Video Generation and Synthesis Network for Long-term Video Interpolation

Nayoung Kim*†, Jung Kyung Lee*, Chae Hwa Yoo*, Seunghyun Cho‡, and Je-Won Kang*†
* Department of Electronic and Electrical Engineering, Ewha W. University, Seoul, Korea
‡Realistic AV Research Group, Electronics and Telecommunications Research Institute
E-mail: †12skdud21g@ehwain.net, ‡shcho@etri.re.kr, †jewonk@ewha.ac.kr

***Abstract*—In this paper, we propose a bidirectional synthesis video interpolation technique based on deep learning, using a forward and a backward video generation network and a synthesis network. The forward generation network first extrapolates a video sequence, given the past video frames, and then the backward generation network generates the same video sequence, given the future video frames. Next, a synthesis network fuses the results of the two generation networks to create an intermediate video sequence. To jointly train the video generation and synthesis networks, we define a cost function to approximate the visual quality and the motion of the interpolated video as close as possible to those of the original video. Experimental results show that the proposed technique outperforms the state-of-the art long-term video interpolation model based on deep learning.**

I. Introduction

Long-term video interpolation technique generates a number of intermediate video frames between the past and the future. It has been widely used for video frame rate-up conversion, video restoration, and video coding applications, etc. Spatio-temporal correlation in successive video frames may allow visually realistic quality of the generated frames. However, the estimation is often challenging because of fast motion changes and occlusions in natural videos.

The conventional video interpolation techniques go through two separated procedures, *i.e.*, the motion prediction and pixel synthesis [3], [4]. Optical flow is the common approach to find motion vectors, but it can predict an accurate motion vector only in temporally adjacent frames. More recently, there have been active studies in a long-term video interpolation using the advanced deep learning models. EpicFlow[1] uses Convolutional Neural Networks (CNNs) to estimate the optical flow to generate the intermediate frames. The model detects edges in an image, and then predict the corresponding points to which a pixel moves. The model requires the groundtruth of optical flows for supervised learning. Deep Voxel Flow (DVF)[2] is proposesd to estimate optical flows from unsupervised learning, so it does not need any labeling in the training. However, abrupt motion changes can easily degrade the performance as the DVF models a linear motion. Adaptive Separable Convolution (AdapSC) [10] learns four convolutional filters from the past and the future frames to estimate the current picture. The kernels are used for convolving pixels to the horizontal and the vertical directions. The AdapSC [10] shows visually pleasing results in a short-term interpolation, yet the convolution filters cannot efficiently reflect the temporal changes.

Most of the previous works are challenged from gradually losing temporal correlation between-in consecutive frames in the long-term interpolation. To tackle this problem, works of prior research uses Long-Short Term Memory (LSTM) based on Recurrent Neural Network (RNN) to maintain the spatio-temporal correlation in the estimated frames. Especially, the convolutional LSTM (ConvLSTM) [8] is efficiently used for preserving the spatio-temporal information though the parameter training is not easy. Kim *et al.* propose to evolve residual frames with ConvLSTM to facilitate the training [11].

In this paper, we propose a bidirectional synthesis video interpolation technique based on deep learning, using a forward and a backward video generation network and a synthesis network. The forward generation network first extrapolates a video sequence, given the past video frames, and then the backward generation network generates the same video sequence, given the future video frames. Next, a synthesis network fuses the results of the two generation networks to create an intermediate video sequence. To jointly train the video generation and synthesis networks, we define a cost function to approximate the visual quality and the motion of the interpolated video as close as possible to those of the original video.

The rest of the paper is organized as follows. In Sec.II, we describe the proposed algorithm. In Sec.III, we present experimental results and analysis. In Sec. IV, we remark the conclusion and the future work.

II. The proposed algorithm

A. Problem Formulation

We define a video sequence as $X = [x_1, x_2, \ldots, x_{2n+m}]$. The proposed technique produces an intermediate video sequence $X_o = [x_{n+1}, x_{n+2}, \ldots, x_{n+m}]$ as an output, given a set of n consecutive video frames in the past, denoted by X_f, and the other set of n frames in the future, denoted by X_b. We use a forward generation when estimating X_o from X_f and a backward generation when estimating $X_o^r = [x_{n+m}, \ldots, x_{n+1}]$ from X_b, as shown in Fig. 1. Given the sets, the proposed technique aims to predict the output video by performing the forward and the backward generation and synthesizing the prediction results.

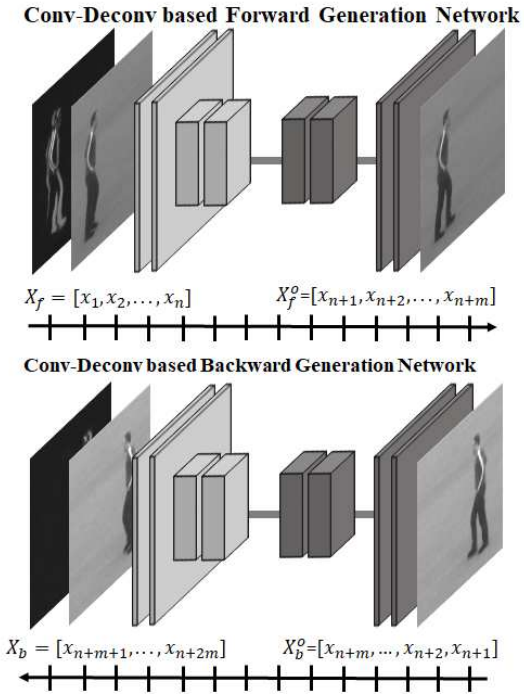

Fig. 1. Video generation process with the forward video generation network and the backward video generation network, using the long-term video extrapolation [11]

B. Unidirectional Video Generation

We have developed a uni-directional long-term video extrapolation technique in [11]. In our work, two streams of the convolutional neural networks (CNN) are employed to encode the spatial and the temporal dynamics of a video. The spatial layout of the input video is trained with the original video, *e.g.* $X = [x_{t_1}, x_{t_2}, ..., x_{t_n}]$ by passing through the convolution and de-convolution model architecture. The motion information is evolved with the residual video $R = [x_{t_2} - x_{t_1}, x_{t_3} - x_{t_2}, .., x_{t_n} - x_{t_{n-1}}] = [r_{t_1}, r_{t_2}, ..., r_{t_{n-1}}]$ by passing through the ConvLSTM model architecture. The model produces the feature vectors of the next residual frames in order to generates the output video sequence.

The forward and the backward generation in this work are conducted by using the video extrapolation technique, motivated by the prior research. The forward and the backward video generation networks play roles in performing the uni-directional generations with the input video sets and their residual video sets, as shown in Fig. 1. Given the input video sets, the forward generation network gives $\hat{X}_o^f$ as an output. Similarly, the backward generation network gives $\hat{X}_o^b$, including the same video frames though the temporal order is reversed. Note the two networks have the same structures to avoid any complicated training. The authors may refer the work [11] for more detailed mechanism of the unidirectional video generation.

C. Bidirectional Video Synthesis

$\hat{X}_o^f(k)$ and $\hat{X}_o^b(k)$ denote the k-th frames in the generated videos. We synthesize the frames into an output video frame $\hat{X}_o(k)$, as shown in Fig. 2. $S(k)$ is the kernel of the k-th feature map in the synthesis network, used for convolving $\hat{X}_o^f(k)$ and $\hat{X}_o^b(k)$. w is the weight parameter. The synthesized video is mathematically given by,

$$\hat{X}_o(k) = \{\hat{X}_o^f(k) \times w + \hat{X}_o^b(k) \times (1-w)\} * S(k), \quad (1)$$

where $k = n+1, n+2, \ldots, n+m$, and $*$ is the convolution operation. w is equal to $k + 1/m + 1$. The parameter is used to consider the decreasing visual quality as the generated frame becomes more distant from the input frame during the extrapolation. In other words, w compensates the degraded quality with the temporal distance in the synthesis, so it can help improve the quality rather than a simple average.

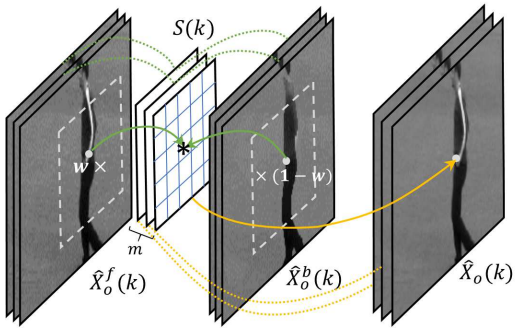

Fig. 2. Description of Bidirectional Video Synthesis network.

D. Cost Function

We use the cost function J to approximate the generated frame into the original frame visually as close as possible, given as

$$J = \alpha J_{IMG} + \beta J_{GRD} + \gamma J_{VGG}, \quad (2)$$

where α, β and γ are the weight parameters. In Eq. (2), J_{IMG} plays a role in creating the spatial layout of the generated frame close to the original frame. J_{GDL} and J_{VGG} are regularized terms to create the generated frame perceptually similar to the original video. We set β to 1 and γ to 0.001. We explain more details of the equation as below.

It is known that the use of L_2 norm can result in blurring effects to the generated frame. The mean-squared-error (MSE) term tend to average pixels out in a frame. Thus, our model includes a L_1 function to avoid the blurring in J_{IMG} in the approximation, as follows:

$$J_{IMG} = \alpha_1|\hat{X}_o^f - X_o| + \alpha_2|(\hat{X}_o^b - X_o)| + \alpha_3|(\hat{X}_o - X_o)|, \quad (3)$$

where $\hat{X}_o^f$ and $\hat{X}_o^b$ denote the output frame by the forward and the backward generation, and $\hat{X}_o$ denotes the output frame by the synthesis. α_1, α_2, and α_3 are the weight parameters. It is noted that our training strategy involves the adaptive changes of the weight parameters in the training. In the early stage of the training, we set higher values of the weight parameters to α_1 and α_2 until the forward generation network and the backward generation network can thoroughly approximate the original frame. After stabilizing the training, the other parameters become higher to improve the performance of the synthesis network. Specifically, α_1, α_2, and α_3 are initially set to 0.45, 0.45, and 0.1. The parameters are changed later to 0.25, 0.25, and 0.5, respectively.

Next, J_{GRD} is to sharpen edges by tuning gradient components in an image and improve the structural similarity index (SSIM) [6]. It is mathematically defined as,

$$J_{GRD} = \sum_{t=0}^{m-1}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1}(|x_{t,i,j} - x_{t,i-1,j}| - |\hat{x}_{t,i,j} - \hat{x}_{t,i-1,j}|)^2 + (|x_{t,i,j-1} - x_{t,i,j}| - |\hat{x}_{t,i,j-1} - \hat{x}_{t,i,j}|)^2, \quad (4)$$

where $x_{t,i,j}$ and $\hat{x}_{t,i,j}$ are the pixel values in a (i, j) coordinate of the original frame and the generated frame, respectively.

The J_{VGG} plays a role in creating a frame perceptually similar to the original video. The loss is computed by the VGG16 network [7]. We extract the features in the network while excluding the fully connected layers to compute the cost, *i.e.*,

$$J_{VGG} = |VGG(\hat{X}_o) - VGG(X_o)|, \quad (5)$$

where a function $VGG()$ extracts the feature map from the input image.

III. Experimental Results

We use two video databases, *i.e.*, Sports 1M [5] dataset and UCF101 dataset [9]. The Sports 1M dataset consists of about one million YouTube videos with 487 sports labels, and the dataset is used for training the proposed network model. The UCF101 is used for testing in the evaluation. We use about $10,000$ video sequences for the evaluation. In the experiments, we set n to 5 and m to 10. In other words, we interpolate 10 intermediate frames by using the five past frames in the forward generation and the five future frames in the backward generation.

We first calculate the PSNR and SSIM on average to measure the quality of the generated videos, as shown in Fig. 3. Fig. 3 shows the PSNR and SSIM values in 10 intermediate frames presented by the time indices 1 to 10. We evaluate the visual quality of the proposed technique as compared to the state-of-the-art video generation technique, i.e., convLSTM [8]. As shown, the proposed technique outperforms the compared technique in PSNR more than 1.9 dB and in SSIM more than 0.11 on average. We use more than 100,000 videos

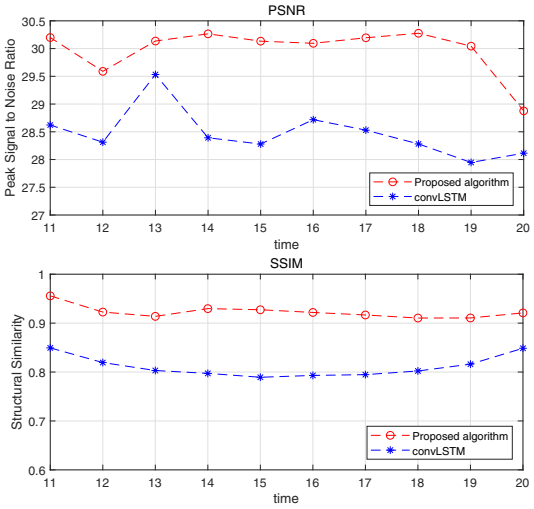

Fig. 3. Result of networks. Each row displays the generated frames of the ground truth, convLSTM [8], the proposed algorithm.

in the evaluation, and furthermore show the performance of the individual test videos in Table I. As shown, the proposed technique outperforms the compared algorithm in most of the test videos.

TABLE I
PSNR and SSIM on each sequence

Sequnce	The Proposed		ConvLSTM [8]	
	PSNR	SSIM	PSNR	SSIM
1	30.95	0.8869	29.61	0.8790
2	33.60	0.9816	28.18	0.8101
3	27.62	0.9447	25.38	0.9006
4	30.64	0.9000	29.12	0.9008
5	28.31	0.9231	28.41	0.8922
6	28.71	0.8853	27.41	0.8894
7	27.56	0.8831	26.09	0.8413
8	30.72	0.9522	28.31	0.8301
9	30.95	0.9260	29.94	0.9112
10	30.84	0.9337	30.63	0.9010
Avg.	30.04	0.9377	28.312	0.8554

We also show the visual quality of the generated videos in III. The first row of the videos include relatively large motions as the bat changes rapidly. In convLSTM, the motion of the bat is not quite clearly presented, but the proposed technique provides stable results in the generation. For the other videos, the proposed technique shows more texture information in the objects than the convLSTM in the slow motion.

IV. Conclusion

In this paper, we propose new long term video generation for video interpolation. We train the forward prediction network and backward prediction network. Then, we combine those frames using synthesis network. Our technique can achieve a PSNR value greater than convLSTM[8] by 1.7 dB , and SSIM greater than convLSTM[8] by 0.08 . Furthermore, our results are more perceptually pleasing for human vision.

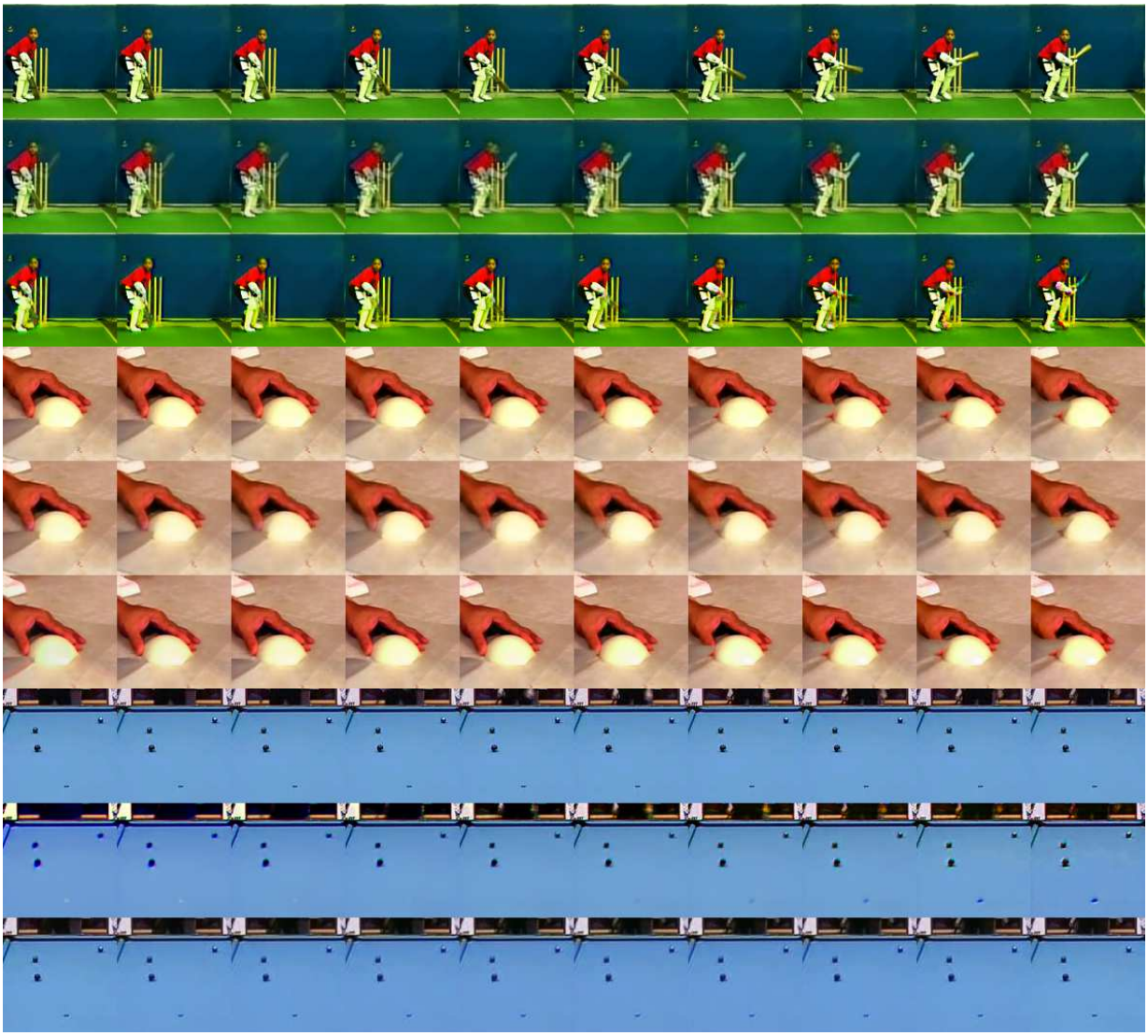

Fig. 4. Result of networks. Each row displays the generated frames of the ground truth, convLSTM [8], the proposed algorithm.

ACKNOWLEDGMENT

This work was supported by Institute for Information and communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (2017-0- 00072, Development of Audio/Video Coding and Light Field Media Fundamental Technologies for Ultra Realistic Tera-media and the Basic Science Research Program through the National Research Foundation of Korea (NRF)(NRF- 2016R1D1A1B03932994).)

REFERENCES

[1] J. Revaud, P. Weinzaepfel, , Z. Harchaoui, C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2015.

[2] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, A. Agarwala, "Video frame synthesis using deep voxel flow," *International Conference on Computer Vision (ICCV), Vol. 2,* 2017.

[3] J. Ascenso, C. Brites, F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services. Smolenice, Slovak Republic,* 2005.

[4] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *European conference on computer vision. Springer, Berlin, Heidelber,* 2004.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732)*, 2014.

[6] L. T. Wang, N. E. Hoover, E. H. Porter, J. J. Zasio, "SSIM: a software levelized compiled-code simulator," *In Proceedings of the 24th ACM/IEEE Design Automation Conference (pp. 2-8). ACM.*1987.

[7] S. Han,H. Mao, W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,"

arXiv preprint / arXiv:1510.00149, 2015.

[8] S. H. I. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting.," *In Advances in neural information processing systems (pp. 802-810),* 2015.

[9] K. Soomro, A. R. Zamir, M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402,* 2012.

[10] S. Niklaus, L. Mai, F. Liu, "Video frame interpolation via adaptive separable convolution," *arXiv preprint arXiv:1708.01692,* 2017.

[11] N. kim and J.-W. Kang, "Long-term Video Generation with Evolving Residual Video Frames"," *IEEE International Conference on Image Processing,* 2018.