A Recurrent Transformer Network for Novel View Action Synthesis

Kara Marie Schatz[1], Erik Quintanilla[2], Shruti Vyas[3], and Yogesh S Rawat[3]

[1] Xavier University, Cincinnati, Ohio, USA
schatzk@xavier.edu
[2] Illinois Institute of Technology, Chicago, Illinois, USA
equintanilla@hawk.iit.edu
[3] Center for Research in Computer Vision, University of Central Florida, USA
{shruti,yogesh}@crcv.ucf.edu

Abstract. In this work, we address the problem of synthesizing human actions from *novel views*. Given an input video of an actor performing some action, we aim to synthesize a video with the same action performed from a novel view with the help of an appearance prior. We propose an end-to-end deep network to solve this problem. The proposed network utilizes the change in viewpoint to transform the action from the input view to the novel view in feature space. The transformed action is integrated with the target appearance using the proposed *recurrent transformer network*, which provides a transformed appearance for each time-step in the action sequence. The recurrent transformer network utilize *action key-points* which are determined in an unsupervised approach using the encoded action features. We also propose a *hierarchical structure* for the recurrent transformation which further improves the performance. We demonstrate the effectiveness of the proposed method through extensive experiments conducted on a large-scale multi-view action recognition NTU-RGB+D dataset. In addition, we show that the proposed method can transform the action to a novel viewpoint with an entirely different scene or actor. *The code is publicly available at https://github.com/schatzkara/cross-view-video.*

Keywords: Novel-view action synthesis, action transformation, video synthesis

1 Introduction

In recent years, we have seen a great interest from the research community in image and video synthesis [18,31,25]. It has a wide range of applications, such as data augmentation, augmented reality, and action imitation. While the research in image synthesis has seen great progress [18,7,25], synthesizing realistic videos is still a challenging problem due to its complexity and high computational requirements [31,24,8].

Some of the recent works in video synthesis have proposed the use of a prior to reduce the complexity of the problem [30,24]. The use of priors, such as

action class [8], pose information [40], and image conditioning [29], leads to a better quality when compared with the videos generated without any prior [30,24,32]. While much work is being done in this area, the existing research in video synthesis is mainly focused on single views.

In this work, we focus on the problem of video synthesis from novel viewpoints. The presence of novel views makes the video synthesis task more complex as both the action and appearance vary significantly with the change in viewpoint. There has been some work in cross-view image synthesis in which the focus is on 3D reconstruction from images [14], multi-view aggregation [9], and transforming ground and aerial images [23]. This requires transforming the appearance of one view to other novel views. In the case of videos, both the appearance as well as the action dynamics must be transformed to the target novel view, which increases the complexity of the problem.

We propose an end-to-end deep framework to solve the problem of cross-view video synthesis. The proposed framework takes a video from a source viewpoint and synthesizes the same action from a novel viewpoint with the help of an appearance prior. The prior is utilized to determine the change in viewpoint, which helps in transforming the source action to the novel viewpoint in latent representation space. The transformed action features need to be effectively integrated with the target appearance to synthesize a realistic action video. To achieve this, we propose a *novel recurrent transformer* network, which takes the transformed latent action features and recurrently transforms the appearance to generate a sequence of target action features in the latent space. The recurrent transformer network make use of *action key-points*, which are determined in an unsupervised approach, to focus on activity regions in the video. Moreover, we propose a *hierarchical structure*, which enables the network to perform the transformation at different feature scales while generating the video from a novel viewpoint.

2 Related work

Video synthesis: The research in deep generative modeling has led to significant progress in the field of image synthesis [18,16,7,25]. This is mainly attributed to the success of Generative Adversarial Networks (GANs) [10], where the realism of the synthesized video is used for the adversarial learning [20,22,24]. However, it remains very challenging to synthesize a realistic looking video due to the complexity and resource requirements of the problem. There has been some preliminary success in the task of video synthesis where the research is focused on future video prediction [22,31,3,38,4], and conditioned video generation [30,24]. The recurrent structures are also found to be effective in spatio-temporal modeling and video prediction [37,35,36]. These works utilize the memory module in the recurrent structure for predicting plausible future video frames.

Generating a video without any given prior is a difficult problem as the network has to learn from the training distribution [8]. The use of priors in video synthesis helps in reducing the complexity of the problem and makes the generation task more tractable. To this end, there has been research focus in video

synthesis where we can use another video for content [2], segmentation prior [34], target pose [33,40], or motion transfer [29,27,6] to aid the generation task. The work in [29] uses a facial image to synthesize a talking video by transferring motion from another video. Similarly, in [27,6] the authors propose to transfer motion from a video prior to a target image. Our work is related to these works in motion transfer as we are also using appearance as a prior while synthesizing the video. However, our problem is different from these in two key aspects. These methods perform image synthesis and transfer the motion from source to target image one frame at time. Whereas, we synthesize the full action video at once which is much more efficient. Also, we are focusing on novel-view synthesis while these works are based on single views.

Novel view synthesis: Cross-view synthesis of data is a challenging problem with multiple applications including augmented reality, data augmentation and view-invariant learning. The existing works in novel-view synthesis mainly focus on cross-view image synthesis [23,9] and 3D reconstruction from images [14]. In [23], the authors propose a generative adversarial network which can transform the ground-view images to aerial-view images and vice-versa. The authors in [9] utilize multiple views to render a image from unseen views with the help of a generative query network. Different from this, the authors in [14] learn a 3D representation from single image which can be used to render the image from multiple other views. All of these works perform cross-view synthesis in image domain, however we are focusing on synthesizing videos.

Cross-view video synthesis adds in more complexity to the problem by introducing action dynamics. The seminal work in cross-view video synthesis [32] proposed to learn a global scene representation which was used to synthesize a video from unseen viewpoint. However, this work was mainly focused on action classification. In [19], the authors propose to render optical flow from novel views for learning a good view-invariant action representation. This is different since they only have to predict the optical flow. In a more recent work [17], the authors propose a recurrent LSTM based network which predicts videos from novel views. Their approach utilizes a strong prior from the target view as they require a sequence of depth and skeleton maps for video synthesis. The depth and skeleton modalities are well known for activity classification and therefore have sufficient action information. Therefore, this approach does not require the transformation of the action from the source video. Our approach on the other hand only uses a single frame from the target view as a prior, and the action from the source video must be transformed to the target view.

3 Proposed Method

Given an input source action video V^i from view i and an appearance prior P^j from target view j, the proposed framework F synthesize the action video $\hat{V}^j$ as seen from view j. We can formalize the problem as,

$$\hat{V}^j = F(V^i, P^j). \tag{1}$$

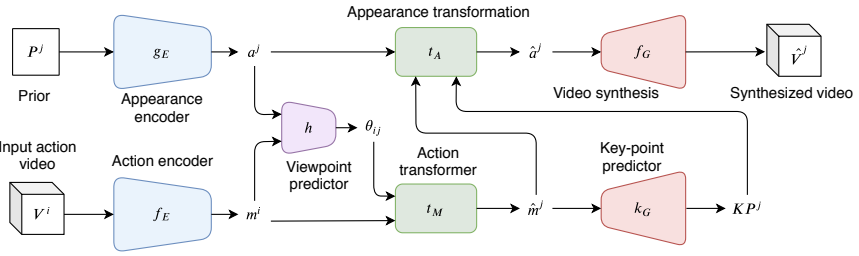

Fig. 1. Overview of the proposed framework. Given a source video V^i and target prior P^i, the proposed framework transforms the source action features m^i to target view action features $\hat{m}^j$ and use them to transform the target prior a^j for synthesizing target view action video $\hat{V}^j$. The network also utilize action key-points KP^j, which are predicted via unsupervised approach, to focus on activity regions in the video

Here $V^j \in \mathbb{R}^{T\times H\times W\times 3}$ and $\hat{V}^i \in \mathbb{R}^{T\times W\times H\times 3}$ with T frames, height H, and width W, and $P^j \in \mathbb{R}^{W\times H\times 3}$ with height H, and width W. An overview of the architecture for the proposed framework is shown in Figure 1. The framework F consists of an action transformer network t_M which takes the change in viewpoint θ_{ij} and action encoding m^i to transform the action features from source view i to the target view j.

$$\hat{m}^j = t_M(m^i, \theta_{ij}). \tag{2}$$

Here $\hat{m}^j$ is the transformed action features.

The transformed action features $\hat{m}^j$ are passed to an appearance transformer network t_A which transforms the prior features a^j to generate appearance features $\hat{a}^j$ for synthesizing the video $\hat{V}^j$ from the target view j. In addition to this, the framework also consists of a key-point predictor network k_G which predicts action key-points KP^j to focus on activity regions in the video. Finally, a generator network f_G takes the transformed appearance features $\hat{a}^j$ along with the predicted action key-points KP^j to synthesize the target video $\hat{V}^j$.

$$\hat{V}^j = f_G(\hat{a}^j, KP^j). \tag{3}$$

We will cover the details of the components in the next subsections.

3.1 Action transformation

The action transformation is performed in the latent representation of the action. The input video V^i is first encoded using a video encoder f_E to get the latent action representation $m^i \in \mathbb{R}^{T_r\times H_r\times W_r\times C_r}$. Here T_r, H_r, W_r, and C_r represents the temporal extent, height, width and, number channels in the latent representation respectively. We utilize 3D convolution based network [5] to encode the input video V^i which is effective in extracting spatio-temporal features.

The transformation of action from one view-point i to another view-point j requires the relative change in view-point θ_{ij}. We propose a view-point change prediction network h which utilize the encoded features m^i and prior information P^j from the target view-point to predict the change in the view-point. The appearance prior P^j is first encoded using a visual encoder g_E which extracts the latent representation $a^j \in \mathbb{R}^{H_r \times W_r \times C_r}$ for the target view j. We use a 2D convolution based network [28] for encoding the appearance prior.

Viewpoint change predictor: The viewpoint change prediction network h estimates the relative change in view-point $\hat{\theta}_{ij}$ between the source and target views,

$$\hat{\theta}_{ij} = h(m^i, a^j). \tag{4}$$

This change in view-point $\hat{\theta}_{ij} \in (-\pi/2, \pi/2)$ is used to perform action transformation from view i to view j. We are only considering a maximum change of $\pi/2$ (which is there in the used datasets) in our experiments, but a maximum change of π can also be used by predicting cosine and sin values. The prediction of change in view-point within the framework avoids the need of providing this externally while generating an action video from target novel view.

The temporal extent of the action representation m^i is not important for inferring the viewpoint. Therefore, average pooling is performed on m^i along the temporal extent to reduce the representation to single frame. The compressed features m^i from the source view is then combined with the features a^j from the target view using a concatenation operation along the channel axis. These concatenated features are passed through two blocks that consist of a 2D convolutional layer followed by ReLU activation and average pooling. A 3x3 kernel is used for the convolutional layers, and a 2x2 kernel is used for the average pooling layer with a stride of 2. Finally, the features are flattened and passed through a single fully connected layer that predicts the angular viewpoint change, $\hat{\theta}_{ij}$.

The change in view-point prediction loss L_{vp} is computed as the mean squared error between the ground truth θ_{ij} and the predicted $\hat{\theta}_{ij}$ viewpoint change.

$$L_{vp} = \frac{1}{N} \sum_{k=1}^{N} (\hat{\theta}_{ij}^k - \theta_{ij}^k)^2 \tag{5}$$

Here, N represents the number of samples.

Action transformer network: The action transformer network t_M computes action features $\hat{m}^j$ for the target view by transforming the action representation m^i of the given input view based on the angular viewpoint change θ_{ij}. The angular change is first expanded to $\mathbb{R}^{T_r \times H_r \times W_r \times 1}$ by repeating it for each spatio-temporal location in the latent action representation. Then, it is passed through two layers of 3D convolutions before concatenating with m^i along the channel dimension. These concatenated features are then passed through three 3D convolutional layers each followed by ReLU activation. A 3x3 kernel is used

for each of the convolutional layers preserving the spatial and temporal dimension of the representation using padding. The transformed action features $\hat{m}^j \in \mathbb{R}^{T_r \times H_r \times W_r \times C_r}$ are then used to transform the appearance features a^j for generating the target video.

Note that the ground truth value of angular view-point change θ_{ij} is used for the transformation during the training phase for a stable network training. However, the predicted view-point change $\hat{\theta}_{ij}$ is used during network inference.

3.2 Action key-point detection

We are interested in the key-point regions which are important from action point of view. These action key-points will be used in an attention mechanism during transforming the prior as well as during synthesizing the target action video. We take an unsupervised approach to detect these action key-points [13]. The key-point detector k_G takes the transformed action features $\hat{m}^j$ and first generate N_k action heatmaps $Z_k \in \mathbb{R}^{T_k \times H_k \times W_k \times N_k}$ corresponding to N_k action key-points. We use a 3D convolutional based network with ReLU activation to predict these heatmaps. The action key-point detector network k_G consists of four convolution layers. The convolution layers are used in conjunction with upsampling via trilinear interpolation to increase the temporal and spatial extent of the predicted heatmaps.

The action keypoints KP^j are extracted from Z_k as Gaussian heatmaps. The reason is that they can be effectively used as attention in the convolution based prior transformation network t_A as well as video synthesis network f_G. The first step in generating the Gaussian heatmaps is to determine the most active position in these heatmaps. The x_m and y_m coordinates for the keypoints are deteremined separately by first computing the mean along all the rows Z_k^x or columns Z_k^y and applying a softmax along the remaining spatial dimension.

$$Z_k^x = \frac{1}{H_k} \sum_{j=1}^{H_k} Z_k^j, \quad Z_k^y = \frac{1}{W_k} \sum_{j=1}^{W_k} Z_k^j. \tag{6}$$

Now the active position along each dimension can be determined by applying the softmax normalization to these vectors,

$$x_m = \frac{\sum_{j=1}^{H_k} j e^{Z_k^x(j)}}{\sum_{j=1}^{H_k} e^{Z_k^x(j)}}, \quad y_m = \frac{\sum_{j=1}^{W_k} j e^{Z_k^y(j)}}{\sum_{j=1}^{W_k} e^{Z_k^y(j)}}. \tag{7}$$

These predicted active coordinates $u_m = (x_m, y_m)$ are used as a mean for the Gaussian which replace the heatmaps. The Gaussian with a small standard deviation σ are centered at u_m to generate an action key-point,

$$KP_i^j = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{1}{2}((u - u_m)/\sigma)^2. \tag{8}$$

Here, KP_i^j is any instance i of an action key-point and $u \in (H_k \times W_k)$.

3.3 Appearance transformer network

The appearance prior P^j has to be transformed according the action features $\hat{m}^j$ to generate the action video $\hat{V}^j$. We propose a recurrent approach which utilize the action features $\hat{m}^j$ for the transformation at latent space. The detected action key-points helps the appearance transformation in two ways. They will be useful in the separation of foreground and background features, and help in focusing on the action regions while transforming the foreground appearance. The appearance transformer network t_A has a recurrent structure based on convolutional Gated Recurrent Unit (Conv-GRU) [1] which takes the prior a^j as input along with transformed action features $\hat{m}^j$ and action key-points KP^j and predict transformed appearance features $\hat{a}^j$. Formally,

$$\hat{a}^j = t_A(a^j, \hat{m}^j, KP^j), \tag{9}$$

where $\hat{a}^j \in \mathbb{R}^{T_r \times H_r \times W_r \times C_r}$.

At each time step t, t_A takes the appearance latent representation $\hat{a}^j_{t-1}$ and transform the appearance to $\hat{a}^j_t$ with the help of action latent features $\hat{m}^j_t$ and action key-points KP^j_t. The first step is to determine the background b_t and foreground f_t based on the appearance, action features, and action key-points.

$$\begin{aligned} b_t &= \sigma(W_b * < \hat{a}_{t-1}, \hat{m}^j_t, KP^j_t >) \\ f_t &= \sigma(W_f * < \hat{a}_{t-1}, \hat{m}^j_t, KP^j_t >) \end{aligned} \tag{10}$$

Here, $*$ denotes convolution operation, $<>$ denotes concatenation operation along channels axis, and W_b and W_f are parameters for 2D kernels. The background features are selected from the appearance as,

$$b^f_t = b_t \odot \hat{a}_{t-1}, \tag{11}$$

where $\odot$ denotes element-wise multiplication. The foreground appearance features are transformed with the help of action features and action key-points.

$$f^f_t = tanh(W^f_f * < \hat{m}^j_t, KP^j_t >, f_t \odot \hat{a}_{t-1} >). \tag{12}$$

Finally, the transformed foreground features are combined with the background features to get the integrated transformed appearance features,

$$\hat{a}^j_t = b^f_t + (1 - b_t) \odot f^f_t. \tag{13}$$

The transformed appearance features from different time-steps are combined together to form $\hat{a}^j$ which is used for synthesizing the action video.

Hierarchical transformation: We propose to perform the appearance transformation on the prior at different resolution of latent representations. The key idea is to transform the appearance at both coarse as well as fine level which helps in improving the performance of video synthesis. In case of prior, appearance features a^j are extracted from multiple higher level layers in g_E. Similarly, action

features m^i are extracted from multiple layers of video encoder f_E. The same set of predicted action key-points KP^j are used at different hierarchies after performing either average pooling or upsampling depending upon the resolution of action and appearance features.

The action features m^i from different levels are first passed through the action transformer network t_M to generate the transformed action features $\hat{m}^j$. Since t_M is a fully convolutional network, it is shared for action transformation at different levels of hierarchy. Similarly, the appearance transformer network t_A is also convolutional, therefore it is also shared by all the levels for performing appearance transformation. The sharing capability of t_M and t_A helps in reducing the number of parameters in the network and it also makes the transformation more robust. The transformed appearance features $\hat{a}^j$ from multiple levels are passed to the video synthesis network f_G and integrated at different layers with matching resolution.

3.4 Action synthesis

The final component of the proposed framework is the action generator network f_G which synthesize the target action video $\hat{V}^j \in \mathbb{R}^{T\times H\times W\times 3}$ using the approximated appearance features $\hat{a}^j \in \mathbb{R}^{T_r\times H_r\times W_r\times C_r}$ and the predicted action key-points $KP^j \in \mathbb{R}^{T_k\times H_k\times W_k\times C_k}$. The action key-points KP^j helps the generator to focus on action regions in the video. They are predicted at a higher resolution in comparison with $\hat{a}^j$. Therefore the key-points are first average pooled down to the same temporal and spatial size as $\hat{a}^j$ and then these are concatenated along the channel dimension. The generator is based on 3D convolutions with ReLU activation and upsampling. The convolutional layers all use 3x3 kernels with zero padding and the upsampling layers use trilinear interpolation. The final layer is followed by a sigmoid activation which generates the target action video $\hat{V}^j$.

We use a pixel-wise reconstruction loss L_r, which is computed using mean squared error between the synthesized video $\hat{V}^j$ and the ground truth video V^j. We also use an adversarial loss [10] and a perceptual loss [15] to help in improving the performance of video synthesis. The adversarial loss L_{adv} is computed using a 3D convolution based discriminator D [5] which critiques whether the synthesized action video is realistic or not. We train F and D alternatively using a standard GAN framework [10]. The adversarial loss is computed as,

$$L_{adv} = E_{x\sim\mathcal{S}(i,j)}[log(1 - D(F(x)))], \tag{14}$$

where L_{adv} represents the adversarial loss and $\mathcal{S}(i, j)$ is the distribution of video and prior pair (V^i, P^j) from view i and j respectively. The discriminator loss is,

$$L_d = \max_D \Big(E_{x\sim\boldsymbol{V_{gt}}}[log(D(x))] + E_{x\sim\mathcal{S}(i,j)}[log(1 - D(F(x)))]\Big), \tag{15}$$

where L_d is the discriminator loss and $\boldsymbol{V_{gt}}$ is the set of real action videos. To improve the visual quality of the synthesized video frames, we also use a perceptual loss L_p which computes the error at feature level. We utilize a pre-trained

VGG-16 network [28] to compute the loss at frame level which is averaged over all the frames in the synthesized video. The loss is computed as mean squared error between the features from predicted video and ground truth video. The overall loss to train the full network is defined as,

$$L = \lambda_{vp} L_{vp} + \lambda_r L_r + \lambda_{adv} L_{adv} + \lambda_p L_p. \tag{16}$$

Here $\lambda_{vp}, \lambda_r, \lambda_{adv}$, and λ_p are loss weights which are determined experimentally. We use $\lambda_{vp} = 1, \lambda_r = 1, \lambda_{adv} = 0.1$, and $\lambda_p = 0.1$ in all our experiments.

3.5 Implementation and training details

We use a modified VGG-16 network [28] as our appearance encoder g_E where we use the features after the first ten convolution layers to get a^j. For the video encoder, we use a 3D convolution based I3D network [5] and extract m^i from the $'Mixed\_5c'$ convolution layer. We use a resolution of 112x112 as input for both source video as well as the prior with 16 frames in the video. The video frames are sampled at 15 frames per second to include more motion int he videos. We compute the ground truth angular viewpoint change based on configuration parameters provided with the dataset.

The appearance features a^j are encoded as $14 \times 14 \times 256$ and the action features are encoded as $4 \times 14 \times 14 \times 256$ with $T_r = 4, H_r = 14, W_r = 14$, and $C_r = 256$. The action key-points are predicted at a resolution of $16 \times 56 \times 56 \times 32$ with $T_k = 16, H_k = 56, W_k = 56$, and $C_k = 32$. A standard deviation of 0.1 is used to compute the Gaussian maps for the action key-points. We use a 3D convolution based I3D network [5] as the discriminator D to compute adversarial loss L_{adv}, where the last prediction layer is modified for binary prediction. The perceptual loss L_p is computed with the help of a pre-trained VGG-16 network [28] where we take 512 dimension features from last layer of the network. We use Adam optimizer, with a learning rate of 1e-4. We implemented the code in Pytorch and perform our experiments on Titan-X GPU with a batch size of 14.

4 Experiments

In this section, we provide details of the experiments we performed to validate the effectiveness of the proposed method. Apart from the qualitative evaluation, we also provide frame level Structural Similarity Index Measure (SSIM) [39] and Peak Signal to Noise Ratio (PSNR) [11] for quantitative evaluation.

4.1 Dataset

We conducted our experiments on the NTU-RGB+D Dataset [26], which is the largest multi-view action dataset containing over 56,000 videos. It has more than 4 million video frames depicting either one or two humans performing the actions. There are a total of 60 different actions depicted in the dataset using

Table 1. A comparison of SSIM scores of all the combinations of three views along with the average score with existing approaches. The scores for VDG [12] and PG^2 [21] are shown as reported by the authors of VDNet [17]

Model	**Pair-view SSIM Score**						**Average**
	$v_1 \rightarrow v_2$	$v_1 \rightarrow v_3$	$v_2 \rightarrow v_1$	$v_2 \rightarrow v_3$	$v_3 \rightarrow v_1$	$v_3 \rightarrow v_2$	
VDG [12]	.502 ± .058	.543 ± .068	.584 ± .060	.563 ± .062	.611 ± .077	.522 ± .063	.554 ± .075
PG^2 [21]	.499 ± .071	.561 ± .060	.600 ± .064	.557 ± .071	.598 ± .075	.543 ± .066	.560 ± .076
VRNet [32]	-	-	-	-	-	-	0.68
ResNet [17]	.705 ± .115	.735 ± .095	.717 ± .130	.690 ± .122	.734 ± .127	.669 ± .150	.708 ± .127
VDNet [17]	.789 ± .076	.791 ± .069	.800 ± .076	.765 ± .079	.797 ± .067	.756 ± .089	.783 ± .078
Proposed	.974 ± .021	.975 ± .021	.975 ± .019	.971 ± .021	.974 ± .017	.971 ± .022	.973 ± .020

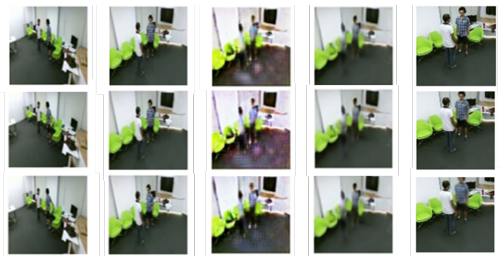

Fig. 2. Comparison of the generated video frames using our method with existing approaches. The video frames are from position 1, 4, and 8. Column 1: source, column 2: target, column 3: ResNet [17], column 4: VDNet [17], and column 5 proposed method

40 different actors. Three different cameras are used at various height settings to capture videos from 80 different viewpoints. The cameras are always placed 45°apart, so they are at -45°, 0°, and +45°. Each actor or actor pair performs each action twice: once facing the left camera and once facing the right camera. This allows the videos to span viewpoints over a total of 90°. For our experiments, we use the subject split as described by the authors in [26].

4.2 Evaluation

We have shown the SSIM score for the synthesized videos of all the combinations of the three views on the test set of NTU-RGB+D in Table 1. We observe that the scores are low for pair v2 and v3 when compared with other pairs. These two views are at +/- 90°from each other and therefore the transformation is more challenging than other pairs where the transformation is within +/- 45°. In figure 3, we have shown some sample video frames synthesized using the proposed method along with the ground truth video frames. We can observe that the network has no issue in rendering the background and the dynamics of the action is also quite visible along the video frames. The motion can be seen in the sequence of frames, but we can also observe that the motion is not well defined with some blur in the activity region. This is interesting as we are not

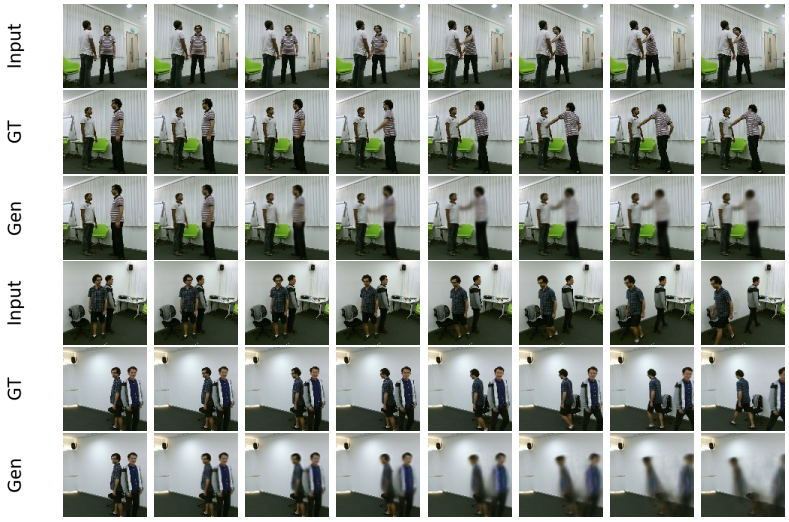

Fig. 3. Synthesized video frames using the proposed model. For each sample example, the top row contains 8 frames of the ground truth video for the novel view and the bottom row contains the same 8 frames of the generated video for the novel view. Our model predicts 16 frames in a video and for each of these examples, frames 1, 3, 5, 7, 9, 11, 13, and 15 are shown. GT: ground-truth, Gen: generated frames

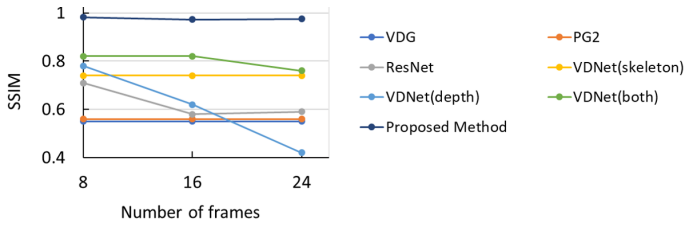

Fig. 4. A Comparison of the variation of SSIM score with varying number of generated video frames with existing approaches

utilizing any action prior from the novel target view, like [17], which make use of depth and skeleton sequences from the target view.

In Figure 5, we have shown some examples of predicted action key-points for a video. We can observe that the predicted action key-points are near the activity region in the frames and therefore they act as an attention mechanism for the recurrent transformer network to focus on activity regions.

Comparison: We also compared our approach with existing methods in Table 1. We observe that the proposed method outperforms all the other approaches in terms of SSIM score. We also present a qualitative comparison with [17]. The comparison is shown in Figure 2. We observe that the video frames generated by

Fig. 5. Action key-points: The center of predicted action key-points shown on the sequence of example video frames. We can observe that the predicted key-points are located close to the performed action in the video frame

Fig. 6. Ablation on components of the proposed model: Synthesized video frames using different model variations. Column 1: source video, column-2: target video, column-3: basic model, column-4: w/ motion transformation, column-5 w/ hierarchical transformation, and column-6 appearance transformation

ResNet [17] has a lots of artifacts and the human body is not properly formed. The VDNet variant improves the quality with no visible artifacts, but still the human body is not well formed. Also, there is no visible motion as we move from frame 1 to 8. The VDNet model use the sequence of depth/skeleton from the target view and still the motion is not quite visible in the synthesized video frames. In our approach, the background is of high quality, which is due to the prior, and we can also observe noticeable motion along the generated frames of the video. We also evaluate the variation in SSIM score with varying number of predicted frames in the video. The evaluation is shown in Figure 4 and we can observe that the video quality using proposed method is consistent with increasing number of frames and it outperforms existing approaches.

4.3 Ablation study

We perform some ablation experiments to study the impact of various components in the proposed model. We experimented with four different variations. The first variation, basic model, does not include action transformation, hierarchical transformation, and appearance transformation. In the other three variation, we add these three components incrementally. We observe that each of these com-

Table 2. Ablation experiments to study the impact of various components of the network on video synthesis. AC-Trans: action transformation, HI-Trans: hierarchical transformation, and AP-Trans: appearance transformation

Model	Pair-view PSNR Score						Average
	$v_1 \rightarrow v_2$	$v_1 \rightarrow v_3$	$v_2 \rightarrow v_1$	$v_2 \rightarrow v_3$	$v_3 \rightarrow v_1$	$v_3 \rightarrow v_2$	
basic model	23.8 ± 1.5	23.8 ± 1.4	24.2 ± 1.4	23.7 ± 1.5	24.2 ± 1.5	23.7 ± 1.6	23.9 ± 1.5
w/ AC-Trans	24.7 ± 1.5	24.7 ± 1.4	25.1 ± 1.6	24.5 ± 1.6	25.0 ± 1.6	24.4 ± 1.7	24.7 ± 1.6
w/ HI-Trans	26.7 ± 2.6	26.8 ± 2.5	26.8 ± 2.7	26.5 ± 2.7	26.8 ± 2.6	26.4 ± 2.7	26.7 ± 2.6
w/ AP-Trans	27.6 ± 2.7	27.7 ± 2.8	27.7 ± 2.7	27.2 ± 2.8	27.6 ± 2.7	27.2 ± 2.8	27.5 ± 2.7
	Pair-view SSIM Score						
	$v_1 \rightarrow v_2$	$v_1 \rightarrow v_3$	$v_2 \rightarrow v_1$	$v_2 \rightarrow v_3$	$v_3 \rightarrow v_1$	$v_3 \rightarrow v_2$	
basic model	$.939 \pm .033$	$.940 \pm .026$	$.943 \pm .026$	$.937 \pm .033$	$.943 \pm .026$	$.936 \pm .040$	$.940 \pm .031$
w/ AC-Trans	$.950 \pm .022$	$.951 \pm .021$	$.954 \pm .026$	$.948 \pm .024$	$.953 \pm .023$	$.947 \pm .033$	$.951 \pm .025$
w/ HI-Trans	$.967 \pm .028$	$.967 \pm .023$	$.967 \pm .023$	$.964 \pm .027$	$.967 \pm .023$	$.964 \pm .030$	$.966 \pm .026$
w/ AP-Trans	$.974 \pm .021$	$.975 \pm .021$	$.975 \pm .019$	$.971 \pm .021$	$.974 \pm .017$	$.971 \pm .022$	$.973 \pm .020$

ponents help in improving the synthesized video quality in terms of both PSNR and SSIM evaluation. A detailed analysis of these ablations is shown in Table 2.

In Figure 6, we show synthesized video frames using different variations in the model. We can observe that without any action and appearance transformation, the actor becomes blurry as the video progresses. The action transformation helps in improving the motion quality in the synthesized video. The hierarchical transformation improves the quality further and the appearance transformation helps in improving the visual quality of the synthesized video.

4.4 Novel view with novel actor

The proposed network takes an appearance prior from the target view-point. This allows us to potentially impose the action of the video from any source view-point onto another person and another location from a novel view-point. We have shown some examples of synthesized video frames in Figure 7 where the prior from the novel view is from a different actor and location. We observe that the proposed approach is able to synthesize the video with the correct appearance from the prior frame and the correct action from the source video. Thus, we know that each branch of our model (the motion branch and the appearance branch) is learning what it is supposed to learn; each only contributes information about motion or appearance appropriately.

4.5 Limitations and failure cases

The proposed approach is able to successfully transform the performed action to a novel view at a coarse level. The main limitation of the current approach is that the finer appearance details are missing in the synthesized video leading to a motion blur. It is important to note that the training is performed at a smaller resolution (112x112) to avoid a higher memory consumption and long training duration due to resource constraints. At this resolution, it is challenging

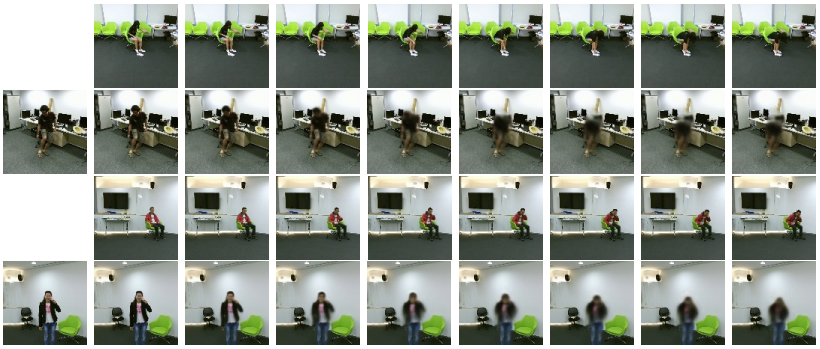

Fig. 7. Novel view and novel actor: The synthesized video frames from a novel view with a different actor and different background. For each sample the top row shows 8 frames of the source video and the bottom row shows a prior from another view followed by synthesized video frames for the novel view and novel actor. We can observe that the motion was successfully transformed to the novel actor (with different initial pose) in the novel viewpoint. For each of these, frames 1, 3, 5, 7, 9, 11, 13, and 15 are shown

to preserve the fine appearance and motion details of the actor in the latent representation space. Also, synthesizing motion from a novel view is much more challenging. Even with the help of motion prior from the target viewpoint, the authors in [17] were not very successful in synthesizing a high-quality action from novel view-points. The availability of resources (multiple GPU's or GPU's with higher memory) will definitely help in improving this quality, but, capturing the fine appearance and motion details in memory constraint environment is a challenging yet interesting problem, which can be explored in the future work.

5 Conclusion

In this work, we address the problem of novel view video synthesis by transforming the source action to target novel view in latent space. We propose a recurrent structure which utilize these action features and transform the prior from target view for video synthesis. The model predicts action key-points in an unsupervised way and enables the appearance transformer and video generator to focus on action regions. We evaluated the effectiveness of the proposed method on the largest multi-view action dataset. The experimental results demonstrate the effectiveness of the proposed framework in cross-view action synthesis even with varying actor and background scenes.

Acknowledgement Kara Marie Schatz contributed to this work while she was an NSF REU student in Center for Research in Computer Vision at University of Central Florida which was supported under NSF CNS grant #1461121.

References

1. Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.
2. Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018.
3. Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, Petros Koumoutsakos, PR Vlachas, ZY Wan, TP Sapsis, F Raue, S Palacio, TM Breuel, et al. Contextvp: Fully context-aware video prediction. In *Proceedings of the IEEE CVPR Workshops*, 2018.
4. Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.
5. Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
6. Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019.
7. Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.
8. Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.
9. SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 2018.
10. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
11. Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010.
12. 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)*, pages 0–0, 2018.
13. Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems*, pages 4016–4027, 2018.
14. Dinesh Jayaraman, Ruohan Gao, and Kristen Grauman. Shapecodes: self-supervised feature learning by lifting views to viewgrids. In *European conference on computer vision*, 2018.
15. Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
16. Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

17. Mohamed Ilyes Lakhal, Oswald Lanz, and Andrea Cavallaro. View-lstm: Novel-view video synthesis through view decomposition. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
18. Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE conference on CVPR*, 2017.
19. Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In *Advances in Neural Information Processing Systems*, 2018.
20. Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1744–1752, 2017.
21. Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.
22. Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2016.
23. Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *IEEE Conference on CVPR*, 2018.
24. Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
25. Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019.
26. 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 CVPR*, 2016.
27. Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.
28. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
29. Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.
30. Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *arXiv preprint arXiv:1707.04993*, 2017.
31. Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016.
32. Shruti Vyas, Yogesh S Rawat, and Mubarak Shah. Time-aware and view-aware video rendering for unsupervised representation learning. *arXiv preprint arXiv:1811.10699*, 2018.
33. Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3332–3341, 2017.
34. Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems*, pages 1144–1156, 2018.

35. Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5110–5119, 2018.
36. Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. *In International Conference on Learning Representations (ICLR)*, 2019.
37. Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems*, pages 879–888, 2017.
38. Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9154–9162, 2019.
39. Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
40. Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.