AN EMERGING CODING PARADIGM VCM: A SCALABLE CODING APPROACH BEYOND FEATURE AND SIGNAL

Sifeng Xia[†], *Kunchangtai Liang*[†], *Wenhan Yang, Ling-Yu Duan and Jiaying Liu*[*]

Peking University, Beijing, China

ABSTRACT

In this paper, we study a new problem arising from the emerging MPEG standardization effort Video Coding for Machine (VCM)[1], which aims to bridge the gap between visual feature compression and classical video coding. VCM is committed to address the requirement of compact signal representation for both machine and human vision in a more or less scalable way. To this end, we make endeavors in leveraging the strength of predictive and generative models to support advanced compression techniques for both machine and human vision tasks simultaneously, in which visual features serve as a bridge to connect signal-level and task-level compact representations in a scalable manner. Specifically, we employ a conditional deep generation network to reconstruct video frames with the guidance of learned motion pattern. By learning to extract sparse motion pattern via a predictive model, the network elegantly leverages the feature representation to generate the appearance of to-be-coded frames via a generative model, relying on the appearance of the coded key frames. Meanwhile, the sparse motion pattern is compact and highly effective for high-level vision tasks, *e.g.* action recognition. Experimental results demonstrate that our method yields much better reconstruction quality compared with the traditional video codecs (0.0063 gain in SSIM), as well as state-of-the-art action recognition performance over highly compressed videos (9.4% gain in recognition accuracy), which showcases a promising paradigm of coding signal for both human and machine vision.

Index Terms— Video coding for machine, joint feature and video compression, human vision, sparse motion pattern, frame generation

1. INTRODUCTION

Video coding aims to compress the videos into a compact form for efficient computing, transmission, and storage. Many efforts are put into this domain, and over the last three decades, four coding standards are built to significantly improve the coding efficiency. The latest video codecs, *i.e.* MPEG-4 AVC/H.264 [1] and High Efficiency Video Coding (HEVC) [2] seek to improve the video coding performance by edging out spatial, temporal and coding redundancies of video frames. In the past few years, data-driven methods have been popular and bring in tremendous progress in the compression task. The latest data-driven methods have largely overpassed performance of the state-of-the-art codecs, *e.g.* HEVC by further improving various kinds of modules like intra-prediction [3], inter-prediction [4, 5], loop filter [6, 7]. These techniques significantly improve the video quality from the perspective of the signal fidelity and human vision.

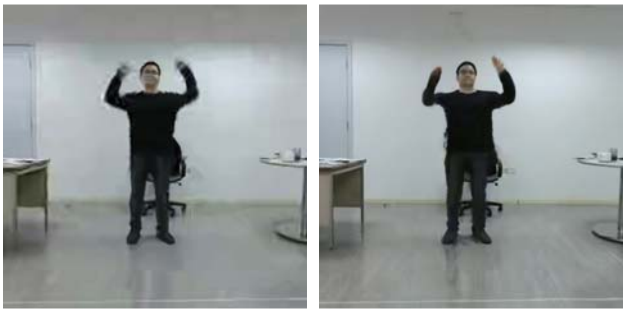


Fig. 1. The visual results of the reconstructed videos by HEVC (left panel) and our method (right panel). Embedded videos are best viewed in Acrobat Reader.

Existing coding techniques run into problems when encountering big data and video analytics. The massive data streaming generated everyday from the smart cities needs to be compressed, transmitted and analyzed to provide high valuable information, such as the results of action recognition, event detection. Given this scenario, it is expensive to perform the analysis on the compressed videos, as the video coding bit-stream is redundant and existing coding mechanism is not flexible to discard the information that is unrelated to analytical tasks [8]. Therefore, in the context of big data, it is still an open problem to perform the scalable video coding, where the requirement of machine vision is first met and additional bitrates can be utilized to further improve visual quality of the reconstructed video progressively and incrementally. It is an urgent need to obtain a scalable feature representation that connects the information of low and high-level vision and

[†]Equal contribution. This work was supported by National Natural Science Foundation of China under contract No.61772043, Beijing Natural Science Foundation under contract No.L182002, and the National Key R&D Program of China under Grand No.2018AAA0102700. [*]Corresponding author.

[1]https://lists.aau.at/mailman/listinfo/mpeg-vcm

978-1-7281-1331-9/20/$31.00 ©2020 IEEE

switches the forms between two purposes freely.

The success of deep learning models has opened a new door. The deep analytic models can extract compact and high-valuable representations, which can convert the redundant pixel domain information into the sparse feature domain. In contrast, deep generative models are responsible to produce the whole images and videos with only the guidance of highly abstracted and compact features. Supported by these tools, we can realize the scalable compression of videos and features jointly, which is close to both practical application demands in the big data context and accords with the mechanism of human brain circuits. The most compact and valuable abstracted features are first extracted via deep analytic models [9, 10, 11] to support the analytics applications. With these features, we can locate the place and time where some key events happen, namely rethinking rough situations. Then, guided by the features, other information is partly generated by deep generative models [12, 13, 14, 15], and partly compressed and decoded to support the video reconstruction, namely rethinking scene details. This solution is potential to address the difficulty in combining video analytics and reconstruction in the big data streaming, which is the main target of video coding for machine (VCM). The first step of the process can provide timely analytical results with a small portion of bitrates to fulfill the need of machine vision and the second stage can further provide the reconstructed videos with regards to the analytical results using more bitrates to meet the need of human vision [16].

Specifically, in this paper, we propose a scalable joint compression method for both features and videos in surveillance scenes, where a learnable motion pattern bridges the gap between machine and human vision. The sparse motion pattern is first extracted automatically via a deep predictive model. After that, the appearance of the currently coded frame is transfered from the coded key frame with the guidance of the motion pattern via a deep generative model. The sparse motion pattern is highly efficient for high-level vision tasks, *e.g.* action recognition, and it can also meet the requirement of human vision. In this way, the total coding cost of features and videos can be largely reduced.

In summary, the contributions of our paper are summarized as follows:

- To the best of our knowledge, we make the first attempt towards VCM to compress features and videos jointly, serving for both machine and human vision. A novel scalable compression framework is designed with the aid of predictive and generative models to support both machine and human vision.
- In our framework, the learned sparse motion pattern is used as a bridge, which is flexible and largely reduces the total coding cost of two kinds of vision. To promote the analysis performance of human action recognition, we additionally apply the constraint of the learned points with the guidance of human skeletons.
- Compared with traditional video codecs, our method not only achieves much better video quality but also offers significantly better action recognition performance at very low bitrates, which showcases a promising paradigm of coding signal for both human and machine vision.

The rest of the article is organized as follows. Sec. 2 illustrates the pipeline of our proposed joint feature and video compression. The detailed network architecture for key point prediction and motion guided target video generation is also elaborated. Experimental results are shown in Sec. 3 and concluding remarks are given in Sec. 4.

2. JOINT COMPRESSION OF FEATURES AND VIDEOS

Given a video sequence $\mathbf{I} = \{I_1, I_2, ..., I_N\}$ where N indicates the frame number, it is necessary to compress $\mathbf{I}$ for transmission and storage. In this section, we will first analyze limitations of traditional video coding methods. Then, we develop our new framework to compress features and videos jointly in a scalable way.

2.1. Sequential Compression and Analytics

The traditional video codec targets to optimize the visual quality of the compressed video from the perspective of signal fidelity. In this process, all frames are coded. For each frame, spatial and temporal predictions are utilized to predict the target frame with existing coded frames to remove the spatial and temporal redundancy. Then, the prediction residue and much syntax information are coded for reconstruction at the decoder side. Though the data can be efficiently compressed via the latest codecs, the scale of data is still massive as a huge amount of data is taken all days and weeks. Therefore, it is intractable to compress and save data with a high quality, and analyze it later.

It is a reasonable trade-off to compress the data into a low-quality format. However, existing compression methods which target at optimizing the human vision are not desirable for high-level analytics tasks. If we lower the quality of the compressed videos, the performance of action recognition will be largely degraded, As demonstrated in Sec. 3.2, our method uses only about 1/3 bitrate cost of the traditional compression method to achieve a better performance in the action recognition task. Another path that leads to effective video analytics is to extract and compress features. However, in this case, we could not obtain the reconstructed videos. This also sets barriers to real applications, where the results usually need to be confirmed by human examiners. Therefore, we seek to develop a flexible and scalable framework which compresses the feature at first for machine vision and reconstructs the video later for human vision with more bits consumption.

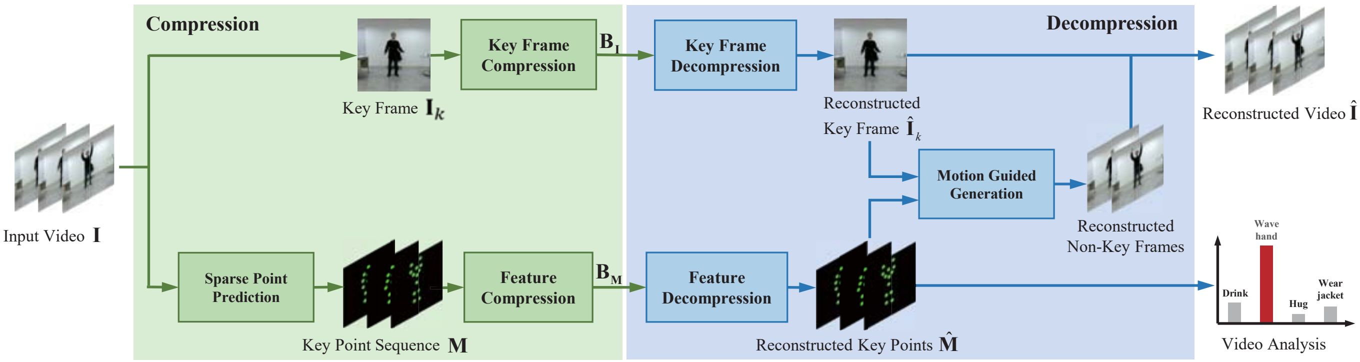


Fig. 2. The coding pipeline of our proposed joint feature and video compression that serves for both human and machine vision.

2.2. An Overview of Joint Feature and Video Compression

Fig. 2 has illustrated the overview pipeline of the proposed joint feature and video compression method. The motivation lies in the fact that in surveillance scenes, the videos can be represented as a background layer (static or slow moving) and moving objects, such as human bodies. Then, the network is capable of learning to represent a video sequence with the learned sparse motion pattern, which can indicate the object motion among frames. In our work, we focus on indoor surveillance videos with a static background and moving humans.

At the encoder side, with the captured video frames $\mathbf{I} = \{I_1, I_2, ..., I_N\}$, a set of key frames $\mathbf{I}_k$ will be first selected and compressed with traditional video codecs and form the bit-stream $\mathbf{B}_\mathbf{I}$. The coded key frames convey the appearance information which includes the background and human appearances and will be transmitted to the decoder side to synthesize the non-key frames. Moreover, the learned Sparse Point Prediction Network (SPPN) extracts sparse key points from video frames and form a point sequence $\mathbf{M} = \{m_1, m_2, ..., m_N\}$. The sparse point sequence can mark the motion areas in the frames and convey the motion trajectories of objects along the temporal dimension, which is viewed as a sparse motion pattern of the video. The point sequence will also be coded to a bit stream $\mathbf{B}_\mathbf{M}$ for transmission.

At the decoder side, key frames will be first reconstructed from $\mathbf{B}_\mathbf{I}$ and we indicate the reconstructed key frames as $\hat{\mathbf{I}}_k$. For reconstructing remaining non-key frames, the key points are decompressed as $\hat{\mathbf{M}} = \{\hat{m}_1, \hat{m}_2, ..., \hat{m}_N\}$ and a learned Motion Guided Generation Network (MGGN) will first estimate the motion flow among frames based on the decompressed sparse motion pattern. Then, MGGN transfers the appearance of the reconstructed key frames to remaining non-key frames with the guidance of the estimated motion flow. Specifically, for the t-th frame to be reconstructed, we denote its previous key frame as $\hat{I}_k$. The target frame is synthesized as $\hat{I}_t = \varphi(\hat{I}_k, \hat{m}_k, \hat{m}_t)$, where φ represents MGGN. Finally, the reconstructed key points $\hat{\mathbf{M}}$ and the video $\hat{\mathbf{I}} = \{\hat{I}_1, \hat{I}_2, ..., \hat{I}_N\}$ can be used respectively for machine analysis and human vision.

2.3. Detailed Network Architecture Illustration

The critical feature of our joint feature and video compression framework needs to be capable of capturing the motion between video frames for both machine analytics and video reconstruction. There are several kinds of ways to model video motion, such as dense optical flow [17] or sparse motion representations based on human poses [14] or unsupervisely learned key points [15]. In our work, we hope the motion representations to be sparse enough for efficient machine analytics. Therefore, we refer to [15] to predict key points of frames as the sparse motion pattern, which is compact enough that costs only a few bits for transmission and storage. For human vision, motion flow among video frames will be later derived from the sparse motion pattern to guide the generation of the target frame.

The framework of the network is shown in Fig. 3. For a key frame I_k and a target frame I_t which is to be generated at the decoder side, their key points will be first predicted by SPPN, and this sparse motion pattern is later combined with I_k for estimating the flow map between frames. Then, the generated flow map will guide the transfer of the appearance of I_k to the target frame. Details of different parts of the network are described as follows.

Sparse Point Prediction. For an input frame, a sub-network of the U-Net architecture followed by softmax activations is used to extract L heatmaps $\mathbf{H} = \{H_1, ...H_L\}$ for key point prediction. Each heatmap $H_l \in [0, 1]^{\mathrm{H}\times\mathrm{W}}$ corresponds to one key point position p_l, which is estimated as follows:

$$p_l = \sum_{p\in\Omega} H_l\left[p\right] p, \tag{1}$$

where Ω is the set of positions of all pixels. Besides the key point position, the corresponding covariance matrix Σ_l is defined as:

$$\Sigma_l = \sum_{p\in\Omega} H_l\left[p\right] \left(p - p_l\right) \left(p - p_l\right)^{\mathrm{T}}. \tag{2}$$

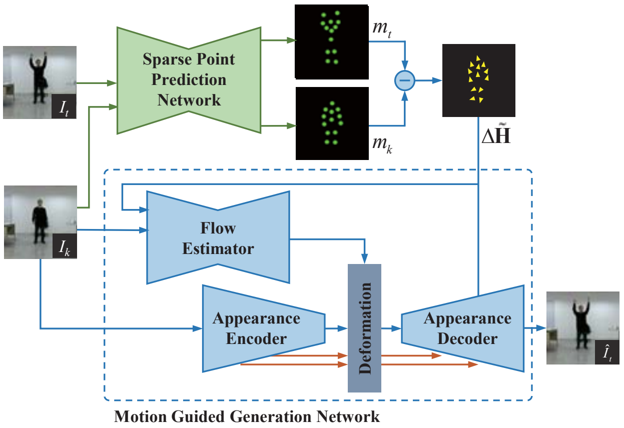


Fig. 3. Framework of our proposed joint feature and video compression, including a sparse point prediction network and motion guided generation network to extract the sparse motion pattern and generate the target frame.

The covariance matrix is generated here because it can additionally capture the correlations between the key point and its neighbor pixels. Consequently, for each key point, totally 6 float numbers including two numbers indicating the position and 4 numbers in the covariance matrix are used for description.

For the succeeding usage, the key point description will be used to generate new heatmaps by a Gaussian-like function. This operation is done for that the new heatmaps are more compatible with convolutional operations. Specifically, the new heatmap $\tilde{H}_l$ will be generated as follows:

$$\tilde{H}_l\,[p] = \exp\left(-\alpha(p-p_l)^{\mathrm{T}}\Sigma_l^{-1}\,(p-p_l)\right), \qquad (3)$$

where α is a normalization constant and set to 0.5. After this progress, two sets of newly generated heatmaps $\tilde{\mathbf{H}}^k = \left\{\tilde{H}_1^k, ..., \tilde{H}_L^k\right\}$ and $\tilde{\mathbf{H}}^t = \left\{\tilde{H}_1^t, ..., \tilde{H}_L^t\right\}$ are generated from frames I_k and I_t, respectively.

Motion Flow Estimation. With the estimated key points and newly generated heatmaps, a sub-network in MGGN will be first used to estimate the motion flow between frames I_k and I_t. The source frame I_k is adopted to form the input for it conveys the appearance information. Meanwhile, the difference heatmaps $\Delta\tilde{\mathbf{H}} = \tilde{\mathbf{H}}^t - \tilde{\mathbf{H}}^k$ between two frames are used to form the input to provide sparse motion information. The flow estimator will finally output a flow map $\xi_{k\rightarrow t}$.

Motion Guided Target Frame Generation. The target frame is generated with a sub-network of the U-Net architecture. Feature maps of different sizes are extracted by the appearance encoder and will be bypassed to the appearance decoder for feature fusion. In order to align the features to the target frame, features will be previously deformed with the estimated flow map $\xi_{k\rightarrow t}$ before fusion. Besides, the difference heatmaps $\Delta\tilde{\mathbf{H}} = \tilde{\mathbf{H}}^t - \tilde{\mathbf{H}}^k$ is used as side information that is inputted to the appearance decoder. Then, the target frame $\hat{I}_t$ can be generated by the appearance decoder.

Skeleton Guided Point Prediction Loss Function. In [15], the key points prediction is learned unsupervisely. In our work, we additionally use human skeleton information to guide the key point prediction. The skeleton information is used for its high efficiency in modeling human actions as the skeleton points cover many human joints, which are highly correlated to human actions. Consequently, the PKU-MMD dataset [18] is used in our work for training and testing, which is a large-scale dataset and contains many human action videos. More importantly, human skeletons are available in this dataset for each human body in the videos.

We sample 16 skeleton points for each human body and employ an L_1 loss function for supervision. The key point detection loss function is defined as follows:

$$\mathcal{L}_{\text{point}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{16} \parallel p_l^i - \pi_l^i \parallel_1, \qquad (4)$$

where π_l^i represents the l-th skeleton point of the human in the i-th training sample.

Overall Loss Function. Besides the point prediction loss, a combination of an adversarial and the feature matching loss proposed in [19] are used for training. The discriminator D will take $\tilde{\mathbf{H}}^t$ concatenated with either the real image I_t or the generated image $\hat{I}_t$ as its input. The discriminator and generator losses are calculated as follows:

$$\mathcal{L}_D = \mathrm{E}_{I_t}[(D(I_t, \tilde{\mathbf{H}}^t) - 1)^2] + E_{(I_t,\hat{I}_t)}[(D(\hat{I}_t, \tilde{\mathbf{H}}^t))^2], \quad (5)$$

$$\mathcal{L}_G = E_{(I_t,\hat{I}_t)}[(D(\hat{I}_t, \tilde{\mathbf{H}}^t) - 1)^2]. \qquad (6)$$

For a better reconstruction quality, a reconstruction loss function $\mathcal{L}_{rec}$ is built to keep I_t and $\hat{I}_t$ to have similar feature representations. $\mathcal{L}_{rec}$ is implemented by calculating the L_1 distance between features extracted from I_t and $\hat{I}_t$ by the discriminator. Features outputted by all layers of the discriminator are all used for calculation.

The final loss function is calculated by $\mathcal{L} = \lambda_{point}\mathcal{L}_{\text{point}} + \lambda_{rec}\mathcal{L}_{\text{rec}} + \mathcal{L}_G$, where λ_{point} and λ_{rec} are respectively set to 20 and 10.

3. EXPERIMENTS

3.1. Experimental Details

PKU-MMD dataset [18] is used to generate the training and testing samples. In total 3317 clips with 32 frames are sampled for training and 227 clips with 32 frames are sampled for testing. All frames are cropped and resized to 512 × 512 during sampling. The skeleton information is also used during the training process. 16 skeleton points are chosen for each frame and mapped to the corresponding two-dimensional s-pace to generate the labels for key point prediction. The network is implemented in PyTorch and the Adam optimizer [20] is used for training. For each training sample, we randomly select two frames from a clip to form it.

In the testing process, we consistently use the first frame in each clip as the key frame. At the encoder side, the key frame is coded with the HEVC codec in the constant rate factor mode. The constant rate factor is set to 32. Besides the key frame, key points of all frames in the clip are predicted by SPPN and compressed for transmission. As mentioned in Sec. 2.3, each key point contains 6 float numbers. For the two position numbers, a quantization with the step 2 is performed for compression. For the other 4 float numbers belonging to the covariance matrix, we calculate the inverse of the matrix in advance, and then quantize the 4 values with a step 64. Then, the quantized key point values are further losslessly compressed by the Lempel Ziv Markov chain algorithm (LZMA) algorithm [21]. At the decoder side, the compressed key frame and points are decompressed and used to generate remaining frames.

To verify the efficiency of our coding paradigm, we use HEVC as the anchor for comparison by additionally compressing all frames with the HEVC codec. The constant rate factor is firstly consistently set to 51, the highest compression ratio. Then, the recognition accuracies of using the learned sparse motion pattern and the compressed videos are compared. To verify the reconstruction quality, we set the constant rate factor to 42 and compare the reconstruction results between HEVC and our method with similar coding cost. The reconstruction quality is compared both quantitatively and qualitatively.

3.2. Action Recognition Accuracy

We identify the efficiency of the learned key points for high-level analytics tasks in the action recognition task. Although there are 6 numbers for each key point, we only use two quantized position numbers for action recognition. Consequently, only bits of the compressed position numbers are considered for calculating the bitrate cost of feature-based action recognition. To align to the bitrate cost of the features, we firstly resize all clips to the size of $256 * 256$ and then use the constant rate factor 51 to compress the testing clips with HEVC.

Table 1. Action recognition accuracy of different methods and corresponding bitrate costs.

Input	Bitrate (Kbps)	Accuracy(%)
Compressed Video	16.2	65.2
Compressed Key Point	5.2	74.6

Table 1 has shown the action recognition accuracy and corresponding bitrate costs of different kinds of data. Our method can obtain considerable action recognition accuracy with only 5.2 Kbps bitrate cost. Although we have chosen the worst coding quality, it still needs 16.2 Kbps to transform and store the compressed videos. More bitrates cannot bring too much performance improvement in action recognition on compressed videos. Unfortunately, the recognition accuracy even drops by 9.4%.

Table 2. SSIM comparison between different methods and corresponding bitrate costs.

Codec	Bitrate (Kbps)	SSIM
HEVC	33.0	0.9008
Ours	32.1	0.9071

3.3. Video Reconstruction Quality

The video reconstruction quality of the proposed method is also compared with that of HEVC. During the testing phase, we compress the key frames with the constant rate factor 32 to maintain a high appearance quality. The bitrate is calculated by jointly considering the compressed key frames and key points. As for HEVC, we compress all frames with the constant rate factor 44 to achieve an approaching bitrate cost.

Table 2 has shown the quantitative reconstruction quality of different methods. SSIM values are adopted for quantitative comparison. It can be observed that, our method can achieve better reconstruction quality than HEVC with a fewer bitrate cost. Subjective results of different methods are shown in Fig. 4. There are obvious compression artifacts on the reconstruction results of HEVC, which heavily degrade the visual quality. Compared with HEVC, our method can provide far more visually pleasing results.

4. CONCLUSION

In our work, we propose a novel framework to bridge the gap between compression for features and videos. A conditional deep generation network is designed to reconstruct video frames with the guidance of a learned sparse motion pattern. This representation is highly compact and also effective for high-level vision tasks, *e.g.* action recognition. Therefore, it is scalable to meet the requirements of both machine and human vision, which reduces the total coding cost. Experimental results demonstrate that our method can obtain superior reconstruction quality and action recognition accuracy with fewer bitrate costs compared with traditional video codecs.

5. REFERENCES

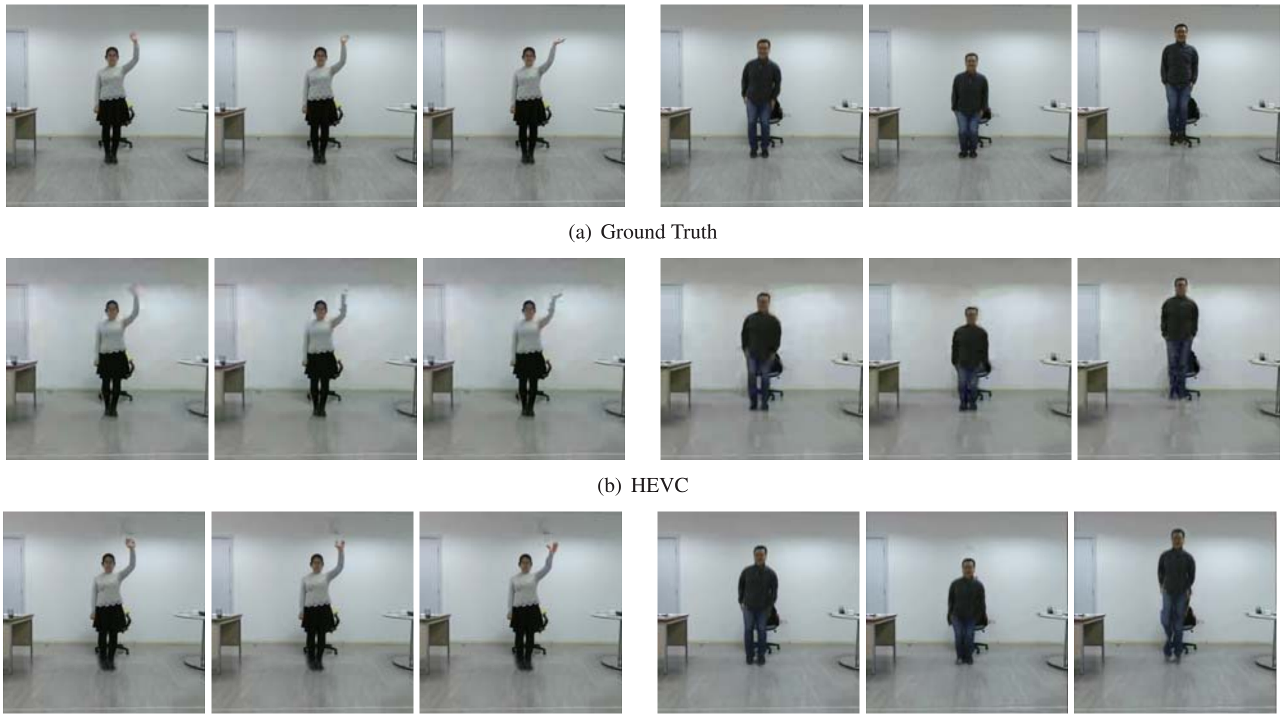
[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H. 264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] Y. Hu, W. Yang, M. Li, and J. Liu, “Progressive spatial recurrent neural network for intra prediction,” *IEEE Transactions on Multimedia*, pp. 1–1, 2019.

[4] S. Xia, W. Yang, Y. Hu, and J. Liu, “Deep inter prediction via pixel-wise motion oriented reference generation,” in *Proc. IEEE Int’l Conf. Image Processing*, 2019.

[5] J. Liu, S. Xia, W. Yang, M. Li, and D. Liu, “One-for-all: Grouped variation network-based fractional interpolation in



(a) Ground Truth

(b) HEVC

(c) Proposed

Fig. 4. Video reconstruction results of different methods. Left and right three panels correspond to two video clips in the testing set, respectively. The average SSIM values of the reconstructed clips are respectively 0.8889 and 0.9204 for HEVC and the proposed method for the left clip. For the right clip, the SSIM values of HEVC and the proposed method are 0.8966 and 0.9143.

video coding," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2140–2151, 2019.

[6] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma, "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE Transactions on Image Processing*, 2019.

[7] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *IEEE Image, Video, and Multidimensional Signal Processing Workshop*, 2016, pp. 1–5.

[8] L. Ding, Y. Tian, H. Fan, Y. Wang, and T. Huang, "Rate-performance-loss optimization for inter-frame deep feature coding from videos," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5743–5757, Dec 2017.

[9] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks," in *Proc. of the Thirtieth AAAI Conf. on Artificial Intelligence*, 2016, pp. 3697–3703.

[10] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. of the Thirtieth AAAI Conf. on Artificial Intelligence*, 2017, pp. 4263–4270.

[11] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for skeleton-based human action understanding," in *Proc. of the Workshop on Visual Analysis in Smart and Connected Communities*, 2017, pp. 1–8.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Annual Conference on Neural Information Processing Systems*, 2014.

[13] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin, "Pose guided human video generation," in *Proc. European Conf. Computer Vision*, 2018.

[14] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proc. IEEE Int'l Conf. Computer Vision*, 2019.

[15] A. Siarohin, S. Lathuilire, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2019.

[16] L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: A paradigm of collaborative compression and intelligent analytics," in *arXiv:2001.03569*, 2020.

[17] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018.

[18] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for skeleton-based human action understanding," in *Proc. ACM Int'l Conf. Multimedia*, 2017.

[19] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2017.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int'l Conf. Learning Representations*, 2015.

[21] I. Pavlov, "Lempel Ziv Markov chain algorithm," in *http://en.wikipedia.org/wiki/LZMA*.