Augmented Coarse-to-Fine Video Frame Synthesis with Semantic Loss

Xin Jin(✉), Zhibo Chen, Sen Liu, and Wei Zhou

CAS Key Laboratory of Technology in Geo-spatial Information Processing
and Application System, University of Science and Technology of China,
Hefei 230027, China
{jinxustc,weichou}@mail.ustc.edu.cn, chenzhibo@ustc.edu.cn,
elsen@iat.ustc.edu.cn

Abstract. Existing video frame synthesis works suffer from improving perceptual quality and preserving semantic representation ability. In this paper, we propose a Progressive Motion-texture Synthesis Network (PMSN) to address this problem. Instead of learning synthesis from scratch, we introduce augmented inputs to compensate texture details and motion information. Specifically, a coarse-to-fine guidance scheme with a well-designed semantic loss is presented to improve the capability of video frame synthesis. As shown in the experiments, our proposed PMSN promises excellent quantitative results, visual effects, and generalization ability compared with traditional solutions.

Keywords: Video frame synthesis · Augmented input
Coarse-to-fine guidance scheme · Semantic loss

1 Introduction

Video frame synthesis plays an important role in numerous applications of different fields, including video compression [2], video frame rate up-sampling [12], and pilot-less automobile [9]. Given a video sequence, video frame synthesis aims to interpolate frames between the existing video frames or extrapolate future video frames as shown in Fig. 1. However, constructing a generalized model to synthesize video frames is still challenging, especially for those videos with large motion and complex texture.

A lot of efforts have been dedicated towards video frame synthesis. Traditional approaches focused on synthesizing video frames from estimated motion information, such as optical flow [10,16,22]. Recent approaches have proposed deep generative models to directly hallucinate the pixel values of video frames [4,13,15,17,19,21,26,27]. However, these models always generate significant artifacts since the accuracy of motion estimation cannot be guaranteed. Meanwhile, due to the straightforward non-linear convolution operations, the results of deep generative models are suffered from blur artifacts.

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11256, pp. 439–452, 2018.
https://doi.org/10.1007/978-3-030-03398-9_38

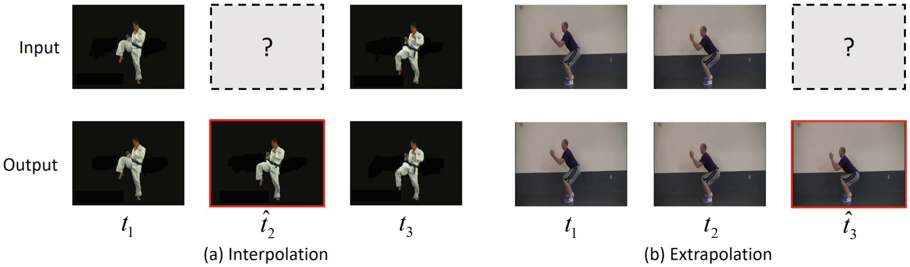


Fig. 1. Interpolation and extrapolation tasks in the video frame synthesis problem.

In order to tackle the above problems, we propose a deep model called Progressive Motion-texture Synthesis Network (PMSN), which is a global encoder-decoder architecture with coarse-to-fine guidance under a brain-inspired semantic objective. Overview of the whole process of PMSN is illustrated in Fig. 2. Specifically, we first introduce an augmented frames generation process to produce Motion-texture Augmented Frames (MAFs) containing coarse-grained motion prediction and high texture details. Second, in order to reduce the loss of detailed information in the feed-forward process and assist the network to learn motion tendency, MAFs are fed into the decoder stage with different scales in a coarse-to-fine manner, rather than the scheme of directly fusion into a single layer as described in [12]. Finally, we also adopt a brain-inspired semantic loss to further enhance the subjective quality and preserve the semantic representation ability of synthesized frames in the learning stage. The contributions of this paper are summarized as follows:

1. Instead of learning synthesis from scratch, we introduce a novel Progressive Motion-texture Synthesis Network (PMSN) to learn frame synthesis with triple-frame input under the assistant of augmented frames. These augmented frames provide effective prior information including motion tendency and texture details to compensate the video synthesis.
2. A coarse-to-fine guidance scheme is adopted in the decoder stage of the network to increase its sensitivity to informative features. Through this scheme, we can maximally exploit the informative features and suppress less useful ones at the same time, which acts as a bridge which combines conventional motion estimation methods and deep learning-based methods.
3. We develop a brain-inspired semantic loss for sharpening the synthetic results and strengthening object texture as well as motion information. The final results demonstrate better perceptual quality and semantic representation preserving ability.

2 Related Work

Traditional Methods. Early attempts at video frame synthesis focused on motion estimation based approaches. For example, Revaud *et al.* [22] proposed

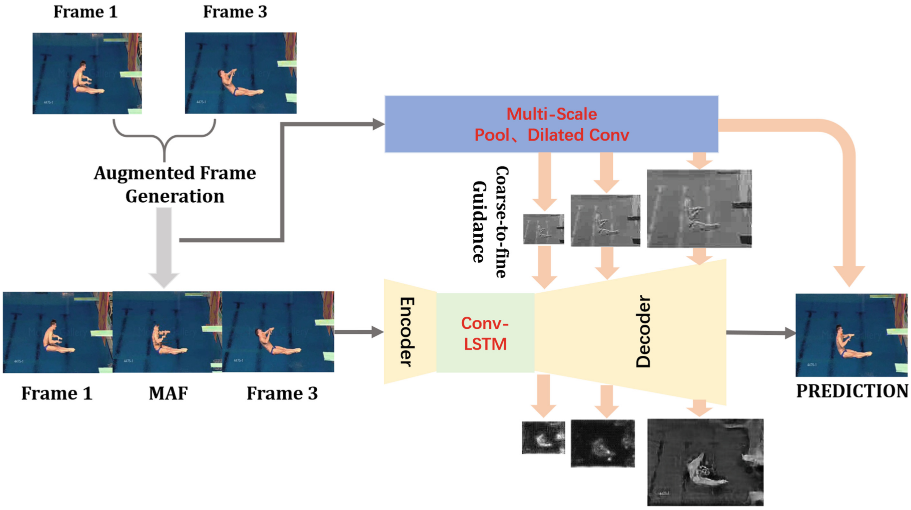


Fig. 2. Overview of our Progressive Motion-texture Synthesis Network (PMSN).

the EpicFlow to estimate optical flow by edge-aware distance. Li *et al.* [10] adopted a Laplacian Cotangent Mesh constraint to enhance the local smoothness for results generated by optical flow. Meyer *et al.* [16] leveraged the phase shift information for image interpolation. The results of these methods are highly relied on the precise estimation of motion information. Significant artifacts can be generated when unsatisfactory estimation happens for videos with large or complex motion.

Learning-Based Methods. The renaissance of deep neural network (DNN) remarkably accelerates the progress of video frame synthesis. Numbers of methods were proposed to interpolate or extrapolate video frames [13–15, 17, 19, 26, 27]. [17] focused on representing series transformation to predict small patches based on recurrent neural network (RNN). Xue *et al.* [27] proposed a model which generates videos with an assumption that the background is uniform. Lotter *et al.* [13] proposed a network called PredNet, which contains a series of stacked modules that forward the deviations in video sequences. Mathieu *et al.* [15] proposed a multi-scale architecture with adversarial training, which is referred as BeyondMSE. Niklaus *et al.* [19] tried to estimate a convolution kernel from the input frames. Then, the kernel was used to convolve patches from the input frames for synthesizing the interpolated ones. However, it is still hard to hallucinate realistic details for videos with complex spatiotemporal information only by the non-linear convolution operation.

Recently, Liu *et al.* [12] utilized the pixel-wise 3D voxel flow to synthesize video frames. Lu *et al.* [14] presented a Flexible Spatio-Temporal Network (FSTN) to capture complex spatio-temporal dependencies and motion patterns with diverse training strategies. [19] just focused on video frame interpolation

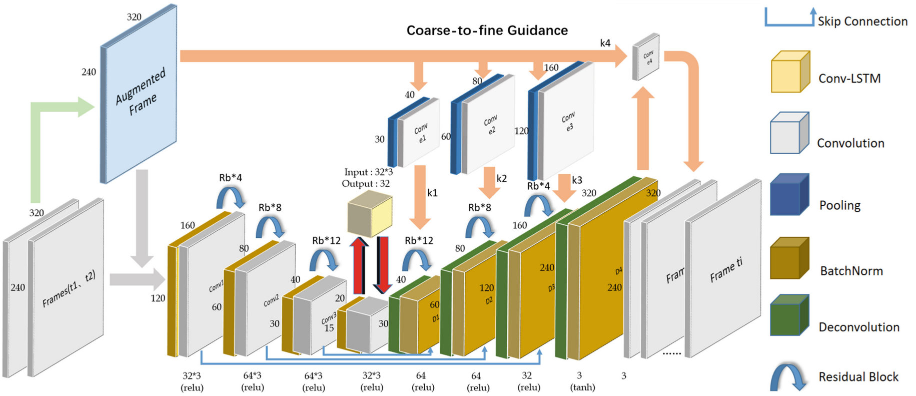


Fig. 3. The whole architecture of our Progressive Motion-texture Synthesis Network (PMSN).

task via adaptive convolution. Liang *et al.* [11] developed a dual motion Generative Adversarial Network (GAN) for video prediction. Villegas *et al.* [25] proposed a deep generative model named MCNet to extract the features of the last frame as content information and then encode the temporal differences between previous consecutive frames as motion information. Unfortunately, these methods usually only have the ability to deal with videos with tiny object motion and simple background which often cause blur artifacts in video scenes with large and complex motion. On the contrary, our proposed PMSN is able to achieve much better results, especially in complex scenes. In the experiment section Sect. 4, we will show adequate evaluations between our method and above methods.

3 Progressive Motion-Texture Synthesis Network

The whole architecture of our Progressive Motion-texture Synthesis Network (PMSN) is shown in Fig. 3, which takes advantage of the spatial invariance and temporal correlation for image representations. Instead of learning from scratch, the model receives original video frames combined with the produced augmented frames as the whole inputs. These triple-frame inputs provide more reference information for motion trajectory and texture residue, which leads to more reasonable high-level image representations. In the following sub-sections, we will first describe the augmented frames generation process. Then, the coarse-to-fine guidance scheme and semantic loss are presented.

Encoder Stage: Each convolutional block is shown in Fig. 4(a). The size of the receptive field for all convolution filters is (4, 4) along with stride (2, 2). A group of *ResidualBlocks* [5] (number of blocks in the group is shown in Fig. 3) is used to strengthen the non-linear representation and preserve more spatial-temporal details. To overcome the overfitting and internal covariant shift problems, we add a batch normalization layer before each Rectified Liner Unit (ReLU) layer [18].

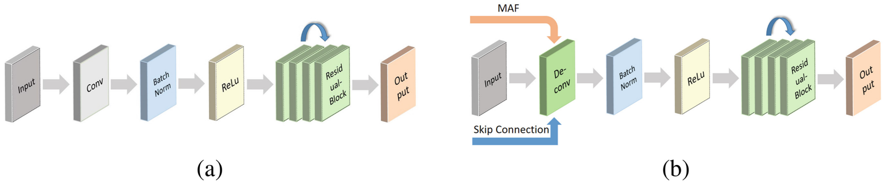


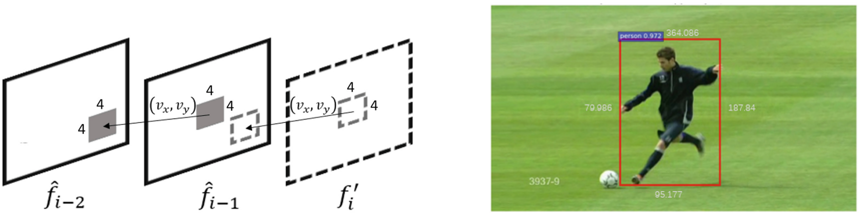
Fig. 4. Two sub-components in PMSN. (a) Convolutional block in PMSN. (b) Deconvolutional block in PMSN.

Decoder Stage: The deconvolutional block is used to upsample the feature maps, as demonstrated in Fig. 4(b), which has a receptive field of $(5,5)$ with stride $(2,2)$. The block also contains BatchNorm, ReLU layer and Residual Block, their parameters are shown in Fig. 3. To maintain the image details from low-level to high-level, we build skip connections, which are illustrated as the thin blue arrows in Fig. 3.

3.1 Augmented Frames Generation

Intrinsically, our PMSN utilizes augmented frames rather than learning from scratch. Then it is important for the augmented frame to preserve coarse motion trajectory and less-blurred texture, for PMSN to further improve the quality under the assistance of coarse-to-fine guidance and semantic loss. Therefore, any frame augmentation scheme satisfying above-mentioned two factors can be adopted in the PMSN framework, we introduce a simple augmented frames generation process in this paper to produce Motion-texture Augmented Frames (MAFs) containing coarse-grained motion prediction and high texture details. Similar to motion-estimation based frame synthesis methods, the original input frames are first decomposed into block-level matrixes. Then, we directly copy the matching blocks to MAFs according to the estimated motion vectors of these blocks. As shown in Fig. 5(a), to calculate the motion vector for generating MAF $f_i^{'}$, we first partition the frame $\hat{f}_{i-1}$ into regular 4×4 blocks, then search backward in the frame $\hat{f}_{i-2}$. When building each 4×4 block of MAF, the motion vectors of corresponding 4×4 block in frame $\hat{f}_{i-1}$ are utilized to locate and copy the data from frame $\hat{f}_{i-1}$. This block-sized thresholding 4×4 is sufficient for our purpose of generating the MAFs.

Note that this frame augmentation scheme can be replaced by any other frame synthesis solution, we verified this in the experiment section Sect. 4.3 by replacing MAFs with augmented frames generated from [19] and then demonstrate the effectiveness of our proposed PMSN.



(a) (b)

Fig. 5. (a) The generation precess of MAFs where the direction of motion vectors is backward. (b) Attention object bounding box extracted by faster R-CNN.

3.2 Coarse-to-Fine Guidance

In order to make use of the information aggregated in the MAFs as well as triple-frame input groups for selectively emphasizing informative features and suppressing less useful ones, we propose a coarse-to-fine guidance scheme to guide our network in an end-to-end manner, which is illustrated as orange arrows in Figs. 2, 3 and 4(b). Specifically, given the double-frame input X and single augmented frame $\tilde{Y}$, our goal is to obtain the synthesized interpolated/extrapolated frames Y', which can be formulated as:

$$Y' = f(G(X + \tilde{Y}), \tilde{Y}), \tag{1}$$

where G denotes a generator which learns motion trajectory and texture residue from triple-frame input group $X + \tilde{Y}$. Function f represents the fusion process to fully capture channel-wise dependencies through a concatenation operation.

In order to progressively improve the quality of synthesized frames in a coarse-to-fine way, we make a series of synthesis from MAFs with gradual increase resolutions, which is depicted as below:

$$\begin{aligned} Y_1' &= f(G_1(X + \tilde{Y}_1), e_1(\tilde{Y}_1)), \\ Y_2' &= f(G_2(Y_1' + \tilde{Y}_2), e_2(\tilde{Y}_2)), \\ &\dots \\ Y_k' &= f(G_k(Y_{k-1}' + \tilde{Y}_k), e_k(\tilde{Y}_k)), \end{aligned} \tag{2}$$

where k represents each level of the coarse-to-fine synthesis process. In our PMSN, we set the size of each level to 40×30 ($k = 1$), 80×60 ($k = 2$), 160×120 ($k = 3$), 320×240 ($k = 4$). G_k is the middle layer of G, and $G_1, G_2, ..., G_k$ compose an integrated network. And e_k is the feature extractor of $\tilde{Y}_k$, we employ two dilated convolutional layers [28] instead of simple downsample operations to preserve the texture details of original images. Since the output Y_k' is produced by a summation through all channels (X and Y), the channel dependencies are implicitly embedded in them. In order to ensure that the network is able to increase its sensitivity to informative features and suppress less useful ones, the final output of each level is obtained by assigning each channel a corresponding weighting factor W. Then we design a Guidance Loss ℓ_{guid} containing four sub-loss functions for each level $\hat{Y}'_k$. Let Y denotes the Ground Truth and δ refers to the activation function ReLu [18]:

$$\hat{Y}'_k = F(Y'_k, W) = \delta(W * Y'_k), \qquad \ell_{guid} = \sum_{k=1}^{4} \|\hat{Y}'_k - Y\|_2. \tag{3}$$

3.3 Semantic Loss

In the visual cortex, neurons are mapped to the visible or salient parts of an image and activated first, then followed by a later spread to neurons that are mapped to the missing parts [3,6]. Inspired by this visual cortex representation process, we design a hybrid semantic loss ℓ_{sem} to further sharpen the texture details of synthesized results and strengthen informative motion information, which consists of four sub-parts: Guidance Loss ℓ_{guid} mentioned above, Lateral Dependency Loss ℓ_{ld}, Attention Emphasis Loss ℓ_{emph}, and Gradient Loss ℓ_{grad}. First, to imitate the cortical neuron filling-in process and capture lateral dependency between neighbors in the visual cortex, ℓ_{ld} is proposed:

$$\begin{aligned}\ell_{ld} = \frac{1}{N}\sum_{i,j=1}^{N} &\big|\|\hat{Y}_{i,j} - \hat{Y}_{i-1,j}\|_2 - \|Y_{i,j} - Y_{i-1,j}\|_2\big| + \\ &\big|\|\hat{Y}_{i,j} - \hat{Y}_{i,j-1}\|_2 - \|Y_{i,j} - Y_{i,j-1}\|_2\big|.\end{aligned} \tag{4}$$

Second, ℓ_{emph} is employed to strengthen the texture and motion information of attention objects in the scene, namely, to emphasize the gradients for attention objects through feedback values during the back-propagation. As shown in Fig. 5(b), we take advantage of the excellent Faster R-CNN [20] to extract the foreground attention objects through a priori bounding box where (W_{box}, H_{box}) is the pair of width and height. Then we define the Attention Emphasis Loss ℓ_{emph} as follows:

$$\begin{aligned}\ell_{emph} = \frac{1}{W_{box} \times H_{box}} \sum_{i,j}^{(i,j)\in box} &\big|\|\hat{Y}_{i,j} - \hat{Y}_{i-1,j}\|_2 - \|Y_{i,j} - Y_{i-1,j}\|_2\big| + \\ &\big|\|\hat{Y}_{i,j} - \hat{Y}_{i,j-1}\|_2 - \|Y_{i,j} - Y_{i,j-1}\|_2\big|.\end{aligned} \tag{5}$$

Finally, ℓ_{grad} is also used to sharpen the texture details by incorporating with image gradients as shown in Eq. 6, and similar operation is also described in [15]. In summary, the semantic loss ℓ_{sem} is a weighted sum of all the losses in our experiment where $\alpha = 1, \beta = 0.3, \gamma = 0.7, \lambda = 1$ are the weights for Guidance Loss, Lateral Dependency Loss, Attention Emphasis Loss and Gradient Loss, respectively:

$$\ell_{grad} = \left|\nabla\hat{Y} - \nabla Y\right|^2. \tag{6}$$

$$\ell_{sem} = \alpha\ell_{guid} + \beta\ell_{ld} + \gamma\ell_{emph} + \lambda\ell_{grad}. \tag{7}$$

4 Experiments

In this section, we present comprehensive experiments to analyze and understand the behavior of our model. We first evaluate our model in terms of qualitative and quantitative performance for video interpolation and extrapolation. Then we show more capacities of our PMSN on various datasets. In the end, we analyze the effectiveness of different components in the PMSN separately. **Datasets:** We train our network on 153,971 triplet video sequences sampled from UCF-101 [24] dataset, and test the performance on UCF-101 (validation), HMDB51 [8], and YouTube-8m [1] datasets. **Training Details:** We adopt an Adam [7] solver to learn the model parameters by optimizing the semantic loss. The batch size is set as 32, and our initial learning rate is 0.005 that decays every 50K steps. We train the model for 100K iterations. The source code will be released in the future. **Baselines:** Here, we divide existing video synthesis methods into three categories for comparison: **(1) Interpolation-Only,** Phase-based frame interpolation [16] is a traditional and well-performed method just for video interpolation. Ada-Conv [19] also only focuses on video frame interpolation task via adaptive convolution operations. **(2) Extrapolation-Only,** PredNet [13] is a predictive coding inspired CNN architecture. MCNet [25] predicts frames by decomposing motion and content. FSTN [14] and Dual-Motion GAN [11] both only focus on video extrapolation task, and the authors do not release their pre-trained weights or training details. Hence, we only compare the PSNR and SSIM presented in their paper. **(3) Interpolation-Plus-Extrapolation,** EpicFlow [22] is a state-of-the-art approach for optical flow estimation, the synthesized frames are constructed by pixel compensation. For CNN-based methods, BeyondMSE [15] is a multi-scale architecture. The official model is trained by using 4 and 8 input frames. Since our method uses 2 input frames, BeyondMSE with 2 input frames is implemented for comparison. U-Net [23], which has a well-received structure for pixel-level generation, is also implemented for comparison. Deep Voxel Flow (DVF) [12] trains a deep network that learns to synthesize video frames by flowing pixel values from existing frames.

4.1 Quantitative and Qualitative Comparison

For quantitative comparison, we use both Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index to evaluate the image quality of interpolated/extrapolated frames, higher values of PSNR and SSIM indicate better results. In terms of qualitative quality, our approach is compared with several latest state-of-the-art methods in Figs. 6 and 7.

Single-Frame Synthesis. As shown in Table 1, it is obvious that our solution outperforms all existing solutions. Compared with the existing best interpolation-only solution Ada-Conv and best extrapolation-only solution Dual-Motion GAN, over 0.5 dB and 0.8 dB PSNR improvement can be achieved respectively. Compared with the existing best interpolation-plus-extrapolation scheme Deep Voxel Flow, over 2.2 dB and 1.7 dB PSNR improvement can be achieved

Table 1. Performance of frame synthesis on UCF-101 validation dataset.

Methods	Interpolation		Extrapolation	
	PSNR	SSIM	PSNR	SSIM
Pred Net [13]	—	—	22.6	0.74
Phase-based [16]	28.4	0.84	—	—
Beyond-MSE [15]	28.8	0.90	28.2	0.89
Epic-Flow [22]	30.2	0.93	29.1	0.91
U-Net [23]	30.2	0.92	29.2	0.92
FSTN [14]	—	—	27.6	0.91
MCNet [25]	—	—	28.8	0.92
Deep Voxel Flow [12]	30.9	0.94	29.6	0.92
Dual-Motion GAN [11]	—	—	30.5	0.94
Ada-Conv [19]	32.6	0.95	—	—
Ours	**33.1**	**0.96**	**31.3**	**0.94**

for interpolation and extrapolation operation respectively. We also show some subjective results for perceptual comparison. As illustrated in Figs. 6 and 7, our PMSN demonstrates better perceptual quality with clearer integrated objects, non-blurred background scene and more accurate motion prediction, compared with existing solutions. For example, Ada-Conv generates strong distortion and losses partial object in the bottom-right "leg" area due to failed motion prediction. On the contrary, our PMSN demonstrates much better perceptual quality without obvious artifacts.

Multi-Frame Synthesis. We further explore the multi-frame synthesis ability of our PMSN on various datasets, which can be used for up-sampling video frame rate and generating videos with slow-motion effect. We can see that the qualitative results in Fig. 8(a) have reasonable motion and realistic texture. And as demonstrated in Fig. 8(b), the PMSN can provide outstanding performance compared with other state-of-the-art methods.

4.2 Generalization Ability

Furthermore, we show the generalization ability of our PMSN by evaluating the model on YouTube-8m and HMDB-51 validation datasets without re-training. Table 2 demonstrates that our model outperforms all previous state-of-the-art models by a even larger gain (over 1.2 dB PSNR improvement on both datasets for interpolation and extrapolation) compared with results in Table 1, which means our PMNS has a much better generalization ability.

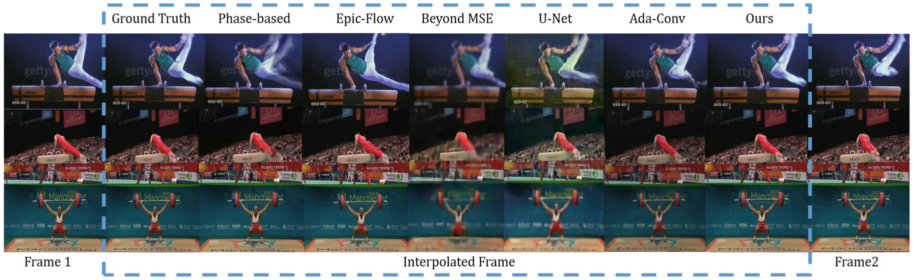


Fig. 6. Qualitative comparisons of video interpolation.

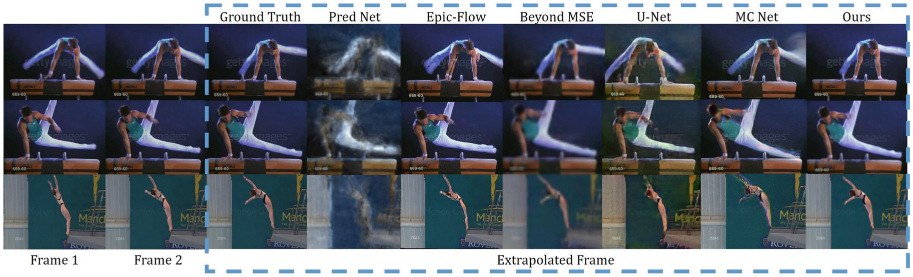
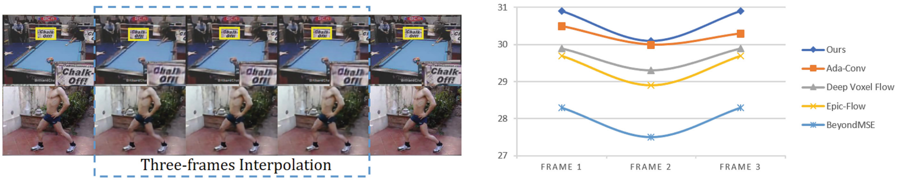


Fig. 7. Qualitative comparisons of video extrapolation.



(a) (b)

Fig. 8. (a) Three-frame interpolation. (b) Performance comparisons on three-frame interpolation.

4.3 Ablation Study

Effectiveness of Coarse-to-Fine Guidance Scheme: We first visualize the output of each deconvolutional block in the decoder stage, which indicates these gradually improved results using MAFs with different resolutions through a coarse-to-fine guidance scheme. As shown in the gray-images of Fig. 9(a), the texture details of the image are enhanced progressively, and the texture of the object becomes increasingly realistic.

In addition, as we mentioned in Sect. 3.1 that frame augmentation scheme can be replaced by any other frame synthesis solution, we adopt more complex adaptive convolution [19] to replace our basic generation of augmented frames

Table 2. Performance of frame synthesis on YouTube-8M and HMDB-51 validation datasets.

Methods	Interpolation		Extrapolation	
	PSNR	SSIM	PSNR	SSIM
Pred Net	—	—	19.7/18.4	0.65/0.59
Phase-based	21.0 /21.7	0.66/0.68	—	—
U-Net	24.2/23.8	0.73/0.72	22.7/22.4	0.70/0.70
BeyondMSE	26.6/26.8	0.78/0.80	25.7/26.1	0.74/0.76
MCNet	—	—	26.9/27.9	0.79/0.81
Epic-Flow	29.5/29.5	0.92/0.92	29.2/29.3	0.90/0.92
Ada-Conv	29.5/29.6	0.93/0.92	—	—
Ours	**31.1/31.4**	**0.94/0.92**	**30.4/30.7**	**0.94/0.93**

(MAFs) in video frame interpolation experiment, then we find that our PMSN obtains extra 0.5 dB gain in PSNR. In general, above ablation studies demonstrate that the proposed coarse-to-fine guidance scheme is really effective in further improving synthesis quality.

Effectiveness of MAFs: As shown in Fig. 9(b), the pure results of MAFs are unsatisfactory with a certain degree of blocking artifacts and uneven motions, the results without MAFs also have significant blur artifacts, which demonstrates that MAFs can provide informative motion tendency and texture details for synthesis.

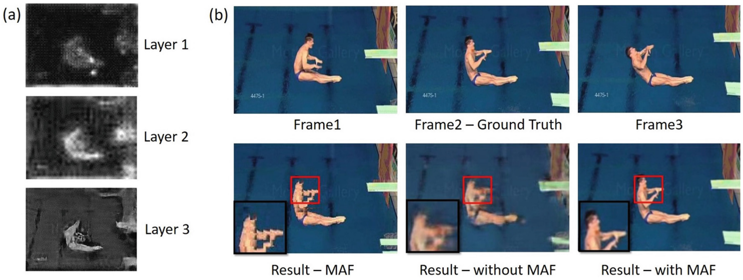


Fig. 9. (a) Output of each layer in the decoder stage. (b) Interpolation example.

Effectiveness of Semantic Loss: The Semantic Loss ℓ_{sem} is comprised of Guidance Loss ℓ_{guid}, Lateral Dependency Loss ℓ_{ld}, Attention Emphasis Loss ℓ_{emph}, and Gradient Loss ℓ_{grad}. To evaluate the contribution of each loss, we implement four related baselines for comparison. As shown in Table 3, we find that $\ell_{ld}+\ell_{guid}$, $\ell_{ld}+\ell_{emph}$ and $\ell_{ld}+\ell_{grad}$ are all higher than basic ℓ_{ld}, which

Table 3. Performance of hybrid losses.

Methods	Interpolation		Extrapolation	
	PSNR	SSIM	PSNR	SSIM
ℓ_{ld}	31.9	0.92	30.5	0.90
$\ell_{ld} + \ell_{guid}$	32.4	0.93	30.6	0.90
$\ell_{ld} + \ell_{grad}$	32.8	0.95	30.9	0.91
$\ell_{ld} + \ell_{emph}$	32.9	0.95	31.0	0.93
$\ell_{\mathbf{sem}}$	**33.1**	**0.96**	**31.3**	**0.94**

means that Guidance, Attention Emphasis and Gradient Loss lead to better performance. The combination of them further improves the overall performance.

5 Conclusions

In order to solve the problems existing in the traditional synthesis framework based on pixel motion estimation or learning based solutions, we try to effectively combine the advantages of the two solutions by establishing the proposed Progressive Motion-texture Synthesis Network (PMSN) framework. Based on the augmented input, the network can obtain informative motion tendency and enhance the texture details of synthesized video frames through the well-designed coarse-to-fine guidance scheme. In the learning stage, a brain-inspired semantic loss is introduced for further refining the motion and texture of objects. We perform comprehensive experiment to verify the effectiveness of PMSN. In the future, we expect to extend PMSN to other types of tasks such as video tracking, video question answering, etc.

Acknowledgement. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2016YFC0801001, the National Program on Key Basic Research Projects (973 Program) under Grant 2015CB351803, NSFC under Grant 61571413, 61632001, 61390514.

References

1. Abu-El-Haija, S., et al.: Youtube-8m: a large-scale video classification benchmark (2016). arXiv preprint: arXiv:1609.08675
2. Choudhary, S., Varshney, P.: A study of digital video compression techniques. PARIPEX-Indian J. Res. **5**(4), 39–41 (2016)
3. De Weerd, P., Gattass, R., Desimone, R., Ungerleider, L.G.: Responses of cells in monkey visual cortex during perceptual filling-in of an artificial scotoma. Nature **377**, 731–734 (1995)
4. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: NIPS, pp. 64–72 (2016)

5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
6. Huang, X., Paradiso, M.A.: V1 response timing and surface filling-in. J. Neurophysiol. **100**(1), 539–547 (2008)
7. Kingma, D., Ba, J.: Adam: a method for stochastic optimization (2014). arXiv preprint: arXiv:1412.6980
8. Kuehne, H., Jhuang, H., Stiefelhagen, R., Serre, T.: HMDB51: a large video database for human motion recognition. In: Nagel, W., Kröner, D., Resch, M. (eds.) High Performance Computing in Science and Engineering 2012, pp. 571–582. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33374-3_41
9. Li, S., Yeung, D.Y.: Visual object tracking for unmanned aerial vehicles: a benchmark and new motion models. In: AAAI, pp. 4140–4146 (2017)
10. Li, W., Cosker, D.: Video interpolation using optical flow and laplacian smoothness. Neurocomputing **220**, 236–243 (2017)
11. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion GAN for future-flow embedded video prediction. In: ICCV (2017)
12. Liu, Z., Yeh, R., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: ICCV, vol. 2 (2017)
13. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. In: ICLR (2017)
14. Lu, C., Hirsch, M., Schölkopf, B.: Flexible spatio-temporal networks for video prediction. In: CVPR, pp. 6523–6531 (2017)
15. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
16. Meyer, S., Wang, O., Zimmer, H., Grosse, M., Sorkine-Hornung, A.: Phase-based frame interpolation for video. In: CVPR, pp. 1410–1418 (2015)
17. Michalski, V., Memisevic, R., Konda, K.: Modeling deep temporal dependencies with recurrent grammar cells. In: NIPS, pp. 1925–1933 (2014)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010) (2010)
19. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: CVPR, vol. 2, p. 6 (2017)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS, pp. 91–99 (2015)
21. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: AAAI, pp. 1495–1501 (2017)
22. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: edge-preserving interpolation of correspondences for optical flow. In: CVPR, pp. 1164–1172 (2015)
23. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015, Part III. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
24. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: a dataset of 101 human actions classes from videos in the wild (2012). arXiv preprint: arXiv:1212.0402
25. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. In: ICLR, vol. 1(2), p. 7 (2017)
26. Wang, Y., Long, M., Wang, J., Gao, Z., Philip, S.Y.: PredRNN: recurrent neural networks for predictive learning using spatiotemporal LSTMs. In: NIPS, pp. 879–888 (2017)

27. Xue, T., Wu, J., Bouman, K., Freeman, B.: Visual dynamics: probabilistic future frame synthesis via cross convolutional networks. In: NIPS, pp. 91–99 (2016)
28. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions (2015). arXiv preprint: arXiv:1511.07122