM Transactions on Graphics (TOG) **34**(4) (2015) 69
5. Grauman, K., Shakhnarovich, G., Darrell, T.: A bayesian approach to image-based visual hull reconstruction. In: Proc. CVPR. (2003)
6. Guillemaut, J.Y., Hilton, A.: Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. International journal of computer vision **93**(1) (2011) 73–100
7. Casas, D., Huang, P., Hilton, A.: Surface-based Character Animation. In Magnor, M., Grau, O., Sorkine-Hornung, O., Theobalt, C., eds.: Digital Representations of the Real World: How to Capture, Model, and Render Visual Reality. CRC Press (April 2015) 239–252
8. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Trans. Pattern Anal. Machine Intelligence **16**(2) (1994)
9. Franco, J., Boyer, E.: Exact polyhedral visual hulls. In: Proc. British Machine Vision Conf. (BMVC). (2003)
10. Volino, M., Casas, D., Collomosse, J., Hilton, A.: Optimal Representation of Multiple View Video. In: Proceedings of the British Machine Vision Conference, BMVA Press (2014)
11. C.Budd, Huang, P., Klaudinay, M., Hilton, A.: Global non-rigid alignment of surface sequences. Intl. Jnrl. Computer Vision (IJCV) **102**(1-3) (2013) 256–270
12. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. Proc. Intl. Conf. Computer Vision (ICCV'17) (2017)
13. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: IEEE conference on computer vision and pattern recognition (CVPR'15). (2015)
14. Sharma, A., Grau, O., Fritz, M.: Vconv-dae: Deep volumetric shape learning without object labels. In: European Conference on Computer Vision. (2016) 236–250
15. Fattal, R.: Image upsampling via imposed edge statistics. In: Proc. ACM SIGGRAPH. (2007)
16. Rudin, L.I., Osher, S., Fatemi, E.: Non-linear total variation based noise removal algorithms. Physics D **60**(1-4) (1992) 259–268
17. Abrahamsson, S., Blom, H., Jans, D.: Multifocus structured illumination microscopy for fast volumetric super-resolution imaging. Biomedical Optics Express **8**(9) (2017) 4135–4140
18. Aydin, V., Foroosh, H.: Volumetric super-resolution of multispectral data. In: Corr. arXiv:1705.05745v1. (2017)
19. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Proc. Neural Inf. Processing Systems (NIPS). (2012) 350–358
20. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.S.: Deep networks for image super-resolution with sparse prior. In: Proc. Intl. Conf. Computer Vision (ICCV). (2015) 370–378

21. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In: Proc. Comp. Vision and Pattern Recognition (CVPR). (2016)
22. Jain, V., Seung, H.: Natural image denoising with convolutional networks. In: Proc. Neural Inf. Processing Systems (NIPS). (2008) 769–776
23. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Machine Intelligence **38**(2) (2016) 295–307
24. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
25. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: Advances in neural information processing systems. (2015) 2377–2385
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
27. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3d surface construction algorithm. ACM Transactions on Graphics (TOG) **21**(4) (1987) 163–169
28. Trumble, M., Gilbert, A., Malleson, C., Hilton, A., Collomosse, J.: Total capture: 3d human pose estimation fusing video and inertial sensors. In: Proceedings of 28th British Machine Vision Conference. 1–13
29. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(7) (jul 2014) 1325–1339
30. Starck, J., Hilton, A.: Surface capture for performance-based animation. IEEE computer graphics and applications **27**(3) (2007)
31. Mustafa, A., Volino, M., Guillemaut, J.Y., Hilton, A.: 4d temporally coherent light-field video. 3DV 2017 Proceedings (2017)
32. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Tran. Image Processing (TIP) **13**(4) (2004) 600–612