A Recurrent Transformer Network for Novel View Action Synthesis (Supplementary)

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}@ucf.edu

In this document, we provide the architecture details for all the components of the proposed network. It includes video encoder f_E (Table 1), prior encoder g_E (Table 2), change in view-point detector h (Table 4), action transformer t_M (Table 5), action key-point detector k_G (Table 6), appearance transformer t_A (Table 7), and action video generator f_G (Table 8). We also show some more qualitative results for cross-view video synthesis (Figure 2 and 1), action key-points (Figure 3), and cross-view video synthesis for different actor and scene (Figure 4). Apart from this, the supplementary material also contains a demo video for cross-view video synthesis.

Table 1. Network details for the Visual Encoder, f_E, which is used to encode the input video V^i into motion features. Note that the input column contains either the tensor used as input to the particular layer or the layer whose output is used as input. Also, note that the Inception module is that from [1]. Since the proposed method involves hierarchical transformation, there are three outputs from this network: VidE-ReLU5a, VidE-ReLU5b, and VidE-ReLU5c. The bottom section of the table indicates three extra convolutional layers, one for each output, used to force a uniform number of channels for all three encodings.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
VidE-Conv1 (skip)	3D Conv	V^i	7 × 7 × 7	2 × 2 × 2	8 × 56 × 56 × 64
VidE-MaxPool1	3D Max Pool	VidE-Conv1	1 × 3 × 3	1 × 2 × 2	8 × 28 × 28 × 64
VidE-Conv2	3D Conv	VidE-MaxPool1	1 × 1 × 1	1 × 1 × 1	8 × 28 × 28 × 64
VidE-Conv3 (skip)	3D Conv	VidE-Conv2	3 × 3 × 3	1 × 1 × 1	8 × 28 × 28 × 192
VidE-MaxPool2	3D Max Pool	VidE-Conv3	1 × 3 × 3	1 × 2 × 2	4 × 14 × 14 × 192
VidE-IncModule1	Inception Module	VidE-MaxPool2	-	-	4 × 14 × 14 × 256
VidE-IncModule2	Inception Module	VidE-IncModule1	-	-	4 × 14 × 14 × 480
VidE-MaxPool3	3D Max Pool	VidE-IncModule2	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 480
VidE-IncModule3	Inception Module	VidE-MaxPool3	-	-	4 × 14 × 14 × 592
VidE-IncModule4	Inception Module	VidE-IncModule3	-	-	4 × 14 × 14 × 512
VidE-IncModule5	Inception Module	VidE-IncModule4	-	-	4 × 14 × 14 × 512
VidE-IncModule6	Inception Module	VidE-IncModule5	-	-	4 × 14 × 14 × 528
VidE-IncModule7	Inception Module	VidE-IncModule6	-	-	4 × 14 × 14 × 832
VidE-IncModule8	Inception Module	VidE-IncModule7	-	-	4 × 14 × 14 × 832
VidE-IncModule9	Inception Module	VidE-IncModule8	-	-	4 × 14 × 14 × 1024
VidE-Conv4	3D Conv	VidE-IncModule9	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 256
VidE-Conv5a	3D Conv	VidE-Conv1	3 × 3 × 3	1 × 1 × 1	8 × 14 × 14 × 128
VidE-ReLU5a	ReLU	VidE-Conv5a	-	-	8 × 14 × 14 × 128
VidE-Conv5b	3D Conv	VidE-Conv3	3 × 3 × 3	1 × 1 × 1	8 × 28 × 28 × 128
VidE-ReLU5b	ReLU	VidE-Conv5b	-	-	8 × 28 × 28 × 128
VidE-Conv5c	3D Conv	VidE-Conv4	3 × 3 × 3	1 × 1 × 1	4 × 56 × 56 × 128
VidE-ReLU5c	ReLU	VidE-Conv5c	-	-	4 × 56 × 56 × 128
Total Parameters:	19,365,664				

Table 2. Network details for the Visual Encoder, g_E, which was based upon [2]. The above table contains an enumeration of all layers and operations used to encode the input frame P^j into an appearance feature map. Note that the input column contains either the tensor used as input to the particular layer or the layer whose output is used as input. Since the proposed method involves hierarchical transformation, there are three outputs from this network: VisE-ReLU4a, VisE-ReLU4b, and VisE-ReLU4c. The bottom section of the table indicates three extra convolutional layers, one for each output, used to force a uniform number of channels for all three encodings.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
VisE-Conv1a	2D Conv	P^j	3 × 3	1 × 1	112 × 112 × 64
VisE-ReLU1a	ReLU	VisE-Conv1a	-	-	112 × 112 × 64
VisE-Conv1b	2D Conv	VisE-ReLU1a	3 × 3	1 × 1	112 × 112 × 64
VisE-ReLU1b	ReLU	VisE-Conv1b	-	-	112 × 112 × 64
VisE-MaxPool1	2D Max Pool	VisE-ReLU1b	2 × 2	2 × 2	56 × 56 × 64
VisE-Conv2a	2D Conv	VisE-MaxPool1	3 × 3	1 × 1	56 × 56 × 128
VisE-ReLU2a	ReLU	VisE-Conv2a	-	-	56 × 56 × 128
VisE-Conv2b	2D Conv	VisE-ReLU2a	3 × 3	1 × 1	56 × 56 × 128
VisE-ReLU2b	ReLU	VisE-Conv2b	-	-	56 × 56 × 128
VisE-MaxPool2	2D Max Pool	VisE-ReLU2b	2 × 2	2 × 2	28 × 28 × 128
VisE-Conv3a	2D Conv	VisE-MaxPool2	3 × 3	1 × 1	28 × 28 × 256
VisE-ReLU3a	ReLU	VisE-Conv3a	-	-	28 × 28 × 256
VisE-Conv3b	2D Conv	VisE-ReLU3b	3 × 3	1 × 1	28 × 28 × 256
VisE-ReLU3b	ReLU	VisE-Conv3b	-	-	28 × 28 × 256
VisE-Conv3c	2D Conv	VisE-ReLU3b	3 × 3	1 × 1	28 × 28 × 256
VisE-ReLU3c	ReLU	VisE-Conv3c	-	-	28 × 28 × 256
VisE-MaxPool3	2D Max Pool	VisE-ReLU3c	2 × 2	2 × 2	14 × 14 × 256
VisE-Conv4a	2D Conv	VisE-ReLU2b	3 × 3	1 × 1	14 × 14 × 128
VisE-ReLU4a	ReLU	VisE-Conv4a	-	-	14 × 14 × 128
VisE-Conv4b	2D Conv	VisE-ReLU3c	3 × 3	1 × 1	28 × 28 × 128
VisE-ReLU4b	ReLU	VisE-Conv4b	-	-	28 × 28 × 128
VisE-Conv4c	2D Conv	VisE-MaxPool3	3 × 3	1 × 1	56 × 56 × 128
VisE-ReLU4c	ReLU	VisE-Conv4c	-	-	56 × 56 × 128
Total Parameters:	1,735,488				

Table 3. Network details for the Viewpoint Expander, which repeats the angular change in viewpoint θ_{ij} for every spatio-temporal location in the motion features extracted from f_E and then performs convolutions. This allows θ_{ij} to be concatenated with the outputs from f_E so that both can serve as input to the action transformation network t_M.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
E-Expand	-	θ_{ij}	-	-	4 × 14 × 14 × 1
E-Conv1	3D Conv	E-Expand	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 4
E-Conv2	3D Conv	E-Conv1	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 1
Total Parameters:	221				

Table 4. Network details for the Viewpoint Change Predictor, h, which uses the visual encoding of the novel viewpoint and the action embedding of the source viewpoint to predict the angular change in viewpoint $\hat{\theta_{ij}}$.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
VCP-AvgPool1	3D Avg Pool	VidE-ReLU5a	4 × 1 × 1	4 × 1 × 1	14 × 14 × 128
VCP-Conv1	2D Conv	VisE-ReLU4a + VCP-AvgPool1	3 × 3	1 × 1	14 × 14 × 128
VCP-ReLU1	ReLU	VCP-Conv1	-	-	14 × 14 × 128
VCP-AvgPool2	2D Avg Pool	VCP-ReLU1	2 × 2	2 × 2	7 × 7 × 128
VCP-Conv2	2D Conv	VCP-AvgPool2	3 × 3	1 × 1	7 × 7 × 32
VCP-ReLU2	ReLU	VCP-Conv2	-	-	7 × 7 × 32
VCP-AvgPool3	2D Avg Pool	VCP-ReLU2	2 × 2	2 × 2	3 × 3 × 32
VCP-Reshape	Reshape	VCP-AvgPool3	-	-	288
VCP-FC	Linear	VCP-Reshape	288	-	1
Total Parameters:	627,137				

Table 5. Network details for the Action Transformer, t_M, which is used to transform the motion features from the Video Encoder f_E according to the angular viewpoint change $\hat{\theta ij}$ or θ_{ij} for training. Note that each layer has three inputs and three outputs due to the inclusion of hierarchical transformation. All three outputs from f_E are transformed in this manner and later used to transform the appearance at three different levels.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
ActT-Conv1	3D Conv	VCP-FC + VidE-ReLU5a	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 257
		VCP-FC + VidE-ReLU5b			8 × 28 × 28 × 257
		VCP-FC + VidE-ReLU5c			8 × 56 × 56 × 257
ActT-Conv2	3D Conv	ActT-Conv1(1)	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 128
		ActT-Conv1(2)			8 × 28 × 28 × 128
		ActT-Conv1(3)			8 × 56 × 56 × 128
ActT-Conv3	3D Conv	ActT-Conv2(1)	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 128
		ActT-Conv2(2)			8 × 28 × 28 × 128
		ActT-Conv2(3)			8 × 56 × 56 × 128
Total Parameters:	5,329,948				

Table 6. Network details for the action Key-point Detector, k_G, which extracts 32 action key-points as Gaussian heatmaps from the transformed action embeddings. Note that hierarchical key-point detection is performed as one of the three embeddings of transformed action features is concatenated as input where appropriate. The key-point centers are determined by the first eleven layers. Then, the final layer, Gaussian, turns those centers into Gaussian heatmaps.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
KPD-Conv1	3D Conv	ActT-Conv3(1)	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 128
KPD-ReLU1	ReLU	KPD-Conv1	-	-	4 × 14 × 14 × 128
KPD-Inter1	Interpolate	KPD-ReLU1	-	-	8 × 28 × 28 × 128
KPD-Conv2	3D Conv	KPD-Inter1 + ActT-Conv3(2)	3 × 3 × 3	1 × 1 × 1	8 × 28 × 28 × 128
KPD-ReLU2	ReLU	KPD-Conv2	-	-	8 × 28 × 28 × 128
KPD-Inter2	Interpolate	KPD-ReLU2	-	-	8 × 56 × 56 × 128
KPD-Conv3	3D Conv	KPD-Inter2 + ActT-Conv3(3)	3 × 3 × 3	1 × 1 × 1	8 × 56 × 56 × 128
KPD-ReLU3	ReLU	KPD-Conv3	-	-	8 × 56 × 56 × 128
KPD-Inter3	Interpolate	KPD-ReLU3	-	-	16 × 56 × 56 × 128
KPD-Conv4	3D Conv	KPD-Inter3	3 × 3 × 3	1 × 1 × 1	16 × 56 × 56 × 32
KPD-Sig	Sigmoid	KPD-Conv4	-	-	16 × 56 × 56 × 32
KPD-Gaussian	Gaussian	KPD-Sig	-	-	16 × 56 × 56 × 32
Total Parameters:	2,322,848				

Table 7. Network details for the Appearance Transformer, t_A, which is used to transform the appearance embeddings from the Visual Encoder g_E according to the transformed action features and the predicted action key-points. Note that t_A is a recurrent network and only one of the recurrent cells is detailed above. This cell would be repeated T times, where T is the size of the temporal dimension of the transformed action features. Then, the T cell outputs are concatenated in the temporal dimension to produce a transformed appearance of the same size as the transformed action features. Also, note that hierarchical transformation is used, so the recurrent network is used three times, once for each of the appearance embeddings.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
AppT-Conv1	2D Conv	ActT-Conv3(1) + KPD-Gaussian + VisE-ReLU4a	7 × 7	1 × 1	14 × 14 × 256
AppT-Split	Split	AppT-Conv1	-	-	14 × 14 × 128 14 × 14 × 128
AppT-Sig1	Sigmoid	AppT-Split(1)	-	-	14 × 14 × 128
AppT-Sig2	Sigmoid	AppT-Split(2)	-	-	14 × 14 × 128
AppT-Conv2	2D Conv	ActT-Conv3(1) + KPD-Gaussian + AppT-Sig1 * VisE-ReLU4a	7 × 7	1 × 1	14 × 14 × 128
AppT-Tanh	Tanh	AppT-Conv2	-	-	14 × 14 × 128
AppT-Final	Concat	(1 - AppT-Sig2) * VisE-ReLU4a + AppT-Sig2 * AppT-Tanh	-	-	14 × 14 × 128
Total Parameters:	5,419,008				

Table 8. Network details for the Action Generator, f_G, which generates the final output video $\hat{V}^j$ based upon the three sets of transformed appearance features and the predicted action key-points. Note that hierarchical generation is used, so the larger appearance features are concatenated as input where appropriate. The final output has the same dimensions as the input video V^i.

Name	Layer	Input	Kernel Dims (T × H × W)	Strides (T × H × W)	Output Dims (T × H × W × C)
AG-Conv1a	3D Conv	AppT-Final(1)			
	+ KPD-Gaussian	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 128	
AG-ReLU1a	ReLU	AG-Conv1a	-	-	4 × 14 × 14 × 128
AG-Conv1b	3D Conv	AG-ReLU1a	3 × 3 × 3	1 × 1 × 1	4 × 14 × 14 × 128
AG-ReLU1b	ReLU	AG-Conv1b	-	-	4 × 14 × 14 × 128
AG-Inter1	Interpolate	AG-ReLU1b	-	-	8 × 28 × 28 × 128
AG-Conv2a	3D Conv	AppT-Final(2)			
		+ KPD-Gaussian			
		+ AG-Inter1	3 × 3 × 3	1 × 1 × 1	8 × 28 × 28 × 128
AG-ReLU2a	ReLU	AG-Conv2a	-	-	8 × 28 × 28 × 128
AG-Conv2b	3D Conv	AG-ReLU2a	3 × 3 × 3	1 × 1 × 1	8 × 28 × 28 × 64
AG-ReLU2b	ReLU	AG-Conv2b	-	-	8 × 28 × 28 × 64
AG-Inter2	Interpolate	AG-ReLU2b	-	-	8 × 56 × 56 × 64
AG-Conv3a	3D Conv	AppT-Final(3)			
		+ KPD-Gaussian			
		+ AG-Inter2	3 × 3 × 3	1 × 1 × 1	8 × 56 × 56 × 128
AG-ReLU3a	ReLU	AG-Conv3a	-	-	8 × 56 × 56 × 128
AG-Conv3b	3D Conv	AG-ReLU3a	3 × 3 × 3	1 × 1 × 1	8 × 56 × 56 × 32
AG-ReLU3b	ReLU	AG-Conv3b	-	-	8 × 56 × 56 × 32
AG-Inter3	Interpolate	AG-ReLU3b	-	-	16 × 112 × 112 × 32
AG-Conv4a	3D Conv	AG-Inter3	3 × 3 × 3	1 × 1 × 1	16 × 112 × 112 × 8
AG-ReLU4a	ReLU	AG-Conv4a	-	-	16 × 112 × 112 × 8
AG-Conv4b	3D Conv	AG-ReLU4a	1 × 1 × 1	1 × 1 × 1	16 × 112 × 112 × 3
AG-Sig	Sigmoid	AG-Conv4b	-	-	16 × 112 × 112 × 3
Total Parameters:	3,104,131				

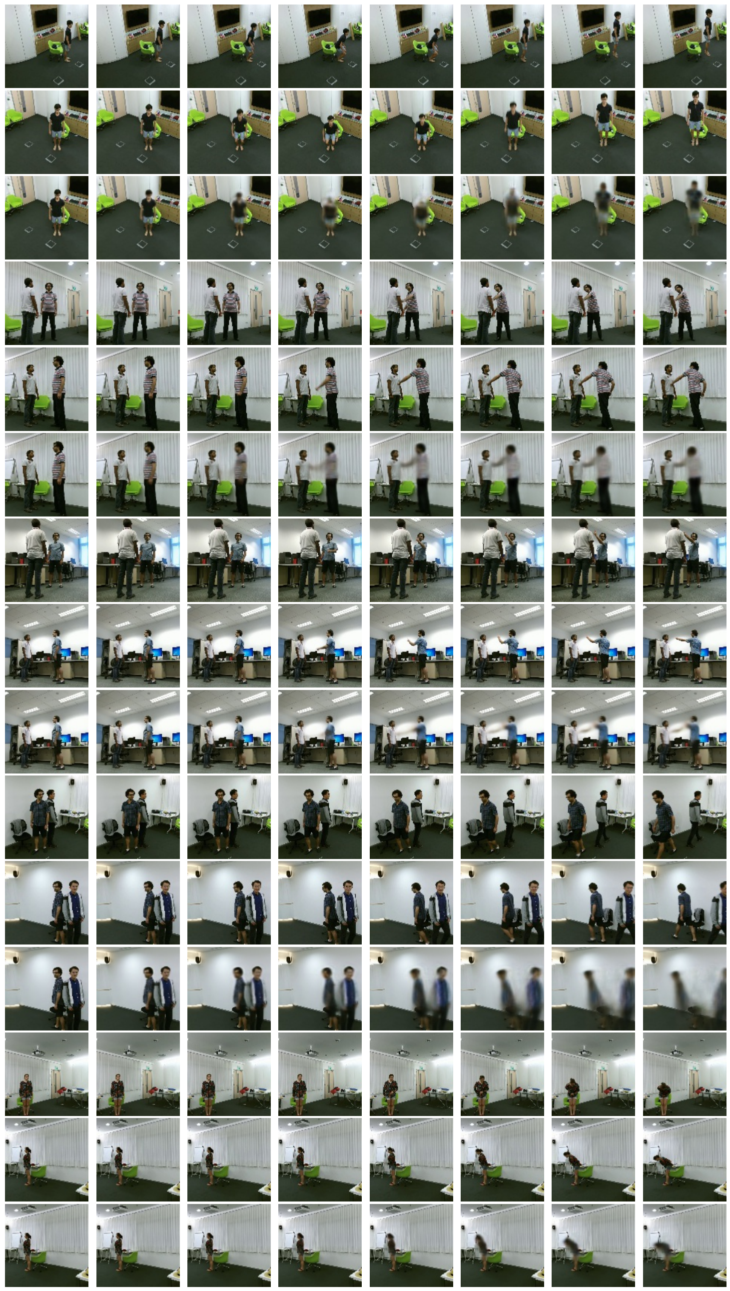

Fig. 1. Video frames for cross-view synthesis. Source video frames (row 1,4,7,10), ground truth target video frames (row 2,5,8,11), and generated video frames from target view (row 3,6,9,12)

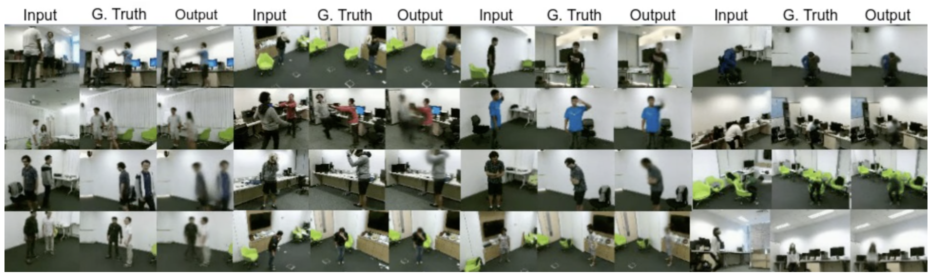

Fig. 2. Sample generated video frames for multiple example cases along with corresponding input view and target view frames from the demo video

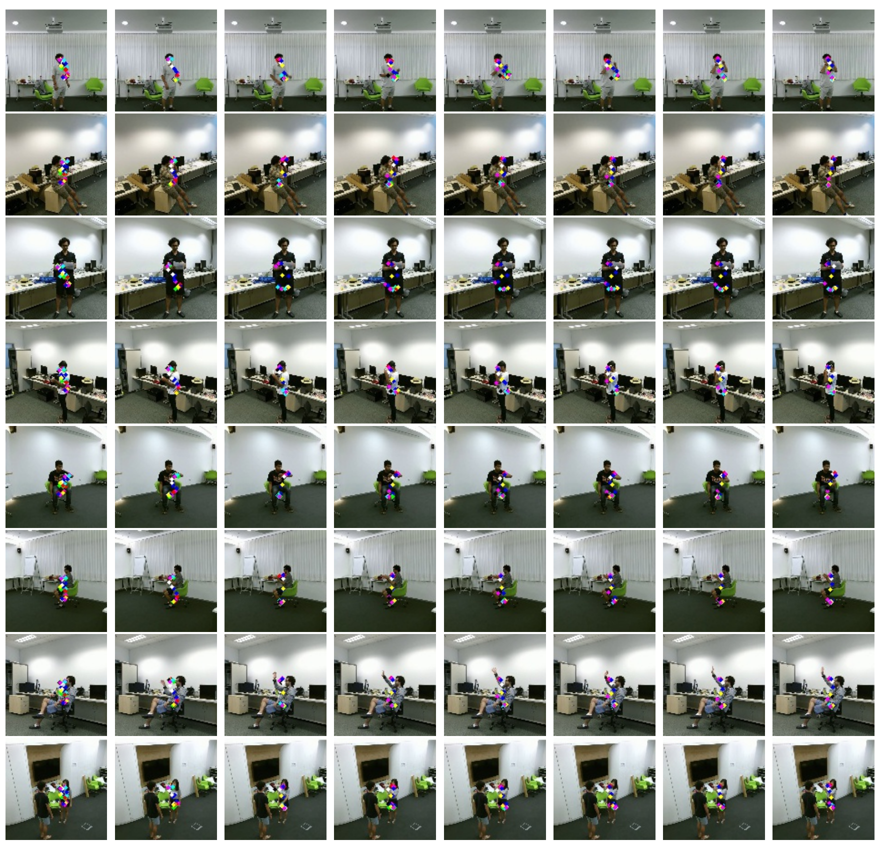

Fig. 3. Overlay of the center of the detected action key-points on the video frames from target view-point.

Fig. 4. Novel view with novel actor: qualitative results for cross-view video synthesis where the actor and the scene is different from the source action video. Source action video (row 1, 3, 5, 7, 9), and synthesized action video from target view-point (row 2, 4, 6, 8, 10)

References

1. Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2
2. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2