Implicit texture mapping for multi-view video synthesis

Mohamed Ilyes Lakhal[1,2]
mohamed.ilyes.lakhal@huawei.com
Oswald Lanz[3]
lanz@inf.unibz.it
Andrea Cavallaro[1]
a.cavallaro@qmul.ac.uk

[1] Queen Mary University of London, London, UK
[2] Helsinki Research Center, Huawei Technologies Oy (Finland) Co. Ltd., Helsinki, Finland
[3] Free University of Bozen-Bolzano, Bolzano, Italy

Abstract

Multi-view video synthesis generates the scene dynamics from a viewpoint given a source view and one or more modalities of a targeted view. In this paper, we frame video synthesis as a feature learning problem and solve it as target-view motion synthesis with spatial refinement. Specifically, we propose a motion synthesis network with a novel recurrent neural layer that learns the spatio-temporal representation of the target-view. Next, a refinement network corrects the generated coarse texture by learning the residual (*i.e.* high-frequency textures) through a UNet generator. Experimental results show visual quality enhancement of the proposed pipeline over state-of-the-art methods.

1 Introduction

Multi-view synthesis generates an object or a scene from partial information of a target viewpoint [14, 16, 24]. The main challenges of this problem are maintaining temporal consistency of the synthesised motion as well as managing occluded regions of the source view. Unlike novel-view synthesis [19, 20, 34] that generates a target RGB view from a *dense* multi-camera setup, multi-view video synthesis [14] relies on a *sparse* camera setup and assumes the availability of one or more modalities of the target view *e.g.* (depth, skeleton, 3D mesh, semantic segmentation).

In this paper, we propose a learning-based approach for sparse camera setting. Inspired by the success of two-stage pipelines for pose-guided human image synthesis [13, 18], we propose View Adaptive Network (VA-Net) a neural network generator that exploits the ability of recurrent neural networks to approximate spatio-temporal target-view features. Specifically, we estimate a *foreground mask* and *optical flow* with separate networks to help guiding the network during the synthesis, and we estimate the *foreground feature representation* of the target view using only a depth prior. We use the source-view video for implicit texture mapping and, to improve the preservation of texture across views, we propose View-Adaptive LSTM (`VA-LSTM`), a recurrent neural network structure that improves the target-view feature representation of the target-view video by aggregating a texture-less represen-

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

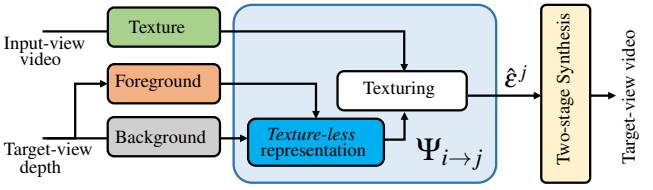

Figure 1: **Multi-view video synthesis.** The target view is estimated through an operator $\Psi_{i\to j}$ that estimates the target-view feature $\hat{\varepsilon}^j$ combining the input-view video as implicit texturing and a *Texture-less* representation, obtained through foreground estimation and background scene structure.

tation and texture mapping (Fig. 1). The resulting framework is a two-stage pipeline that first synthesizes the target-view using the feature approximation approach and then refines the synthesized video to remove artifacts produced during the first stage.

2 Related Work

The novel-view synthesis problem can be tackled using graphics-based methods, which aim at producing high-fidelity novel-view images of objects of interest. These objects can be rigid (*e.g.* cars [4]) or non-rigid (*e.g.* synthetic [37] or real [22] human bodies or animals, such as zebras [39], chimpanzees [23]). While graphics-based methods produce high-quality images, they are time expensive *i.e.* order of seconds which hinders such methods to be applied to real-time synthesis scenarios. Learning-based methods [18, 34] offer a trade-off between quality and running time. These methods rely on modalities such as skeleton [18], scene parsing [3, 7], depth [14], 3D meshes [16], or the first-frame of the novel view [24].

We categorise the synthesis methods as: *graphics* and *learning* based. The goal of *graphics* methods is to achieve photorealistic rendering of a scene at a reasonable frame-rate [7]. This is achieved by modeling the physical properties of a scene (geometry, surface properties) [30]. Mesh-based methods focus on deforming a 3D body mesh to fit onto the 2D image of a person captured from a certain viewpoint. Such methods can be classified into template-based, model-based, and free-form [36]. Despite the impressive visual quality that the *graphics* methods achieve, they fall short on the capturing system of the novel-view data. For example, for template-based methods, the human body is scanned using an acquisition setup that consists of multiple cameras placed on a dome shape. Besides the good mesh estimation of the human body, they also require a template mesh for each human body and clothing dress which makes them impractical in some scenarios.

The *learning* based approach exploits the ability of neural network encoder-decoders to synthesize the target-view. The main assumption of this class of method is the multi-view camera configuration, where the goal is to synthesize the target-view from sparse camera configuration. We group the method as pose-guided and modality-based. Pose-guided methods [15, 18, 28, 29, 38] use mainly a 2D human skeleton or a heatmap derived from the skeleton. The skeleton provides a sparse discrete 2D location that guides the generator network to transform the image in its initial pose to the desired pose. Modality-based method [14] extends the pose-guided image synthesis [18] to the video domain. View-LSTM [14] proposed to decompose the target-view feature space as a view-invariant representation, shared among all the views, and a view-specific representation. GTNet [16] estimates the foreground of the target-view video using 3D mesh correspondence, then, a generator refines the foreground

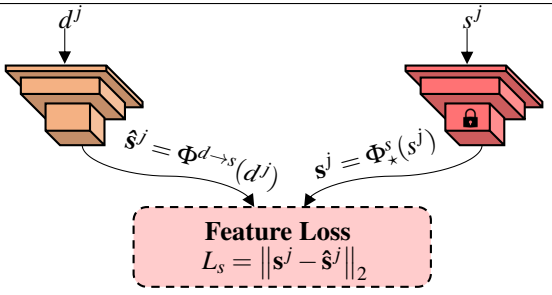

Figure 2: **Foreground feature learning.** We use a teacher-student approach [8] to estimate the feature of the foreground motion. The teacher model $\Phi^s_\star$ is pre-trained on synthesizing the target-view video using ground-truth s^j. The student model $\Phi^{d\to s}$ tries to focus on the motion contained in the depth d^j.

estimation and synthesizes the background. Dual Representation [27] learns a global representation of the scene using images from different views. The global representation is then combined with a view-dependent representation as query to synthesis the targeted view. RT-Net [24] extends the image animation problem to the multi-view setting. Given only the first frame of the target-view video, a motion transfer module animates the image based on the input-view video.

3 View Adaptive LSTM

We address the multi-view synthesis as a feature learning problem. Therefore, we present a formulation to estimate the target-view feature ε^j as a composition of a *texture-less* representation and an implicit texturing. To make use of the temporal feature learning abilities of RNN, we propose a recurrent network formulation, View Adaptive LSTM (`VA-LSTM`) that aggregates a *texture-less* representation and the input-view video used as an implicit texture mapping. We estimate the foreground motion feature from the depth sequence as the only target-view modality input. Next, we estimate the *texture-less* representation using the foreground motion and the feature representation of the background. Finally, we aggregate the *texture-less* representation along with the input-view video as texture mapping.

Foreground feature learning. To estimate the spatio-temporal feature ε^j, it is enough to estimate the feature of the foreground modality s^j, represented as P keypoints, obtained through a mapping $\Phi^s : \mathbb{R}^{\Omega_T \times P} \to \mathbb{R}^m$ where $m \in \mathbb{N}$, $\Omega_T = W \times H \times T$ is a grid of width W, height H and time span T, respectively. However, the estimation of Φ^s is intractable as the model would need to learn the representation for all possible poses and camera locations. To overcome this problem, we adopt a data-driven approach to learn another mapping, $\Phi^s_\star$, to approximate Φ^s. Specifically, we use the depth d^j to learn to focus on the motion using the mapping $\Phi^{d\to s} : \mathbb{R}^{\Omega_T} \to \mathbb{R}^m$ to estimate $\Phi^s_\star$ with a teacher-student approach [8]. In particular, we use the encoder of a synthesis generator trained using the ground-truth foreground modality. Through the network training, the feature from the encoder $\Phi^{d\to s}$ is forced to estimate the feature from $\Phi^s_\star$ (Fig. 2).

To estimate the foreground feature of s^j, we use an encoder from a pre-trained generator to force the representation $\Phi^{d\to s}(d^j)$ to approximate the motion feature as:

$$\hat{\mathbf{s}}^j = \Phi^{d\to s}(d^j) \approx \Phi^s_\star(s^j). \tag{1}$$

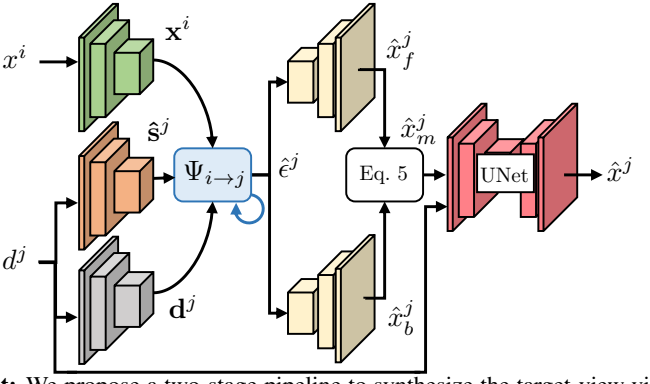

Figure 3: **VA-Net:** We propose a two-stage pipeline to synthesize the target-view video. The motion network is a two-stream generator. We estimate the foreground motion $\hat{\mathbf{s}}^j$ using the depth d^j. The operator $\Psi_{i\to j}$, implemented as `VA-LSTM`, estimates the target-view feature $\hat{\epsilon}^j$. The refinement network enhances the synthesis from the motion network using a 3D-CNN UNet architecture.

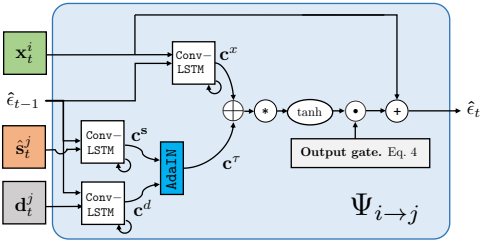

Figure 4: **View Adaptive LSTM (`VA-LSTM`).** We extend the Conv-LSTM [28] to implement $\Psi_{i\to j}$. We obtain the *texture-less* representation using `AdaIN` with $\mathbf{d}^j$ as a style and the estimated motion feature $\hat{\mathbf{s}}^j$ as content. $\mathbf{x}^j$ is used as implicit texture.

This enforces the student model $\Phi^{d\to s}$ to retain the foreground motion contained in the depth d^j. We apply a feature loss, L_s, to force $\Phi^{d\to s}$ to learn discriminative feature as $\Phi^s_\star$. The loss encourages the prediction from $\Phi^{d\to s}$ to look like the feature produced using $\Phi^s_\star$.

Texture-less representation. In the context of images, a style is defined as the texture describing the overall look *e.g.*, mountain, beach, or the artistic look *e.g.* a Van Gogh painting from a Picasso [11]. In our problem setting, the motion feature $\hat{\mathbf{s}}^j$ provides the content and the scene structure, represented by the depth feature $\mathbf{d}^j$ as the style. The intuition is to add to the motion feature representation "content" as a style, defining the scene where the motion is taking place. The style $\mathbf{d}^j$ does not provide a texture and therefore, the aggregation of these representations using an operator $\Psi_{i\to j}$ produces a *texture-less* representation.

Inspired by [8, 21], we adopt the adaptive instance normalization (`AdaIN`) module as aggregator to learn the representation τ. The use of `AdaIN` encourages the *texture-less* representation to match the mean μ and the standard deviation σ of the distribution of the ground-truth target-view samples [19, 20]. The representation is obtained as:

$$\texttt{AdaIN}(\hat{\mathbf{s}}^j,\, \mathbf{d}^j) = \sigma(\mathbf{d}^j)\left(\frac{\hat{\mathbf{s}}^j - \mu(\hat{\mathbf{s}}^j)}{\sigma(\hat{\mathbf{s}}^j)}\right) + \mu(\mathbf{d}^j). \tag{2}$$

Target-view feature approximation. The feature $\mathbf{x}^i$ is the representation of the input-view video using an encoder $\Phi^x : \mathbb{R}^{\Omega_T \times 3} \to \mathbb{R}^m$. The operator $\Psi_{i\to j} : \mathbb{R}^{2m} \to \mathbb{R}^m$ approxi-

mates the spatio-temporal feature representation of the target-view video as: $\hat{\varepsilon}^j \approx \Psi_{i\to j}(\tau^j|\mathbf{x}^i)$. We present View Adaptive LSTM (VA-LSTM) that implements $\Psi_{i\to j}$ using Conv-LSTM [26]. The memory cell of each input are extracted independently in order to retain separate memory for each of the inputs ($\mathbf{x}^i, \hat{\mathbf{s}}^j$, and $\mathbf{d}^j$) which can be seen as the accumulated spatio-temporal feature (up to time-step t) of the input on the feature space induced by Conv-LSTM. The hidden state of the combined inputs is used to approximate ε^j (Fig. 4).

The memory cell of the texture map is obtained as[1] $\mathbf{c}^x_t = \text{LSTM}(\mathbf{x}^i, \mathbf{h}_{t-1})$. The memory cell of the depth is $\mathbf{c}^d_t = \text{LSTM}(\mathbf{d}^j, \mathbf{h}_{t-1})$. Likewise, the memory cell of the foreground is computed as $\mathbf{c}^s_t = \text{LSTM}(\hat{\mathbf{s}}^j, \mathbf{h}_{t-1})$. We combine $\mathbf{c}^s_t$ and $\mathbf{c}^d_t$ to obtain the *texture-less* representation $\mathbf{c}^\tau_t$ defined in Eq. 2 as $\mathbf{c}^\tau_t = \texttt{AdaIN}(\mathbf{c}^s_t, \mathbf{c}^d_t)$. However, these LSTM modules are related through a single hidden state $\mathbf{h}$ (Eq. 4) which estimates the target-view feature ε^j.

Note that using $\mathbf{c}^\tau$ in computing $\mathbf{c}^s_t$ and $\mathbf{c}^d_t$ helps both memory cells in learning discriminative feature towards the motion in $\mathbf{c}^s_t$ and the target-view scene structure $\mathbf{c}^d_t$. The aggregator $\Psi_{i\to j}$ receives $\mathbf{x}^i$, $\hat{\mathbf{s}}^j$, and $\mathbf{d}^j$ from separate LSTM units. The output gate controls the importance of each LSTM gate state to be used by the hidden state.

$$\mathbf{o}_t = \texttt{sigmoid}\Big(\underbrace{\mathsf{W}_{xo} * \mathbf{x}^i_t + \mathsf{W}_{so} * \hat{\mathbf{s}}^j_t + \mathsf{W}_{do} * \mathbf{d}^j_t}_{\text{Input information}} + \underbrace{\mathsf{W}_{co} * \mathbf{c}^x_t + \mathsf{W}_{\tau o} * \mathbf{c}^\tau_t}_{\text{Texture map + texture-less cell}}\Big), \tag{3}$$

where $*$ is the convolution operation, and W with subscript are the learnable weights. We maintain separate the foreground $\hat{\mathbf{s}}^j$ and the background $\mathbf{d}^j$ as input information to put equal emphasis on each part.

For small camera extrinsic changes from the input-view camera, the input view feature $\mathbf{x}^i$ provides rich information on the target-view scene structure. The interpretation of this can be implemented through a skip connection [5]. The hidden state $\mathbf{h}_t$ approximates ε^j as:

$$\hat{\varepsilon}^j_t \approx \mathbf{h}_t = \mathbf{o}_t \odot \tanh\Big(\mathsf{W}_{ch} * \mathbf{c}^x_t + \mathsf{W}_{\tau h} * \mathbf{c}^\tau_t\Big) + \lambda \mathbf{x}^i_t, \tag{4}$$

where λ is the importance weight of the residual connection.

3.1 View Adaptive Network (VA-Net)

We present View Adaptive Network (VA-Net), a two-stage pipeline for multi-view video synthesis. The first stage, Motion network, synthesizes low-frequency details using our VA-LSTM to estimate the spatio-temporal target-view feature $\hat{\varepsilon}^j$. The second stage, Refinement Network, refines the synthesis and corrects artifacts (Fig. 3).

Motion network. We follow a two-stream [32] architecture to synthesizes the foreground and the background separately. Let the network outputs $\hat{x}^j_f$, $\hat{x}^j_b$ represent the foreground and the background, respectively (Fig. 3).

We obtain the target-view video using the predicted $\hat{m}^j$, $\hat{\mathcal{O}}^j$, and a warping function $\mathcal{W}$.

$$\hat{x}^j_m = \mathcal{W}(\hat{x}^j_f, \hat{\mathcal{O}}^j) \odot \hat{m}^j + \hat{x}^j_b \odot (1 - \hat{m}^j), \tag{5}$$

we apply the predicted binary mask $\hat{m}^j$ to the foreground and $1 - \hat{m}^j$ to the background stream. Additionally, we guide the foreground stream to learn a residual from the synthesis of the previous frame.

[1] We use the same notation for the Conv-LSTM to represent the variables

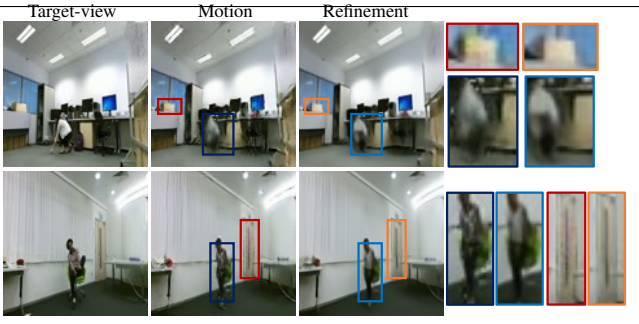

Figure 5: The Motion Network synthesizes low frequency details and the Refinement Network removes artifacts as shown in zoomed boxes (left: Motion; right: Refinement).

To train the network, we use the total loss: $L = L_r + .1L_s + .01(L_t + L_a)$, with a reconstruction loss, L_r, a feature loss, L_s, a perceptual loss, L_t, and an adversarial loss, L_a. The *reconstruction loss* uses an L_1 reconstruction term between the network output $\hat{x}^j_m$ and x^j:

$$L_r = \left\| x^j - \hat{x}^j_m \right\|_1. \tag{6}$$

The *feature loss* enforces the feature $\Phi^{d\to s}(d^j)$ to approximate $\Phi^s_\star(s^j)$ using an L_2 term:

$$L_s = \left\| \Phi^s_\star(s^j) - \Phi^{d\to s}(d^j) \right\|_2. \tag{7}$$

The *perceptual loss* projects both x^j and $\hat{x}^j_m$ using the I3D [1] perceptual network ϕ and computes the L_2 error:

$$L_t = \left\| \phi(x^j) - \phi(\hat{x}^j_m) \right\|_2. \tag{8}$$

Finally, the *adversarial loss* uses a discriminator D to distinguish true from synthesized videos on a min-max optimization:

$$L_a = \mathbb{E}[\log(D(x^i, x^j))] + \mathbb{E}[\log(1 - D(x^i, \hat{x}^j_m))]. \tag{9}$$

Refinement network. The synthesized video $\hat{x}^j_m$ from the motion network may contain some visible artifacts that decrease the overall perceptual quality of the video, a 3D UNet generator is therefore used to remove such artifacts. The final target-view video is $\hat{x}^j = \text{UNet}(\hat{x}^j_m | x^i)$ where we condition the synthesis with the input-view x^i in order to correct the texture from $\hat{x}^j_m$. To train the network, we combine a reconstruction loss, L_{SSIM}, and an edge loss, L_e. The *reconstruction loss* uses SSIM [33] to penalize high-frequency prediction:

$$L_{\text{SSIM}} = 1 - \text{SSIM}(x^j, \hat{x}^j). \tag{10}$$

The *edge loss* is based on an edge detection algorithm (we chose `Sobel` filter [12]) and penalizes the prediction by applying the pre-defined filters $\mathbf{C}_x$ and $\mathbf{C}_y$ over x^j and $\hat{x}^j$ to obtain the vertical and horizontal derivative, respectively:

$$L_e = \left\| \mathbf{C}_x * x^j - \mathbf{C}_x * \hat{x}^j \right\| + \left\| \mathbf{C}_y * x^j - \mathbf{C}_y * \hat{x}^j \right\|. \tag{11}$$

The training loss is $L_{ref} = L_{\text{SSIM}} + .1L_e$ and the value .1 is defined empirically.

Table 1: **VA-Net ablation.** FG: ablation of the foreground estimation. Teacher-student approach (`student`) vs. Dedicated encoder of the semantic segmentation s^j trained jointly with the generator (`random`). We also highlight the feature loss L_2 vs. L_σ; Skip: importance of the residual skip connection in the proposed `VA-LSTM`; H-freq: refinement network with different losses. KEY – M: mask.

Ablation	Strategy	SSIM	M-SSIM	PSNR	M-PSNR
FG	`random`	.844	.976	24.26	30.82
	`student` w/ L_σ	.847	.977	24.43	30.99
	`student` w/ L_s	.862	.978	24.71	31.15
Skip	w/o skip	.796	.966	22.55	28.12
	w skip	.862	.978	24.71	31.15
H-freq	$L_{\text{SSIM}} + L_e$	.895	.979	25.48	31.36
	L_{SSIM}	.892	.979	25.33	31.31
	L_1	.888	.979	25.56	31.45

4 Validation

We validate the proposed model and compare it with state-of-the-art models. The proposed model is implemented as ResNet [9], which is widely adopted by [14, 16, 29, 38]. We use the cross-subject split of the NTU RGB+D dataset [25] and we follow the evaluation defined in [14]. We evaluate the quality of the synthesized video with Structural Similarity (SSIM), Peak Signal-to-Noise-Ratio (PSNR) [33] and their masked versions [18]; Percentage of Correct Keypoints (PCK) [35]; and Fréchet Video Distance (FVD) [31].

Foreground feature learning. We compare the encoder that estimates the foreground motion feature, $\Phi^{d\to s}$, with the pre-trained teacher network, $\Phi^s_\star$, against `random`, a model that trains the encoder Φ^s from scratch. Next, we look at the feature loss and we replace the L_2 term with a loss based on the feature statistics L_σ, similar to [17]:
$L_\sigma = \sum \left\| \sigma(\mathbf{s}^j) - \sigma(\hat{\mathbf{s}}^j) \right\|_1 + \left\| \mu(\mathbf{s}^j) - \mu(\hat{\mathbf{s}}^j) \right\|_2 .$

Tab. 1 compares the teacher-student approach to learn the encoder Φ^s against `random` and L_σ. The improvement with the proposed teacher-student learning approach is due to the explicit loss term over the feature, instead of implicitly learning with a pixel-reconstruction loss. The L_2 feature loss improves over L_σ as $\Phi^{d\to s}$ learns the feature representation of s^j and penalizing over the mean and variance with L_σ is sub-optimal.

Motion network. Tab. 2 compares the synthesis performance and model size of variants of VA-LSTM: `Branch` ($\mathbf{x}^i$ applied to $\mathbf{d}^j$ and $\hat{\mathbf{s}}^j$ with separate `AdaIN` module), `Conv` (`AdaIN` replaced by convolution), and `Sum` (`AdaIN` replaced by summation). Because of the teacher-student approach, unlike View-LSTM we do not require an additional trainable encoder. The `AdaIN` module does not require additional trainable parameters. The encoder uses non-linear operations and therefore produces features that are related in a non-linear way. Tab. 3 compares the proposed VA-LSTM and View-LSTM with the same input modalities. VA-LSTM outperforms View-LSTM and requires fewer trainable weights. Also, dense foreground modality using the human parsing provides a richer spatial locality that improves visual quality.

To avoid extensive hyper-parameter search, we set empirically the residual factor λ to 10^{-2}. Tab. 1 shows that adding the residual skip connection improves the quality of the synthesized video. Thanks to the spatio-temporal feature learning using our `VA-LSTM`, the synthesized videos are temporally consistent which is validated by good FVD scores (Tab. 4).

Refinement network. Tab. 1 shows the benefit of the proposed loss for the refinement network. The `Sobel` loss L_e enhances the overall perceptual quality over the L_1 and using

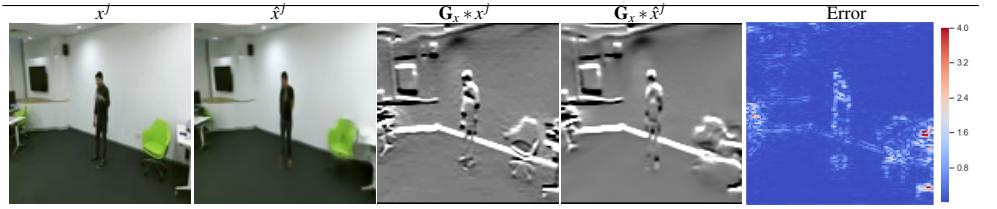

Figure 6: **Edge loss.** The refinement network was trained on penalizing the edges using `Sobel` filters over which results in better high-frequency details.

Table 2: **VA-LSTM ablation.** We replace the `AdaIN` operation to compute the *texture-less* representation with `conv`, `sum`, and a variant `branch` where `AdaIN` is applied to $\mathbf{d}^j$ and $\hat{\mathbf{s}}^j$ separately. We highlight the benefit of VA-LSTM in terms of trainable weight against View-LSTM.

Method		SSIM	M-SSIM	PSNR	M-PSNR	#param
View-LSTM		.821	.972	23.18	29.70	(14.36 + 22.41) M
VA-LSTM	+ `AdaIN`	.862	.978	24.71	31.15	13.70 M
	+ `branch`	.851	.979	24.14	31.63	15.47 M
	+ `conv`	.847	.979	24.42	31.65	13.83 M
	+ `sum`	.842	.979	24.37	31.63	13.70 M

Table 3: VA-LSTM vs. View-LSTM using the same modalities. KEY – M: mask; s^j: skeleton; $\mathcal{S}^j$: human parsing.

Model	Modalities	SSIM	M-SSIM	PSNR	M-PSNR
View-LSTM	d^j, s^j	.821	.972	23.18	29.70
	$d^j, \mathcal{S}^j$	.833	.975	23.44	30.35
VA-LSTM	d^j, s^j	.830	.976	23.78	30.89
	$d^j, \mathcal{S}^j$	.845	.980	24.50	31.70

Figure 7: **Pose-guided comparison.** Due to the lack of explicit temporal consistency term in pose-guided methods, they fail to capture the motion in the target-view resulting in misclassifying the background synthesis as well.

only the SSIM loss. It is worth noticing that the foreground quality is almost the same with all the losses. This is because the refinement mainly focused on correcting visible artifacts that are mostly present in the background. Fig. 6 shows an example of the edge loss, the heatmap of the error between the ground-truth and the synthesized video using the filter $\mathbf{C}_x$. We note good high-frequency details from the synthesized videos, we also observe similar behavior with the filter $\mathbf{C}_y$.

SOTA comparison. Tab. 4 compares the proposed VA-Net against state-of-the-art methods. Video-based methods outperform image-based methods [18, 29, 38] as 2D-CNNs do not provide temporal context and generate inconsistent background synthesis. The networks that use both input and output skeleton [29, 38] fail to properly synthesize pose in the target-

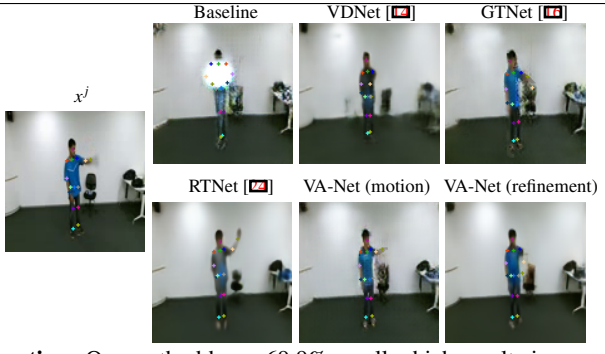

Figure 8: **Pose-estimation.** Our method has a 69.9% recall which results in many images where the pose estimator can estimate the keypoints. White image means that the pose estimator fails to estimate the keypoints.

Table 4: VA-Net against pose-guided and modality-based target-view synthesis methods. RTNet [24] (highlighted in gray) uses ground-truth first frame of x^j. KEY - M: mask; Image: image based; Gen: generator with a single decoder; Motion: motion transfer; Two-stream: generator with two decoders; Colors: **best**, second-best. * Model as reported in [24].

Method	Cat.	$\Psi_{i\to j}$	Modality	SSIM	M-SSIM	PSNR	M-PSNR	FVD	L_2	PCK 0.20	PCK 0.05	PCK 0.01	Precision	Recall	F1
PG^2 [18]	Image	CNN	s^j	.582	.954	16.90	25.87	11.84	10.91	97.8	74.7	14.4	88.1	12.8	22.4
PATN [36]	Image	CNN	s^i, s^j	.534	.948	16.24	24.55	13.11	11.68	98.0	69.7	10.2	88.4	.1	17.8
XingGAN [25]	Image	CNN	s^i, s^j	.445	.933	13.32	23.29	14.47	26.41	89.6	17.8	.01	84.5	.1	.1
VDNet	Gen	RNN	d^j, s^j	.821	.972	23.18	29.70	5.78	4.37	99.3	92.4	51.2	91.0	55.3	68.7
Baseline	Gen	CNN	d^j	.813	.965	22.82	27.87	5.28	5.80	99.4	89.4	41.5	**92.5**	30.6	46.0
RTNet [24]	Motion	RNN*	$x^j_{t=1}$	.933	–	29.07	–	–	–	–	–	–	–	–	–
	Motion	RNN	$\hat{x}^j_{t=1}$	.878	–	25.27	–	–	–	–	–	–	–	–	–
	Motion	RNN	$d^j_{t=1}$	.887	.977	25.76	30.68	4.14	4.13	99.4	93.0	53.4	91.7	56.6	70.0
GTNet	Two-stream	CNN	$\mathcal{T}^j, \mathcal{S}^j, d^j$	.823	.981	23.81	32.50	4.96	3.95	99.5	93.0	57.6	92.3	52.7	67.1
VA-Net (motion)	Two-stream	RNN	$\mathcal{S}^j, d^j$	.845	.980	24.50	31.70	**3.63**	**2.87**	99.5	**95.5**	**67.7**	91.1	**69.9**	**79.1**
	Two-stream		d^j	.862	.978	24.71	31.15	3.70	3.75	99.4	93.1	58.6	91.9	58.3	71.3
VA-Net (refinement)	Two-stream	–	–	**.895**	.979	25.48	31.36	3.67	3.46	**99.6**	94.3	59.4	91.2	65.3	76.1

view (Fig. 7). We note the benefit of two-stream architecture against using one decoder in and the baseline. Having two decoders allows the generator to focus on equal importance on each of the foreground and background. Using foreground modalities helps the foreground synthesis but cannot focus more on the overall synthesis.

RTNet [24] is not comparable to other methods since it uses the first frame of the target-view $x^j_{t=1}$. To further highlight this, we use RTNet to synthesize $\hat{x}^j_{t=1}$ instead of the ground-truth and we notice a major quality drop. We use the depth $d^j_{t=1}$ to fairly compare RTNet.The foreground synthesized is blurry and the model cannot preserve the texture well (Fig. 9). The first row of Fig. 9 shows the motion transition using a different method. Our VA-Net has fewer artifacts around the body boundaries compared to RTNet. The foreground estimation provides comparable foreground results as in GTNet which uses stronger body modalities. VA-Net has a good FVD score, which shows the advantage of the proposed `VA-LSTM` that can learn better the spatio-temporal feature $\hat{\varepsilon}^j$ and leads to better overall visual quality.

We note a good FVD score of the proposed VA-Net which shows the advantage of the proposed `VA-LSTM` that can learn better the spatio-temporal feature $\hat{\varepsilon}^j$ and leads to better overall visual quality. We also note a consistency between the FVD scores and the pose estimation (Fig. 8).

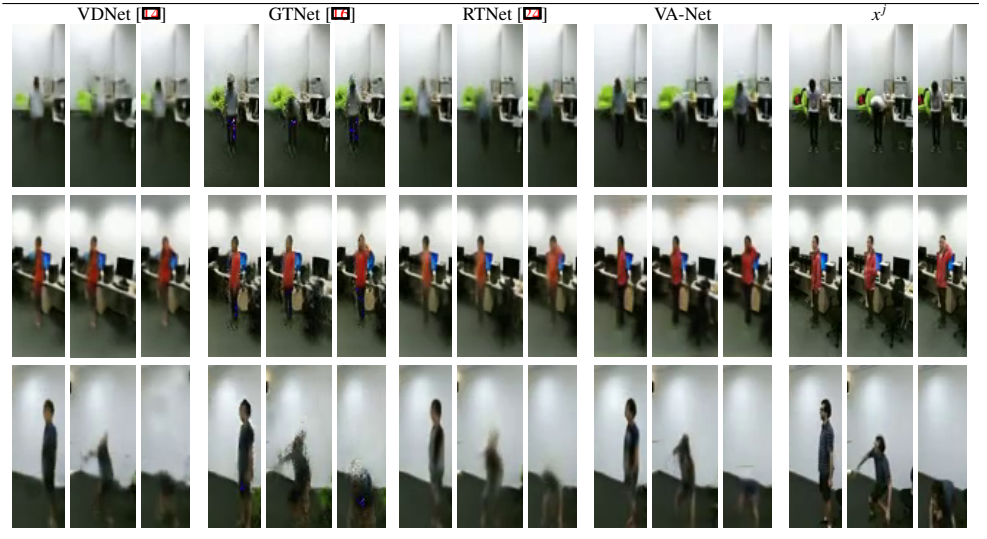

Figure 9: **Modality based comparison.** Results showing the proposed VA-Net against state-of-the methods. We highlight the motion ability and the overall synthesis quality of each method.

5 Conclusion

We addressed the multi-view video synthesis from a feature learning perspective. We presented View Adaptive LSTM (VA-LSTM) that decomposes a view as a texture-less representation and texture mapping. We tackle the synthesis as a two-stage pipeline. The first stage, motion network, uses the proposed VA-LSTM for the feature estimation. The second stage, the refinement network, uses skip connections in the UNet model to correct artifacts. We show that our estimated feature representation of the foreground obtains similar performance when using the raw foreground modality. Experimental results show that by having fewer modality the network can focus more on synthesizing the texture and have better refinement step.

Despite the great qualitative improvement in recent NeRF-based [10] methods, we believe that they are not directly applicable to our problem setting. For the foreground synthesis, body-NeRF [10] relies heavily on the SMPL estimation. On high kinematic motion, the estimation is very unstable, and thus taking the T-pose (rest pose) and warping it by motion field using the vertices as support is very prone to error. In our case, the most important part is synthesizing smooth motion transition which thanks to our VA-LSTM can handle while. It is, however, an interesting direction to combine both approaches on multi-view video synthesis.

Acknowledgements This project acknowledges the use of the ESPRC funded Tier 2 facility, JADE. Oswald Lanz gratefully acknowledges the support from Amazon AWS Machine Learning Research Awards (MLRA).

References

[1] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017. doi: 10.1109/CVPR.2017.502.

[2] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Computer Graphics Forum (Proc. of Eurographics EG)*, 2008.

[3] Tewodros Habtegebrial, Varun Jampani, Orazio Gallo, and Didier Stricker. Generative View Synthesis: From Single-view Semantics to Novel-view Images. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Nicolai Häni, Selim Engin, Jun-Jee Chao, and Volkan Isler. Continuous Object Representation Networks: Novel View Synthesis without Target View Supervision. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Arxiv*, 2015.

[7] Hsin-Ping Huang, Hung-Yu Tseng, Hsin-Ying Lee, and Jia-Bin Huang. Semantic view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[8] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. NeuMan: Neural Human Radiance Field from a Single Video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[12] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 1988.

[13] Mohamed Ilyes Lakhal, Oswald Lanz, and Andrea Cavallaro. Pose guided human image synthesis by view disentanglement and enhanced weighting loss. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[14] Mohamed Ilyes Lakhal, Oswald Lanz, and Andrea Cavallaro. View-LSTM: Novel-view video synthesis through view decomposition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[15] Mohamed Ilyes Lakhal, Oswald Lanz, and Andrea Cavallaro. Learnable masks for pose-guided view synthesis. In *IEEE International Conference on Image Processing (ICIP)*, 2019.

[16] Mohamed Ilyes Lakhal, Davide Boscaini, Fabio Poiesi, Oswald Lanz, and Andrea Cavallaro. Novel-view human action synthesis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020.

[17] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose Guided Person Image Generation. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[19] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[20] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[23] Artsiom Sanakoyeu, Vasil Khalidov, Maureen S. McCarthy, Andrea Vedaldi, and Natalia Neverova. Transferring Dense Pose to Proximal Animal Classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] Kara Marie Schatz, Erik Quintanilla, Shruti Vyas, and Yogesh S Rawat. A Recurrent Transformer Network for Novel View Action Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[25] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. In *Neural Information Processing Systems (NeurIPS)*, 2015.

[27] Sarah Shiraz, Krishna Regmi, Shruti Vyas, Yogesh Singh Rawat, and Mubarak Shah. Novel view video prediction using dual representation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2021.

[28] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable GANs for Pose-Based Human Image Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] Hao Tang, Song Bai, Li Zhang, Philip H. S. Torr, and Nicu Sebe. XingGAN for Person Image Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[30] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhöfer. State of the art on neural rendering. In *The Eurographics Association*, 2020.

[31] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new Metric for Video Generation. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, 2019.

[32] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[33] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 2004.

[34] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-End View Synthesis From a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[35] Yi Yang and Deva Ramanan. Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

[36] Tao Yu, Jianhui Zhao, Zhang Zerong, Kaiwen Guo, Dai Quionhai, Hao Li, Gerard Pons-Moll, and Yebin Liu. DoubleFusion: Real-time Capture of Human Performance with Inner Body Shape from a Depth Sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] Fang Zhao, Shengcai Liao, Kaihao Zhang, and Ling Shao. Human Parsing Based Texture Transfer from Single Image to 3D Human via Cross-View Consistency. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[38] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive Pose Attention Transfer for Person Image Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] Silvia Zuffi, Angjoo Kanazawa, Tanja Berger-Wolf, and Michael J. Black. Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture From Images "In the Wild". In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.