Video Snapshot: Single Image Motion Expansion via Invertible Motion Embedding

Qianshu Zhu*, Chu Han*, Guoqiang Han, Tien-Tsin Wong, and Shengfeng He

***Abstract*—Unlike images, finding the desired video content in a large pool of videos is not easy due to the time cost of loading and watching. Most video streaming and sharing services provide the video preview function for a better browsing experience. In this paper, we aim to generate a video preview from a single image. To this end, we propose two cascaded networks, the Motion Embedding Network and the Motion Expansion Network. The Motion Embedding Network aims to embed the spatio-temporal information into an embedded image, called *video snapshot*. On the other end, the Motion Expansion Network is proposed to invert the video back from the input video snapshot. To hold the invertibility of motion embedding and expansion during training, we design four tailor-made losses and a motion attention module to make the network focus on the temporal information. In order to enhance the viewing experience, our expansion network involves an interpolation module to produce a longer video preview with a smooth transition. Extensive experiments demonstrate that our method can successfully embed the spatio-temporal information of a video into one "*live*" image, which can be converted back to a video preview. Quantitative and qualitative evaluations are conducted on a large number of videos to prove the effectiveness of our proposed method. In particular, statistics of PSNR and SSIM on a large number of videos show the proposed method is general, and it can generate a high-quality video from a single image.**

***Index Terms*—Video Snapshot, Video Expansion, Information Embedding, Motion Attention**

I. Introduction

Videos can be found everywhere on social media in our daily life, especially for video sharing services like YouTube or TikTok. There are so many videos that we cannot easily browse to the desired video content. Therefore all these video sharing services provide a video preview function for enhancing the browsing experience. However, enabling video preview requires additional storage of 3-second videos, which may not be a memory-efficient solution. Various methods convert a video to a single or a sequence of image(s) to generate video abstraction [1], [2], [3]. The resulted images may be representative, but they cannot generate video previews. On the other hand, some other works [4], [5], [6] are proposed to make a still image moving by leveraging the inference from the image itself, but they are independent to the video content. We then raise a question – can we embed a video into a single or a few frames? It is believed that an image is unexpandable, but what if we embed spatio-temporal information as encoding patterns in the image so that they can be inverted back to the video?

In this paper, we propose to embed the motion information of a video into an embedded image. We call this embedded image "*video snapshot*", as shown in Fig. 1(a). The embedded video snapshot is actually "alive" that it can be decoded to the input video. Fig. 1(b) demonstrates the restored frames generated by our proposed method from only one video snapshot in Fig. 1(a). The PSNR, SSIM and MAE values, as well as the difference maps, show that the quality of the restored frames in Fig. 1(b) is extremely high. It has almost no visual difference comparing with the groundtruth, which is more than enough for a video.

Our proposed method consists of two major components, the Motion Embedding Network and the Motion Expansion Network. The Motion Embedding Network is proposed to embed motion information into a single image. However, the complex motion changes make it challenging and it cannot be naively solved by a common encoder. Thus we propose a motion attention module to enforce the network concentrates to the dynamic regions. With the feature maps that convey the spatio-temporal information, an encoder is introduced to embed them into the video snapshot. The Motion Expansion Network, which consists of a decoder and an interpolation module, is designed to bring the video snapshot back life. The decoder inverts the embedded image back to the input frames. However, the decoder alone cannot provide additional information that does not exist during the encoding process. Moreover, it is not capable for a lightweight encoder to embed too much information into only one image. For example, even the encoder encodes 11 frames into one video snapshot, which has already pushed the encoder to the limit. The decoder can give us at most 11 frames, which produces an around 0.3 second video (at 30 FPS). It is too short for a video to deliver sufficient information with good viewing experience. Thus, we propose an interpolation module after the decoder to breakthrough the limit. With the proposed Motion Expansion Network, we can obtain a longer and smoother video

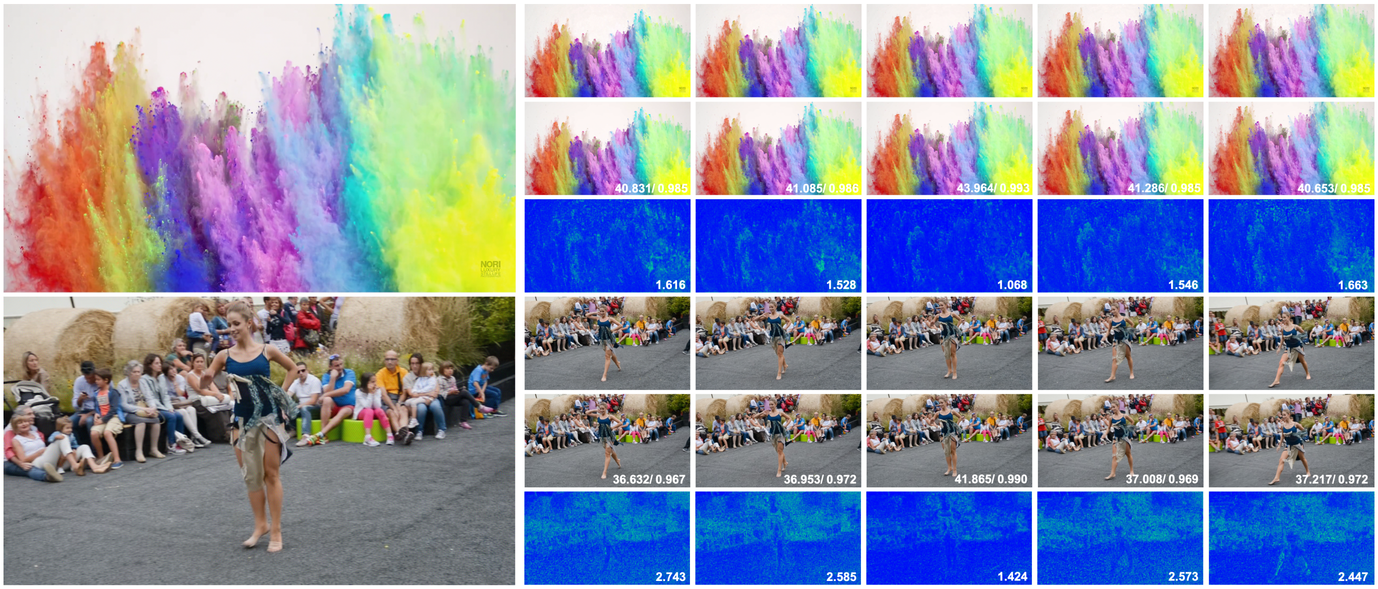
This project is supported by the National Natural Science Foundation of China (No. 61472145, No. 61972162, and No. 61702194), the Special Fund of Science and Technology Research and Development of Applications From Guangdong Province (SF-STRDA-GD) (No. 2016B010127003), the Guangzhou Key Industrial Technology Research fund (No. 201802010036), the Guangdong Natural Science Foundation (No. 2017A030312008), and the CCF-Tencent Open Research fund (CCF-Tencent RAGR20190112). (Corresponding author: Shengfeng He.)

Qianshu Zhu, Guoqiang Han, and Shengfeng He are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. E-mail: chuchienshu@gmail.com, csgqhan@scut.edu.cn, hesfe@scut.edu.cn.

Chu Han is with the Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China. E-mail: zq1992@gmail.com.

Tien-Tsin Wong is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China. E-mail: ttwong@cse.cuhk.edu.hk.

*The first two authors contribute equally.



(a) Video Snapshot

(b) GT, restored frames, difference maps

Fig. 1. Our proposed method encodes spatio-temporal information of the frame sequence into an embedded image (a), called video snapshot. Later, this video snapshot can be restored back to the video. In (b), images from the first row to the third row of each case are the groundtruth frames, restored frames from the video snapshot and the difference map of MAE. Note that, since we have to compare the restored frames with the groundtruth frames, the results in (b) are the direct output from the decoder without interpolation. Our complete outputs with interpolation in GIF format can be found in the supplementary materials. The PSNR, SSIM and MAE values are shown at the bottom right corner in each corresponding image.

from only one video snapshot. Furthermore, four tailor-made losses are introduced to guarantee the invertibility of motion embedding and motion expansion as well as the faithfulness of the results during the training phase. Our well-trained model is general enough and can take arbitrary input videos with no matter large or small motions. We also demonstrate the application of making a long video preview using concatenated video snapshots.

We have conducted extensive experiments to both qualitatively and quantitatively evaluate our proposed network. An ablation study is introduced to demonstrate the impact of each specific component and different settings in our network, *e.g.*, how many frames can be compacted into one video snapshot. A user study is conducted to demonstrate the faithfulness of our restored videos comparing with the groundtruth. The complete results including the groundtruth in GIF format can be found in the supplementary materials. The contributions of our paper are summarized as follows:

- We propose a general and innovative method to embed the spatio-temporal information of a video into a video snapshot, which can be inverted back to a long video with smooth motion.
- We present a motion attention module to help the network focus on dynamic regions, which enriches the spatio-temporal representations of the learned features.
- We develop an interpolation network that predicts arbitrary intermediate optical flow between two consecutive frames. It allows generating a longer and smoother video preview.

II. Related Work

Given a video, our work aims to generate a video snapshot which can be inverted back to the video. But currently, there is no other research shares the same spirit with ours. Therefore, we discuss the related works on two aspects according to the sub-goals of our approach, video to image and image to video. We also discuss related steganography methods as they aim to embed information in different forms.

A. Video to Image

Video Abstraction: The major purpose and the most related research field to convert a video to a single or a sequence of image(s) is video abstraction/summarization. Video abstraction [1], [3], [7] actually gives a short summary of video content according to the saliency of video frames. Similar to video summarization, video highlight methods [8], [9] are proposed to identify the significant sub-events of the video.

However, the goals of the above methods are different from ours. Existing video summarization methods aims to find out the representative frames of the video. In contrast, our proposed method is designed to hide the spatio-temporal information into a single image while endows it the invertibility.

Video Compression: Although video compression [10], [11], [12], [13], [14] is not to compress a video to a single image, it reduces the size of the video that shares a similar spirit of ours. However, the proposed method is significantly different from video compression due to the focus of the intermediate representation of the video. Our work generates a unique type of image representation, video snapshot, that can be viewed, printed, and processed in different applications like printing on a paper for animation purposes.

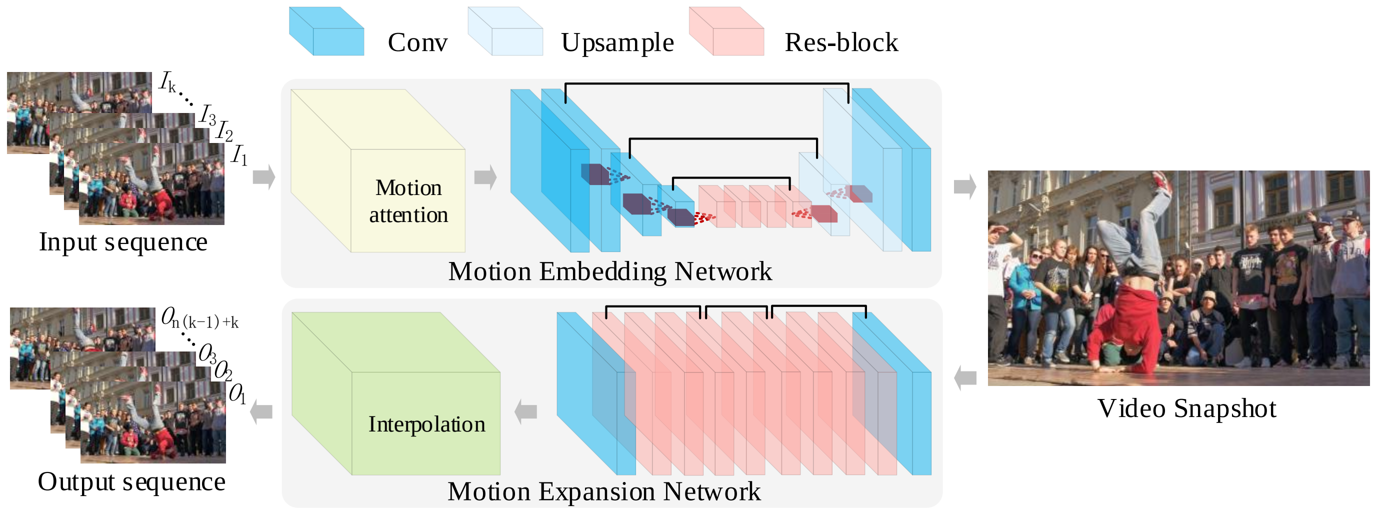


Fig. 2. Overall architecture of our proposed method. The Motion Embedding Network aims to encode a sequence of consecutive frames $\mathcal{I}$ into a video snapshot by a motion attention module and an encoder. The Motion Expansion Network then tries to make the video snapshot alive and generates the output sequence $\mathcal{O}$ by a decoder and an interpolation network. Note that, with the interpolation network, the number of output frames $\mathcal{O}$ will be much larger than the original input sequence $\mathcal{I}$. This leads to a much longer and smoother video than the original one.

B. Image to Video

Video Generation: Generating videos from a single image or very limited examples is challenging but interesting. Existing works have already shown a lot of fascinating effects, *i.e.*, generating video motions, cartoon/sketch animations, human poses, etc.

Horry *et al.* [15] propose an early attempt, "Tour Into the Picture", for making animations from 2D pictures or photographs. Motion texture techniques [16], [5], [17] are proposed to synthesize the animated motions of a still image with higher image quality and resolution. Joshi *et al.* [6] generate "Cliplets", which is a stable video with looped small motion. It is obtained from a handheld video by interactively segmenting the regions of the desired motion. Some methods [18], [19] animate humans or human-like subjects by utilizing the skeleton representations. Dvoroznák *et al.* [20] present an example-based method for generating cartoon animation which can preserve the visual appearance and stylized motion. Xu *et al.* [4] animate animal motions from a still image by rearranging the motion snapshots of repeated animals in the image. Su *et al.* [21] propose a video-driven dynamic deformation method which allows users to interactively bring static drawings to life. Thanks to the advanced feature representation and generation of the neural network, more and more learning-based techniques are proposed for image to video generation with higher quality, *i.e.*, facial animation [22], cinemagraph [23], 3D animatable character [24], 3D human bodies [25], etc. Although the above methods can show visually pleasing effects, their objective is not to generate an invertible video snapshot.

Video Prediction: Video prediction aims to predict the following few frames of a given frame [26], [27], [28], [29], [30], [31]. It shares the same spirit of ours to expand a single image to a short video sequence. However, video prediction depends solely on the learned spatio-temporal patterns, which is impossible to recover the original video in high-quality. On the contrary, the proposed method embeds spatio-temporal patterns into a single image, leading to high-quality video recovery.

C. Steganography

Steganography aims to hide confidential information within different information carriers such as text, audio, and IP datagram. The most widely adopted media is digital image. Early image steganography involves adjusting the least significant bits (LSBs) of each image pixel depending on the bits of the secret message [32], [33]. However, LSB methods produce image perturbations with fixed filters, leading to easily detected embedding. Latest image steganography leverages deep network for better information hiding. Zhu *et al.* [34] propose to hide secret messages within an image using deep networks by considering noise interference, while Wengrowski *et al.*[35] embed information in light field images. Other than image steganography, Yang *et al.* [36] aim to hide a video in an audio without compromising of the audio fidelity. Sharing a similar spirit, we explores the possibility of hiding a video in an image.

III. Proposed Method

Our proposed network shown in Fig. 2 consists of two subnetworks, the Motion Embedding Network and the Motion Expansion Network. The Motion Embedding Network (discussed in Sec. III-A) takes a consecutive frame sequence from a video as the inputs and aims to encode the whole frame sequence into an embedded image, called "video snapshot". The Motion Expansion Network (discussed in Sec. III-B) is proposed to make video snapshot alive again and generate a longer and smoother video than the original one.

A. Motion Embedding Network

Given a sequence of consecutive frames $\mathcal{I} = \{I_i | i = 0, 1...k\}$ from a video, we proposed the Motion Embedding

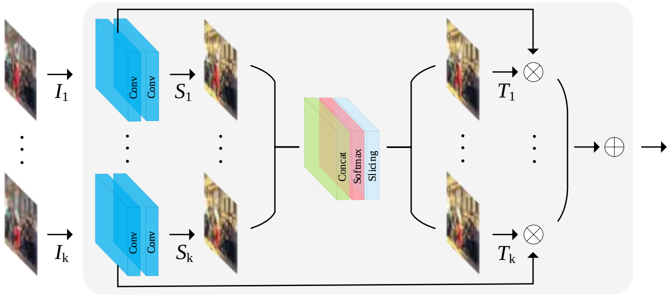


Fig. 3. Network structure of Motion Attention Module.

Network to encode the color changes and motions along the temporal domain into a video snapshot E. The complex color and motion changes make it hard to achieve by using a simple encoder. Thus, we propose the Motion Attention Module to find out where the motions happen. Then we cooperate the motion attention map with an encoder to embed the spatio-temporal information into the video snapshot.

1) Motion Attention Module: To capture the motions along the frame sequence, we propose a motion attention mechanism. Fig. 3 shows the network structure of motion attention module. For each input frame I_i, we first extract the spatial features S_i by an independent branch consists of two convolutional layers. The first one is utilized to extract the spatial image features. The second one is a 1×1 convolutional layer to reduce the dimension of the channel to 1. Since finding motion mainly relies on local matching, we find two convolutional layers achieve a good balance between the performance and computational cost. Now the extracted feature map of each branch only focuses on the spatial feature of each frame independently. We let the network further pays attention to the motion along the temporal domain by building up the connections of all frames. Note that, shown in Fig. 3, we apply a Jet colormap on attention map for better visualization effect. Thus, we concatenate the feature maps from all branches and feed them into a Softmax layer as follows:

$$T_{i,j} = \frac{\exp\left(S_{i,j}\right)}{\sum_{i=1}^{k} \exp\left(S_{i,j}\right)}. \tag{1}$$

Here, the j-th pixel in T_i is calculated by the Softmax of the pixels at the same coordinate over the time axis. Then a slicing layer is applied to cut the feature maps into k 1-channel motion attention maps $T = \{T_i | i = 0, 1...k\}$. The skip connection [37] of each branch is introduced to monitor the motion attention map using the spatial information by an element-wise multiplication. Lastly, we can obtain the complete spatio-temporal information by concatenating all the motion attention maps. Fig. 4 visualizes two consecutive motion attention maps by directly drawing them onto their reference frames. We can observe that the visualized results successfully convey the attentions on motions in the following frames.

2) Encoder: Following the motion attention module, the encoder is introduced to embed the spatio-temporal information into the video snapshot. It contains two convolutional blocks, four residual blocks [38] and two deconvolutional blocks. Additionally, the skip connections preserve much more low-level features and suppress the blurring artifacts. The output embedded image E carries not only the spatial information but also the temporal information. Fig. 5(b) shows the video snapshot and its blow up area. Comparing with the groundtruth image in Fig. 5(a), they are almost visually the same except some barely visible patterns. When we zoom into the yellow box, regular patterns are shown more clearly in Fig. 5(b). Here we call these pattern the “motion textures”, where the spatio-temporal information hides. It reveals that our proposed Motion Embedding Network identifies motions, and they are successfully encoded into the video snapshot. In our experiments, we set the number of input frames $k = 5$ due to its best visual quality. Some discussions on the numbers of embedding frames are presented in Sec. IV-C.

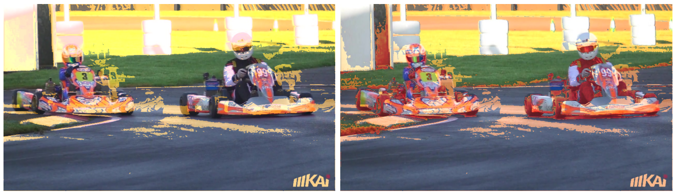
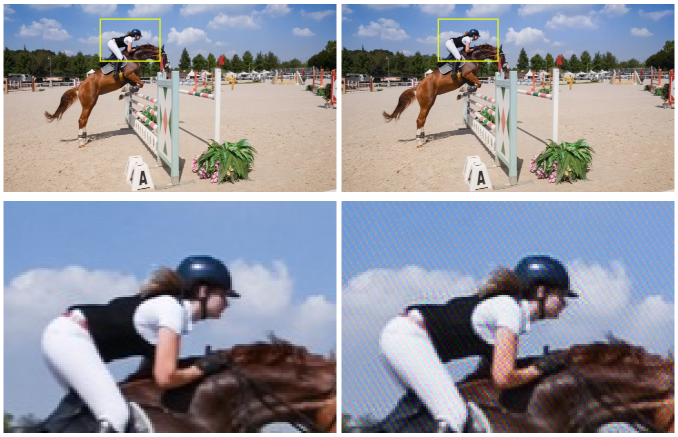


Fig. 4. Motion attention maps on two consecutive frames. We visualize the motion attention maps (mapped to color space) by directly drawing it onto their reference frames.



(a) GT (b) video snapshot

Fig. 5. Visualization of video snapshot. (a) is the groundtruth and (b) is video snapshot generated by Motion Embedding Network. When we zoom in the images, motion textures can be observed at the bottom row of (b). We can easily observe that the patterns in static and moving areas are different, like the motion textures around the girl.

B. Motion Expansion Network

To bring the video snapshot back to live, we propose the Motion Expansion Network. It consists of two components, a decoder and an interpolation module. The decoder is introduced to restore the video snapshot back to the original frame sequences. Although it can decode high quality frames, the restored video is still as short as the input one. Because it is impractical for an encoder to compact so much information into one single image, especially for abrupt motions. Also, a too short video cannot offers a good viewing experience.

(a) GT (b) Decoded image

Fig. 6. Demonstration of the decoded image output by the decoder. Here, we only select one frame from the output sequence for a simple illustration. The difference between the groundtruth image (a) and decoded image (b) is visually imperceptible. The SSIM and PSNR values between (a) and (b) are: SSIM:0.9933 PSNR:40.166.

In order to obtain a longer video, a pre-trained interpolation module is proposed after the decoding process.

1) Decoder: The decoder is made up by eight residual blocks, one flat convolutional layer, and skip connections across different layers. It takes the video snapshot as input and output a frame sequence. Fig. 6(b) shows a selective output frame from the decoder. When comparing it with the groundtruth frame in Fig. 6(a), the visual difference between them is completely unobservable. SSIM and PSNR values also show that the decoder alone has already provided the capacity to restore the high quality frame sequence from only one embedded image. Training the decoder together with the Motion Embedding Network, we can transfer a video to a video snapshot and vise versa. To hold the invertibility training, we tailored four losses.

The following losses are designed for the training of Motion Embedding Network with the decoder. They are utilized to measure how good the Motion Embedding Network can hide spatio-temporal information into the embedded image, and how good the decoder can restore the original frames back.

Given a sequence of consecutive frames $\mathcal{I} = \{\mathcal{I}_i | i = 0, 1...k\}$ from a video, the Motion Embedding Network generates the embedded image E. The decoder restores the frames $\mathcal{I}' = \{\mathcal{I}'_i | i = 0, 1...k\}$.

a) Restoration Loss: We introduce an L_2 loss to model how good the network can restore frames from the embedded images.

$$\mathcal{L}_r = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{k}\|\mathcal{I}'_{i,j} - \mathcal{I}_{i,j}\|_2, \tag{2}$$

where N is the total number of the videos. k is the number of frames in each video. $\mathcal{I}_{i,j}$ denotes to the i-th frame in the j-th video.

b) Embedding Loss: Since we want the video snapshot can be the representative image of the video. We select the intermediate frame as the reference image of the video snapshot. The embedding loss is introduced to ensure the color consistency between the video snapshot and the reference image. The reason why we choose the intermediate frame of the frame sequence is that we want to balance the motion differences of the reference image and the first/last frame. We do not want the difference between the reference image and any other frame to be too large or too small. Thus, the embedding loss is defined as follows:

$$\mathcal{L}_e = \frac{1}{N}\sum_{j=1}^{N}\|max\{|E_j - \bar{\mathcal{I}}_j|, M_\theta\}\|_1, \tag{3}$$

where M_θ is a threshold matrix with the same size of the image resolution that filled with constant θ. $\bar{\mathcal{I}}_j$ is the intermediate frame in the j-th video. We loose the color difference to obtain a larger solution space for spatio-temporal information encoding. Empirically, we set θ = 90.

c) Perceptual Loss: To guarantee a fine and sharp video snapshot, we introduce a perceptual loss [40] to enforce the perceptual similarity, *e.g.*, sharpness, global contrast, etc.

$$\mathcal{L}_p = \frac{1}{N}\sum_{j=1}^{N}\|\phi(E_j) - \phi(\bar{\mathcal{I}}_j)\|_2, \tag{4}$$

where ϕ denotes the conv4_4 features of an ImageNet pre-trained VGG-19 model [41].

d) Unimodal Loss: Since the attention maps might go astray if we left it uncontrolled. Also, we observe that without unimodal loss, the pixel values of temporal attention map $T_{\frac{k+1}{2}}$ are close to 1 while the others are almost 0. That is to say, the other frames except the intermediate one do not contribute to the Motion Embedding Network when generating the video snapshot. So we define the unimodal loss to avoid this unbalance problem as follows:

$$\mathcal{L}_u = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{k} max\{T_{i,j} \odot T_{k-i,j} - T^2_{\frac{k+1}{2},j}, 0\}, \tag{5}$$

where T is the attention map defined in Sec. III-A. Note that, we let k be an odd number in our paper so that we can acquire exactly one intermediate frame from one video.

Last, our final objective function is a linear combination of all losses:

$$\mathcal{L} = \alpha_r\mathcal{L}_r + \alpha_e\mathcal{L}_e + \alpha_p\mathcal{L}_p + \alpha_u\mathcal{L}_u \tag{6}$$

The loss weights are set empirically as α_r = 3, α_e = 1, α_p = 1, and α_u = 1.

2) Interpolation Module: Since we want to breakthrough the length of the video from the decoder. We propose an interpolation module which can achieve multiple frame interpolation. Given two consecutive frames I_1 and I_2, we aim to interpolate an frame I_t at arbitrary time $t \in (1, 2)$. We propose an interpolation network to estimate the intermediate optical flow, $F_{t\rightarrow1}$ and $F_{t\rightarrow2}$, at time t. Then the interpolated frame can be computed by a warping function as follows:

$$I_t = t \odot W(I_1, F_{t\rightarrow1}) + (1 - t) \odot W(I_2, F_{t\rightarrow2}), \tag{7}$$

where $\odot$ denotes the element-wise multiplication. $W(\cdot, \cdot)$ is the backward warping. Here, we consider the temporal consistency by weighting the two warped images. When the time t is closer to 1, I_1 should response more contribution and vise versa. Moreover, the warping function is differentiable.

We design the interpolation network using a U-Net architecture [37] as shown in Fig. 7. It takes $(I_1, I_2, \hat{F}_{t\rightarrow1}, \hat{F}_{t\rightarrow2}, \hat{I}_{t\rightarrow1}, \hat{I}_{t\rightarrow2})$ as the inputs and returns the refined flow

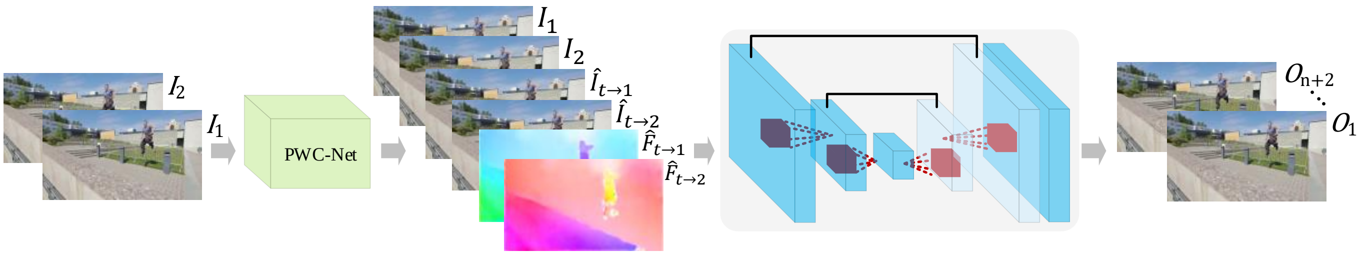


Fig. 7. Network structure of interpolation network. The PWC-Net [39] is used to calculate the optical flow. The second network is the proposed interpolation network. The interpolated frame can be obtained by warping according to the intermediate flow predicted by the interpolation network.

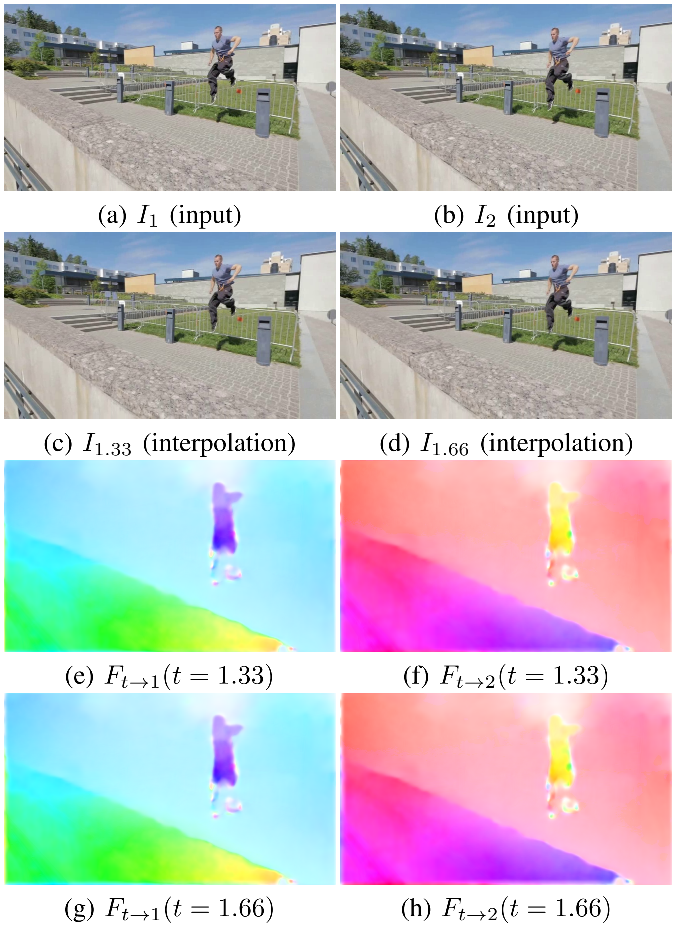

(a) I_1 (input)
(b) I_2 (input)
(c) $I_{1.33}$ (interpolation)
(d) $I_{1.66}$ (interpolation)
(e) $F_{t\to1}(t=1.33)$
(f) $F_{t\to2}(t=1.33)$
(g) $F_{t\to1}(t=1.66)$
(h) $F_{t\to2}(t=1.66)$

Fig. 8. Multiple frames interpolation. Given two input images (a) and (b), our proposed interpolation module can predict inter-frame at arbitrary time t. (c) and (d) are the results at timesteps $t=1.33$ and $t=1.66$ respectively. (e)-(h) are bi-directional optical flow generated by our interpolation network at timesteps $t=1.33$ and $t=1.66$, respectively.

$F_{t\to1}$ and $F_{t\to2}$. $\hat{F}_{t\to1}$ and $\hat{F}_{t\to2}$ are the approximated flow at time t which are calculated as follows:

$$\begin{aligned}\hat{F}_{t\to1} &= -(1-t)tF_{1\to2} + t^2F_{2\to1} \\ \hat{F}_{t\to2} &= (1-t)^2F_{1\to2} - t(1-t)F_{2\to1}\end{aligned} \quad (8)$$

where $F_{1\to2}$ and $F_{2\to1}$ are the bi-directional optical flow of $I_1$and I_2. They are calculated by PWC-Net [39]. $\hat{I}_{t\to1}$ and $\hat{I}_{t\to2}$ denote two approximated images warped by $\hat{F}_{t\to1}$ and $\hat{F}_{t\to2}$ respectively. With the refined intermediate flow, we can obtain the interpolation frame by the warping function in Eq. 7. Fig. 8 (c) & (d) shows the interpolation results when $t=1.33$ and $t=1.66$ respectively. Fig. 8 (e)-(h) are the bi-directional flow generated by our interpolation network.

a) Training: To train the interpolation network, we introduce several losses. Given two input images I_1 and I_2, we have a set of groundtruth intermediate frames $\{I_{t_i}|i=1,...,N\}$ and the interpolation frames $\{\hat{I}_{t_i}|i=1,...,N\}$. Here, the times $\{t_i|i=1,...,N\}$ are in a uniform distribution. First, an L_1 loss is introduced to measure the interpolation frames I_{t_i} with the groundtruth $\hat{I}_{t_i}$.

$$\mathcal{L}_{l_1} = \frac{1}{N}\sum_{i=1}^{N}\|\hat{I}_{t_i} - I_{t_i}\|_1. \quad (9)$$

We also introduce a perceptual loss [40] to measure the perceptual differences and to ensure the image sharpness.

$$\mathcal{L}_p = \frac{1}{N}\sum_{i=1}^{N}\|\phi(\hat{I}_{t_i}) - \phi(I_{t_i})\|_2, \quad (10)$$

where ϕ denotes the conv4_4 features of an ImageNet pretrained VGG-19 model [41].

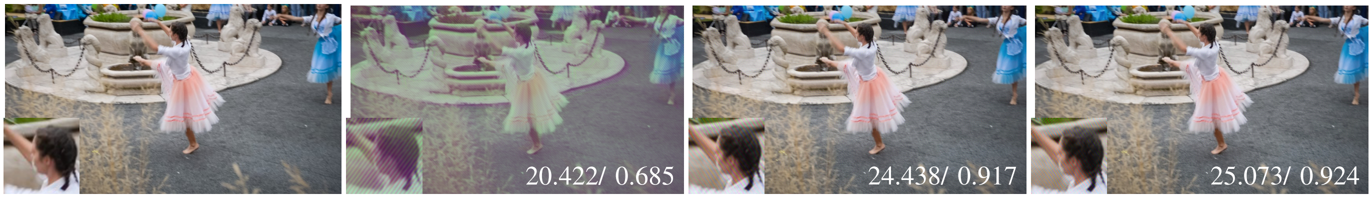
An additional Laplacian pyramid loss [42], [43] is introduced to provide the measurement from local to global features along the Laplacian pyramid. It is defined as follows:

$$\mathcal{L}_{lap} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{5}2^{i-1}\|L^j(\hat{I}_{t_i}) - L^j(I_{t_i})\|_1. \quad (11)$$

Here, we conduct a five-level Laplacian pyramid. The deeper levels in the pyramid make stronger contributions due to the larger spatial support. We trained the interpolation network using the Adobe240-fps [44]. We place the interpolation module after the decoder. Then our proposed Motion Expansion Network can breakthrough the limit of the information conveyed by the embedded image and obtain a much longer and smoother video.

IV. Experiments

In this section, we conduct several experiments to quantitatively and qualitatively evaluate our method. Since we are the first one trying to embed spatio-temporal information into one embedded image which can be converted back to live. We mainly focus on the analysis of the importance of each proposed component and the intuition at every specific design of our approach. Ablation studies are demonstrated in



(a) Groundtruth (b) w/o perceptual (c) w/o embedding (d) Full losses

Fig. 9. Visualization of the embedded image using two embedding related losses. Artifacts can be found at the blow up areas of (b) and (c). The embedded image generated from the complete model with full losses in (d) has the least artifacts and are more similar with the groundtruth image (a).

TABLE I
STATISTICS OF RESTORED FRAMES ON DIFFERENT NETWORK CONFIGURATIONS.

Configurations	PSNR	SSIM
w/o perceptual loss	38.498	0.9752
w/o embedding loss	39.113	0.9793
w/o unimodal loss	37.823	0.9733
w/o attention module	36.278	0.9612
Shared weights (attention)	36.372	0.9630
w/o two-stage training	38.821	0.9784
Final model	**39.346**	**0.9804**

TABLE II
STATISTICS OF RESTORED FRAMES ON DIFFERENT EMBEDDING RANGES.

Embedding Range	PSNR	SSIM
5	**39.346**	**0.9804**
7	36.557	0.9656
9	34.594	0.9493
11	32.372	0.9262

TABLE III
STATISTICS OF RESTORED FRAMES ON DIFFERENT FRAME INTERVALS.

Interval	PSNR	SSIM
1	39.123	0.9797
3	38.885	0.9787
5	38.807	0.9774

Sec. IV-C. We also conduct a user study in Sec. IV-F to compare our results with the groundtruth. More video snapshots and restored frames are shown in Sec. IV-E. The results in GIF format are shown in the supplementary materials. Unless explicitly stated otherwise, all the results and statistics shown in the experiments are from the model trained with 5 input frames.

A. Dataset

In this paper, we chose the 20BN-SOMETHING-SOMETHING-V2 dataset [45] to train our model. It consists of 220,847 video clips that capture people interacting with everyday objects. Since this work does not consider high-level semantic analysis, we only take 20,650 video clips from the original dataset for network training to trade-off the training time and the network capacity. For each video in the training data, we do not use the original frame rate because we want the network has the capacity to process larger motions. Thus, we extract frames with a frame interval $(n-1)$. It means that we only extract one frame in every n consecutive frames, where $n \in \{5, 6, 7\}$. The reason why we introduce the randomness of the frame interval is to improve the generality of our network on different motion levels.

In the testing phase, we use Davis 2017 [46], which contains 150 video sequences, as the test set. We then double the test set by random sampling two video clips in one original video without overlapping.

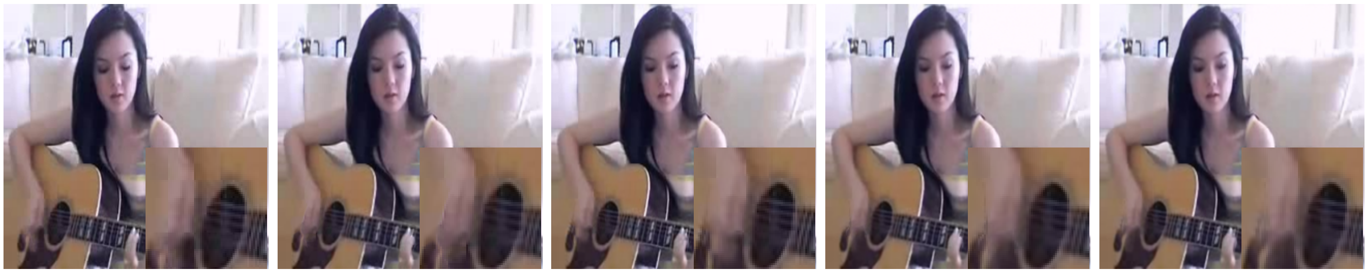
B. Training Details

The complete network training process involves two stages. In the first stage, we train the Motion Embedding Network with the decoder together for 90 epochs. The embedded image generated by the Motion Embedding Network is quantized to an integer, which may lead to information loss. In the second stage, we finetune only the decoder for extra 30 epochs to overcome the quantization error. By the two-stage training, decoder further improves the quality of temporal reconstruction. Statistics in ablation studies (Table I) show that the two-stage training strategy performs better than one-stage. In addition, the whole training used Adam optimizer [47]. We initialize the learning rate as 2e-4 and apply a polynomial decay to it. The minimal learning rate is set to 2e-6, the default power of the polynomial is 0.9. The batch size is up to 8.

C. Ablation Studies

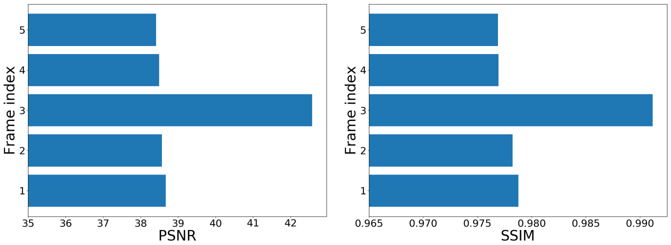
We conduct ablation studies to prove the importance and effectiveness of each component in our network. We first compare our models in different configurations. Then we discuss the information loss when increasing the embedding range, which is the number of input frames. Next, we discuss how large the motion will affect the performance in the testing phase by keep increasing the frame interval. Finally, we conduct a quantitative comparison of different restored frames.

a) Comparisons on Different Configurations: We compare our complete model with the following six variants in different configurations: 1) without perceptual loss; 2) without embedding loss; 3) without unimodal loss; 4) without motion attention module; 5) with motion attention module that shares weights of each branch; 6) w/o two-stage training mentioned in Sec. IV-B. Statistics are shown in Table I.



(a) Groundtruth (b) DVF [48] (c) SepConv [49] (d) SuperSlo [50] (e) Ours

Fig. 10. Qualitative comparison with existing interpolation methods. Our method produces a smooth result with less artifacts.



(a) PSNR (b) SSIM

Fig. 11. PSNR and SSIM on different restored frames.

TABLE IV
INTERPOLATION COMPARISON ON THE UCF101 DATASET.

Methods	PSNR	SSIM
DVF [48]	29.37	0.861
SepConv-L_1 [49]	30.18	0.875
SuperSloMo_Adobe240fps [50]	29.80	0.870
Ours	**30.23**	**0.881**

Since perceptual loss and embedding loss are designed for improving the quality of the embedded image. Without any of these two losses actually will not affect the information encoding and decoding. So the PSNR and SSIM [51] values of restored frames are comparable with our complete model. However, without these two losses, the embedded image (video snapshot) has more artifacts than the one generated by the complete model. Fig. 9 visualizes the embedded images generated by these two embedding related losses. The visual result and statistics from our complete model are the best with less artifacts.

The other components mainly focus on the encoding and decoding performances. Without the motion attention module, the PSNR and SSIM values decrease a lot which reveals the importance of our proposed motion attention module. The motion attention module can identify and emphasize the motions before passing the feature maps to the encoder. The unimodal loss helps the motion attention module to balance the weights of each individual frames and obtain more accurate motion attention maps. Without this loss, the quality of the restored frames also decreases. By sharing weights of convolutional layers in the motion attention module, it greatly affect the efficiency of motion encoding-decoding. It reveals that the features of different frames should be learned and extracted individually, and the shared convolutional layers cannot guarantee producing proper features for all the frames. Lastly, statistics show that two-stage training strategy in Sec. IV-B improves the frame restoration quality. Because fine-tuning the decoder individually helps overcome the information loss in image quantization of video snapshot.

b) Embedding Range Selection: Here, we discuss how the embedding range, which is the number of input frames, affects our proposed method. Intuitively, larger embedding range means that more information has to be encoded and it will be more challenging. So that we launch this study by starting from encoding 5 frames while keep increasing the number of frames. Table II shows the statistics of restored frames on different embedding ranges. It is no doubt that our model performs the best when the embedding range is 5 due to the least information to be encoded. While increasing the embedding range, the PSNR and SSIM values drastically decrease. When the embedding range sets to 9, the PSNR and SSIM values reach 34.594 and 0.9493, which are still reasonable restoration qualities. However, if we increase the embedding range to 11, some apparent artifacts occur. We can conclude that to gain the best restoration quality, 5 input frames are more appropriate. If we want to embed a longer video into the video snapshot, 9 is also good enough. In this paper, the results we have shown are all from the trained model with 5 input frames for the best quality. Supplementary materials provide more results in different embedding ranges.

c) Frame Intervals Study: To test the tolerance of our method against the large motions in the testing phase, we conduct a quantitative comparison by introducing the frame interval n while sampling the testing videos. More specifically, for each $n+1$ frames, we only sample one frame. By doing this, we can obtain larger motions by just simply increasing the frame intervals. As can be seen in Table III, the PSNR and SSIM values are more or less stable even when $n=5$, which usually contain large motions. It is because that we introduced random frame intervals in the training phase. It improves the generality of our model on different motion levels. However, even the restored frames are perfect, the interpolation network may not be able to provide visually appropriate results when the motions are too large.

d) Quantitative Comparison of Difference Restored Frames: We also conduct a comparison shown in Fig. 11 on different restored frames. The reason why the restored

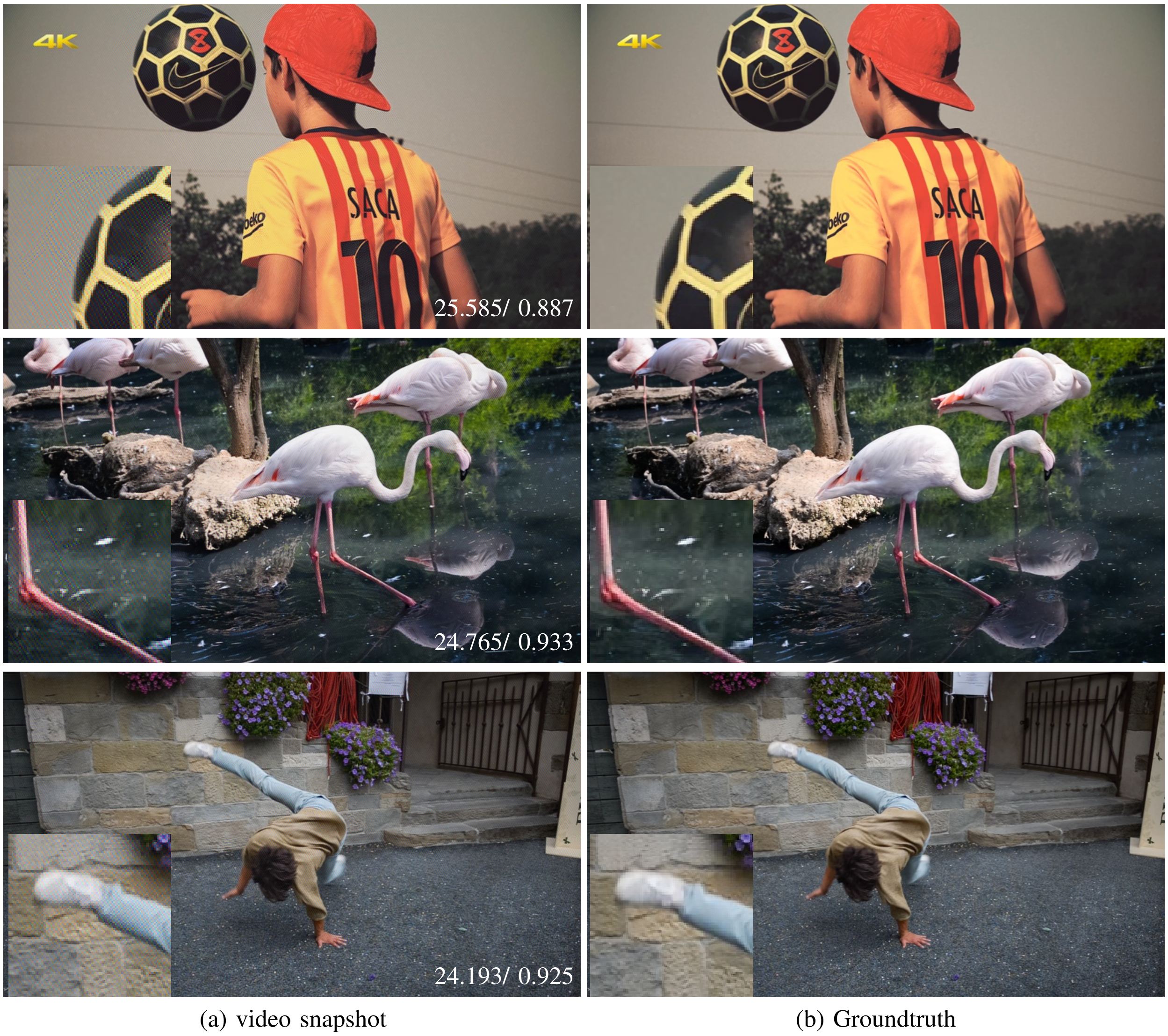


Fig. 12. Result gallery of video snapshots. Motion patterns are observable when we zoom in (a). Video snapshots are visually similar with the groundtruths (b) no matter the motions are large or small. The corresponding input frames can be found in the first row of each case in Fig. 13.

frame in the middle shows the best performance is that we use the intermediate frame of the input frames as the reference image of the embedded image. Even though, the other restored frames still show great statistical results.

D. Evaluation on the Interpolation Module

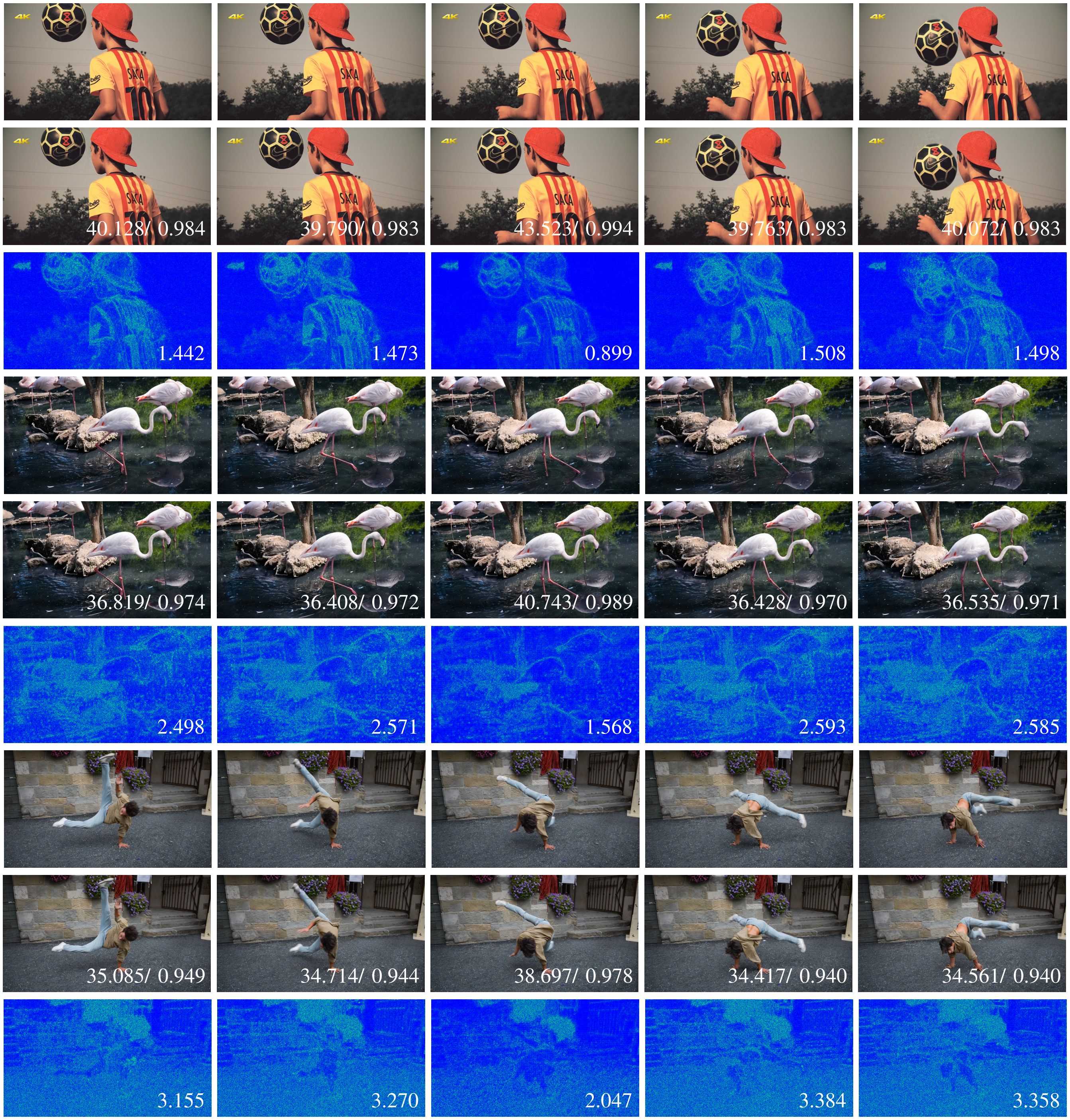
Here we examine our interpolation network by comparing to state-of-the-art interpolation methods. Table IV reports the single frame interpolation results on the UCF101 dataset [52]. Note that we adopt the motion masks provided by [48] to calculate all metrics. Our method achieves a lightly better results over the best competitor SepConv-L_1 [49]. This is because we involve warping into the process to generate multiple interpolated frames for a better optimization. Qualitative comparison is shown in Fig. 10.

E. Results Visualization

Here, we visualize the outputs of our methods, video snapshots, restored frames by the decoder, and interpolated frames, in three different scenes. Fig. 12 shows video snapshots generated by the Motion Embedding Network comparing with the groundtruth images. When we zoom in the video snapshots, regular patterns are perceptible. That is the major reason of getting low PSNR and SSIM values. However, when we zoom out, the video snapshots are quite similar to the groundtruths even the motion is large, like the dancing boy.

Fig. 13 shows the restored frames generated by the decoder in the Motion Expansion Network. In each scene, images from the first to the third rows in Fig. 13 are the groundtruth frames, restored frames and the difference maps of MAE. MAE, PSNR and SSIM values are shown on the bottom right corner of each corresponding image. We can observe that our method performs very well on spatio-temporal encoding-decoding both qualitatively and quantitatively.

Fig. 14 demonstrates the interpolated frames generated by the interpolation module in the Motion Expansion Network. Since the interpolated frames have no corresponding groundtruth frames, we only show the results. We can find that



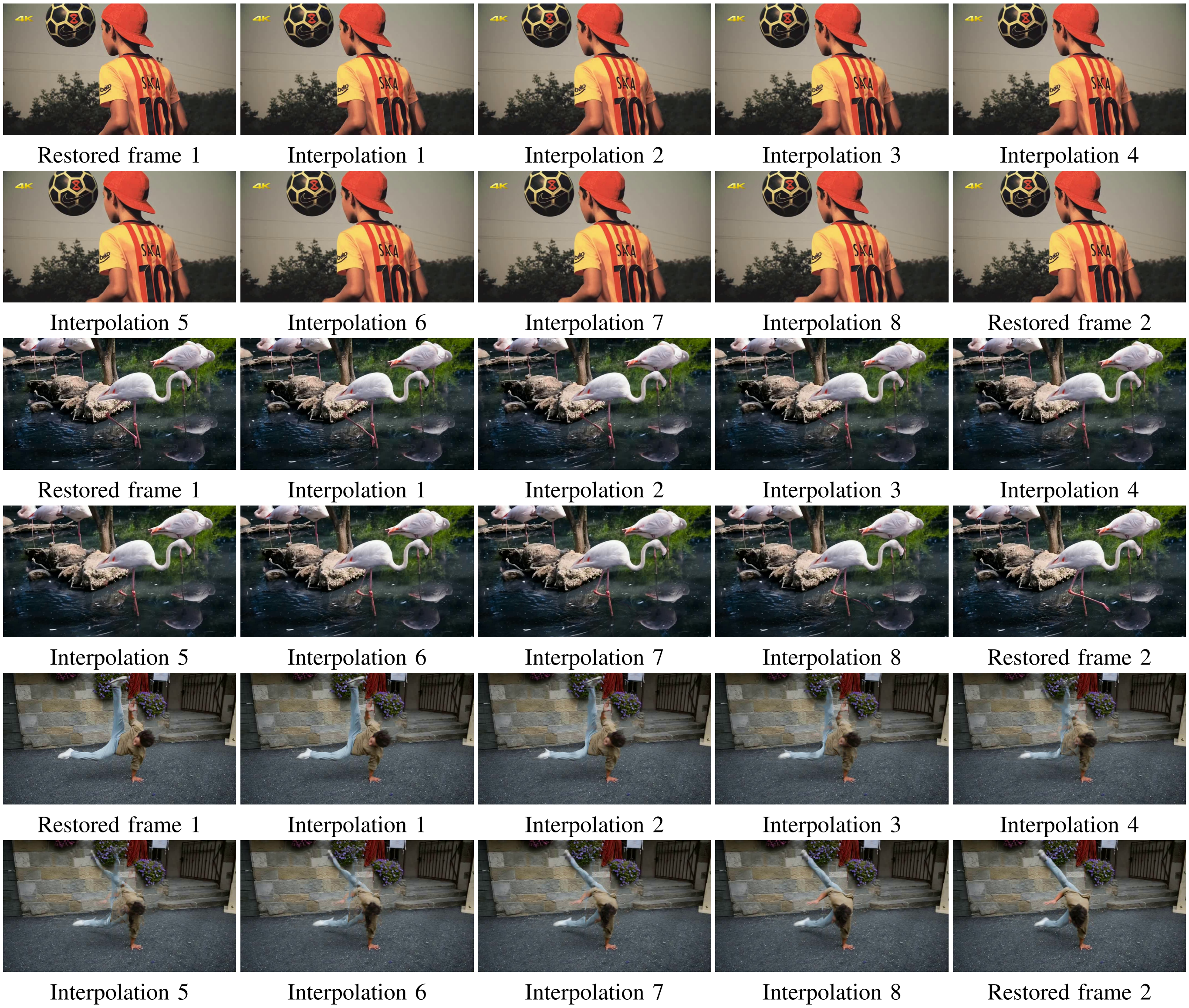
(a) Frame 1 (b) Frame 2 (c) Frame 3 (d) Frame 4 (e) Frame 5

Fig. 13. Result gallery of the restored frames from the decoder. (a) to (e) contain of the complete input frames of the video. For each scene, images in the first row to the third row are the groundtruth frames, restored frames and the difference maps. PSNR, SSIM and MAE value are shown at the bottom right corner of corresponding images.

when the motion is small, like the first scene, the interpolation module predicts great and smooth interpolation results. When the large motion and occlusion appears, like the dancing boy, the interpolated results may contain artifacts at dynamic regions. That is also the major challenge and limitation of existing multiple frame interpolation techniques. With the interpolation module, our output video is much smoother than the one directly generated by the decoder. More results in GIF format are shown in the supplementary materials.

F. User Study

We have conducted a user study on Amazon Mechanical Turk (AMT) to compare our final video results with the groundtruth videos. All the video results are generated from their corresponding video snapshot. Since our Motion Expansion Network contains a frame interpolation module. To be fair, we passed all the groundtruth videos into our proposed interpolation network with the same setting. Total 30 participants aging from 20 to 33 joined this user study


Restored frame 1 Interpolation 1 Interpolation 2 Interpolation 3 Interpolation 4

Interpolation 5 Interpolation 6 Interpolation 7 Interpolation 8 Restored frame 2

Restored frame 1 Interpolation 1 Interpolation 2 Interpolation 3 Interpolation 4

Interpolation 5 Interpolation 6 Interpolation 7 Interpolation 8 Restored frame 2

Restored frame 1 Interpolation 1 Interpolation 2 Interpolation 3 Interpolation 4

Interpolation 5 Interpolation 6 Interpolation 7 Interpolation 8 Restored frame 2

Fig. 14. Result gallery of interpolation frames. For the restored frames 1 and 2 in Fig. 13, we interpolate 8 frames between them.

and they were asked for evaluating 30 pairs of randomly selected videos. For each group of comparison, we show participants two videos, the groundtruth and our result, with random location. Participants were asked only one question: “Which video looks more natural?”. Basically, the ideal user study result should be 50/50, since users should not be able to distinguish two videos when the results are as good as the groundtruth. The user study result is shown in Fig. 15, which is 53/47 that very close to 50/50 even with the limited samples and participants. It proves that the output video from video snapshot is almost as good as the groundtruth video.

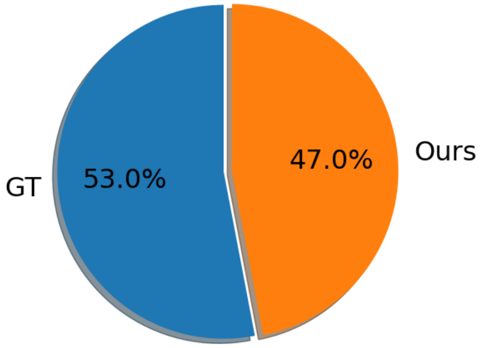
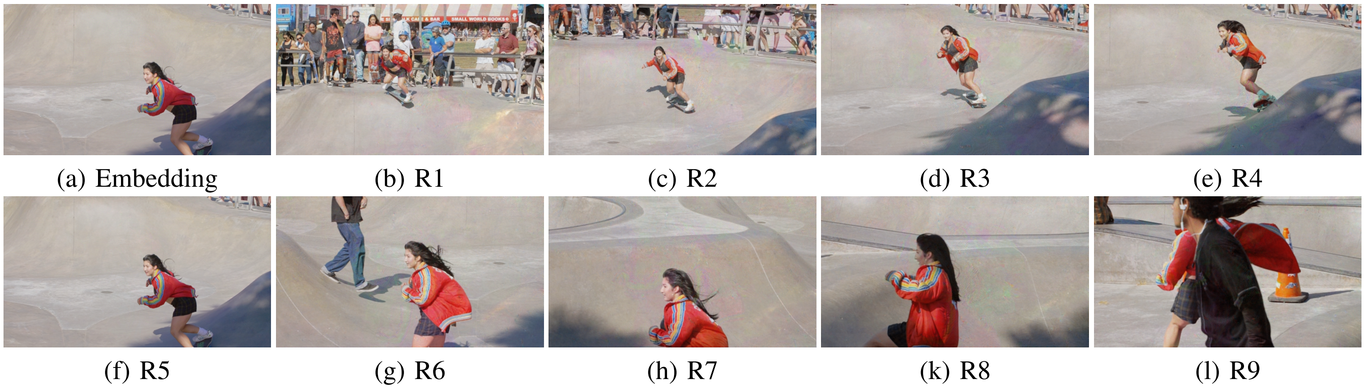


Fig. 15. User study result. Most participants cannot differentiate our restored video and groundtruth.

G. *Timing Statistics*

Here we show some timing statistics. In the training phase, the image resolution of training data was resized to 256 times 256 as needed. We trained the resized dataset on a workstation with a single Nvidia TITAN Xp GPU and Intel(R) Core(TM) i7-6900K CPU @ 3.20GHz. The whole training process takes approximately 3.5 days to obtain the final model. Table V demonstrates the timing statistics in the testing phase under three different resolutions with and without GPU. All the timing values are obtained by running the test set 10 times



(a) Embedding (b) R1 (c) R2 (d) R3 (e) R4

(f) R5 (g) R6 (h) R7 (k) R8 (l) R9

Fig. 16. Video summary example. Given an embedding image (a) that encodes 9 frames with a large interval range (100 frames), our method can restored the representative frames well (average PSNR and SSIM are 34.487 and 0.9393).

TABLE V
TIMING STATISTICS IN TESTING PHASE (IN SECONDS).

Resolution	CPU only		With GPU	
	Embedding	Expansion	Embedding	Expansion
256 × 256	0.641	51.752	0.018	0.103
512 × 512	2.409	201.481	0.057	0.293
1024 × 1024	10.978	848.515	0.219	1.143

and then calculating the average.

H. Discussion

1) Potential Applications: The proposed method can support many applications, here we discuss some of them.

Animated thumbnail. A straightforward application of our method is to produce a short, around 1 second animation by using the snap image as the thumbnail.

Animated printing. Video motion is stored in a viewable video snapshot. Therefore, another application of our invertible motion embedding is that our video snapshot can be printed on a paper, and the motion can be restored on a mobile phone.

Except producing short animations, our method can handle long video clips.

Video preview. Our proposed method can also provide an invertible video preview or video fast-forward for an arbitrary-length video by concatenating multiple consecutive video snapshots. These video snapshots can also be restored back to a longer video by expanding the video snapshot one by one through the Motion Expansion Network. However, this scheme has one limitation. When the scene changes within one video snapshot, our method might fail on this particular video snapshot and introduce artifacts due to the large motion changes and pixels disappear. Results of video preview are shown in the supplementary material.

Video summary. Handling continuous long video requires to expand multiple video snapshots. We further demonstrate discontinuous video frames (*e.g.*, interval range sets to 100 frames) can be well embedded into one video snapshot, allowing representing a video by a few representative frames. Fig. 16 shows an example of video summary. Our method can cope with a large interval range and restore discontinuous representative frames with a high fidelity (average PSNR and SSIM are 34.487 and 0.9393).

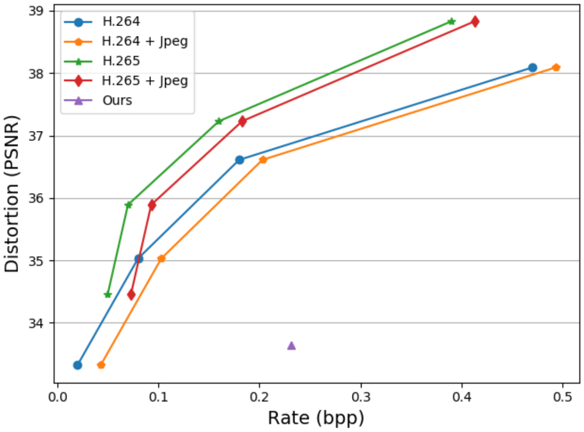


Fig. 17. Compression ratio comparison in the application of video preview.

2) Comparing to Video Compression: The proposed method is designed for embedding data instead of compressing it. Therefore, our video preview application can generate a unified viewable form for a video. Comparing to traditional video preview solution that using video compression combines with a compressed image as the thumbnail, our method does not outperform this solution in terms of compression ratio (see the results on the UVG dataset [53] in Fig. 17). However, the unique viewable and embedding nature makes our method distinct from video compression, leading to more possible applications other than video preview. For instance, video compression cannot handle the applications of animated printing, as well as the video summary because it relies on compressing continuous spatio-temporal redundancy.

3) Limitations: There are three limitations of our proposed method. First, the video snapshot can not be processed by common image processing techniques, *i.e.*, image resizing or image compression. Because those lossy post-processing may destroy the implicit pattern of the motion. Fig. 18 shows an example of the restored frame by the decoder. Fig. 18(b) & (c) are the restored frames from the video snapshot with and without JPEG compression. More artifacts can be found

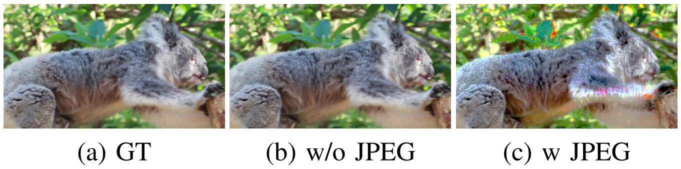


Fig. 18. Demonstration of the decoded image via JPEG compression. Here, we only select one frame from the sequence for a simple illustration. The visual difference between groundtruth image (a) and decoded without compression image (b) is imperceptible. But manifest visual difference observed with JPEG compression.

in Fig. 18 (c) than in Fig. 18(b). Secondly, when the scene changes within one video snapshot, large motion changes and pixels disappears. This scenario breaks the correlation between the frames and harms the encoding and decoding process. One more limitation is the artifacts introduced by the interpolation module when the motions are large. This is also the common limitation of existing multiple frame interpolation techniques.

V. Conclusion

In this paper, we present an invertible motion embedding and motion expansion technique. It can embed a video into one video snapshot. Later, the video snapshot can be restored back to a video again. A well designed motion attention module associated with an encoder makes valuable contribution to improve the spatio-temporal encoding. The proposed interpolation module makes the restored video longer and smoother. In addition, our work provides an new perspective on information encoding and decoding on temporal domain.

References

[1] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," Technical Report HPL-2001-191, HP Laboratory, Tech. Rep., 2001. 1, 2

[2] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM TOMM*, vol. 3, no. 1, p. 3, 2007. 1

[3] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *CVPR*, June 2016. 1, 2

[4] X. Xu, L. Wan, X. Liu, T. Wong, L. Wang, and C. Leung, "Animating animal motion from still," *ACM TOG*, vol. 27, no. 5, pp. 117:1–117:8, 2008. 1, 3

[5] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. Salesin, and R. Szeliski, "Animating pictures with stochastic motion textures," in *SIGGRAPH*, 2005. 1, 3

[6] N. Joshi, S. Mehta, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. Cohen, "Cliplets: juxtaposing still and dynamic imagery," in *ACM User Interface Software and Technology Symposium*, 2012, pp. 251–260. 1, 3

[7] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *ECCV*, 2014, pp. 540–555. 2

[8] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *ECCV*, 2014, pp. 787–802. 2

[9] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo, "Unsupervised extraction of video highlights via robust recurrent auto-encoders," in *ICCV*, 2015, pp. 4633–4641. 2

[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE TCSVT*, vol. 13, no. 7, pp. 560–576, 2003. 2

[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE TCSVT*, vol. 22, no. 12, pp. 1649–1668, 2012. 2

[12] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *ECCV*, 2018, pp. 416–431. 2

[13] Z. Chen, T. He, X. Jin, and F. Wu, "Learning for video compression," *IEEE TCSVT*, 2019. 2

[14] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An end-to-end deep video compression framework," in *CVPR*, 2019, pp. 11 006–11 015. 2

[15] Y. Horry, K. ichi Anjyo, and K. Arai, "Tour into the picture: using a spidery mesh interface to make animation from a single image," in *SIGGRAPH*, 1997. 3

[16] Y. Li, T. Wang, and H. Shum, "Motion texture: a two-level statistical model for character motion synthesis," in *SIGGRAPH*, 2002. 3

[17] C. Ma, L.-Y. Wei, B. Guo, and K. Zhou, "Motion field texture synthesis," *ACM TOG*, vol. 28, no. 5, p. 110, 2009. 3

[18] A. Sorkine-Hornung, E. Dekkers, and L. Kobbelt, "Character animation from 2d pictures and 3d motion data," *ACM TOG*, vol. 26, p. 1, 2007. 3

[19] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM TOG*, vol. 36, no. 6, pp. 196:1–196:13, nov 2017. 3

[20] M. Dvoroznák, W. Li, V. G. Kim, and D. Sýkora, "Toonsynth: example-based synthesis of hand-colored cartoon animations," *ACM TOG*, vol. 37, pp. 167:1–167:11, 2018. 3

[21] Q. Su, X. Bai, H. Fu, C.-L. Tai, and J. Wang, "Live sketch: Video-driven dynamic deformation of static drawings," in *ACM Conference on Human Factors in Computing Systems*, 2018. 3

[22] J. Geng, T. Shao, Y. Zheng, Y. Weng, and K. Zhou, "Warp-guided gans for single-photo facial animation," *ACM TOG*, vol. 37, no. 6, pp. 231:1–231:12, 2018. 3

[23] Y. Zhou, Y. Song, and T. L. Berg, "Image2gif: Generating cinemagraphs using recurrent deep q-networks," *IEEE Winter Conference on Applications of Computer Vision*, pp. 170–178, 2018. 3

[24] C.-Y. Weng, B. Curless, and I. Kemelmacher-Shlizerman, "Photo wake-up: 3d character animation from a single photo," in *CVPR*, 2019. 3

[25] F. Bogo, A. Kanazawa, C. Lassner, P. V. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *ECCV*, 2016. 3

[26] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *ICLR*, 2016. 3

[27] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv preprint arXiv:1605.08104*, 2016. 3

[28] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *NeurIPS*, 2016, pp. 64–72. 3

[29] A. Dosovitskiy and V. Koltun, "Learning to act by predicting the future," in *ICLR*, 2017. 3

[30] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," *arXiv preprint arXiv:1802.07687*, 2018. 3

[31] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d lstm: A model for video prediction and beyond," in *ICLR*, 2019. 3

[32] R. G. Van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *ICIP*, vol. 2, 1994, pp. 86–90. 3

[33] R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in *ICIP*, vol. 3, 1996, pp. 219–222. 3

[34] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *ECCV*, 2018, pp. 657–672. 3

[35] E. Wengrowski and K. Dana, "Light field messaging with deep photographic steganography," in *CVPR*, 2019, pp. 1515–1524. 3

[36] H. Yang, H. Ouyang, V. Koltun, and Q. Chen, "Hiding video in audio via reversible generative models," in *ICCV*, 2019. 3

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015. 4, 5

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 4

[39] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *CVPR*, 2018. 6

[40] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016. 5, 6

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 5, 6

[42] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, "Optimizing the latent space of generative networks," in *ICML*, 2018. 6

[43] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *CVPR*, 2018. 6

[44] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *CVPR*, 2017, pp. 1279–1288. 6

[45] F. Mahdisoltani, G. Berger, W. Gharbieh, D. J. Fleet, and R. Memisevic, "Fine-grained video classification and captioning," *CoRR*, vol. abs/1804.09235, 2018. [Online]. Available: http://arxiv.org/abs/1804.09235 7

[46] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. Sorkine-Hornung, and L. V. Gool, "The 2017 DAVIS challenge on video object segmentation," *CoRR*, vol. abs/1704.00675, 2017. [Online]. Available: http://arxiv.org/abs/1704.00675 7

[47] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016. [Online]. Available: http://arxiv.org/abs/1609.04747 7

[48] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *ICCV*, 2017, pp. 4473–4481. 8, 9

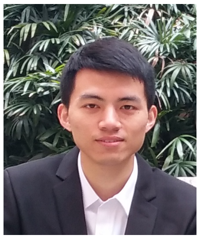
[49] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *ICCV*, 2017, pp. 261–270. 8, 9

[50] H. Jiang, D. Sun, V. Jampani, M. Yang, E. G. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *CVPR*, 2018. 8

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004. 8

[52] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402 9

[53] "Ultra video group test sequences." [Online]. Available: http://ultravideo.cs.tut.fi 12

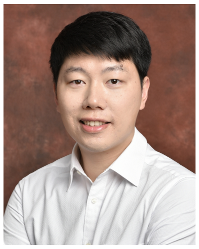


Qianshu Zhu is a master student in the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing and deep learning.

Chu Han is now a postdoctoral fellow at the Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, under the supervision of Prof. Zaiyi Liu and Prof. Changhong Liang. He received his Ph.D. degree from the Chinese University of Hong Kong, under the supervision of Prof. Tien-Tsin Wong.He received the M.Sc. degree in computer science from South China University of Technology, and the B.Sc. degree from South China Agricultural University. His current research interests include medical image analysis, computer graphics, image processing, computer vision and deep learning.

Guoqiang Han received the B.Sc. degree from the Zhejiang University, Hangzhou, China, in 1982, and the masters and Ph.D. degrees from the Sun Yat-sen University, Guangzhou, China, in 1985 and 1988, respectively. He is a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. He was the dean of the School of Computer Science and Engineering. He has published over 100 research papers. His current research interests include multimedia, computational intelligence, machine learning, and computer graphics.

Tien-Tsin Wong received the B.Sc., M.Phil., and Ph.D. degrees in computer science from The Chinese University of Hong Kong, Hong Kong, in 1992, 1994, and 1998, respectively. He is currently a Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His main research interests include computer graphics, computational manga, precomputed lighting, image-based rendering, GPU techniques, medical visualization, multimedia compression, and computer vision. He was the recipient of the IEEE Transactions on Multimedia Prize Paper Award 2005 and the Young Researcher Award 2004.



Shengfeng He is an Associate Professor in the School of Computer Science and Engineering, South China University of Technology. He was a Research Fellow at City University of Hong Kong. He obtained his B.Sc. degree and M.Sc. degree from Macau University of Science and Technology, and the Ph.D degree from City University of Hong Kong. His research interests include computer vision, image processing, computer graphics, and deep learning.