ADVANCED CNN BASED MOTION COMPENSATION FRACTIONAL INTERPOLATION

Han Zhang [*,†], *Li Li* [‡], *Li Song* [*,†], *Xiaokang Yang* [*,†], *Zhu Li* [‡]

[*]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[†]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
[‡]University of Missouri-Kansas City

ABSTRACT

Fractional-sample precision motion compensation has been widely adopted in a series of video coding standards to further improve compression efficiency. Usually, signal decomposition based interpolation filters are used to generate fractional samples from integer pixels. However, the coefficients of these finite impluse response filters may not be suitable for varied video contents and coding conditions because of the assumption when designing these filters. In this paper, we regard the fractional interpolation process as an image generation task, which utilizes the real interger position samples at the reference block to predict and generate fractional samples that are much closer to current coding block. We use the convolutional neural netwok (CNN) as the generator. Moreover, to make the best of CNN's powerful nonlinear learning ability, instead of inputting the reference block directly, we separately input the corresponding prediction and residual parts of reference block. The proposed dual-input CNN-based interpolation scheme has been incorporated into the HEVC framework and experimental results demonstrate our approach achieves average 0.9% bitrate reduction.

Index Terms— Video Coding, Convolutional Neural Network, Fractional Interpolation

1. INTRODUCTION

Among a series of block-based hybrid video coding frameworks, inter prediction is always the key technology of improving compression efficiency by removing temporal redundancy. The basic idea of block-based inter prediction is searching the best matching block for current coding block in reference frames (previously encoded frames) at encoder and transmitting the difference between them (termed *residual*) to decoder. The relative positional relationship between current block and its corresponding reference block indicates the movement of objects in this block, which is described by motion vector (MV). The digital video sequence is discrete due to the sampling process at the capture stage, thus the real continuous motion of moving objects may not exactly follow the sampling grid [1]. To further decrease *residual*, fractional-precision motion vector has been utilized. Since the samples pointed to by the fractional motion vector are not really exist in the reference frame, they all have to be generated by interpolation process in video coding.

The traditional interpolation method has been changing with the evolution of the video coding standards. In H.264/AVC [2], a 6-tap filter is applied to derive the half-pel samples while the quarter-pel samples are generated by simply averaging two neighboring samples. A more advanced approach named DCT-based interpolation filter (DCTIF) is adopted in the latest standard – High Efficiency Video Coding (HEVC) [3]. The DCTIF contains a symmetric 8-tap filter for half-pel samples interpolation and an asymmetric 7-tap filter for quarter-pel samples in luma component. In addition to these fixed coefficient filters, Wddi *et al.* proposed an adaptive interpolation filter method in [4], the coefficients of which are estimated for each frame. A comprehensive study about the effects of fractional sample accuracy on the inter prediction is shown in [5].

Recently, deep learning based methods have achieved great success in some pixel-level tasks. A convolutional neural network based single image super-resolution method is proposed by Dong *et al.*, which learns an end-to-end mapping between low- and high- resolution images [6]. A video frame interpolation network with fully convolutional layer is proposed by Niklaus *et al.* [7]. Inspired by these, some work try to explore the potential for applying deep learning to the video coding task. Dai *et al.* [8] proposed a CNN-based post-processing method for HEVC and showed superior performance. Zhao *et al.* [9] presented a novel bi-directional motion compensation with CNN for improving the prediction accuracy. With respect to the fractional interpolation task mentioned above, Yan *et al.* [10] introduced a CNN-based interpolation method with a special training data preparation step to generate the half-pel samples. Our previous work [11] also presented a CNN-based interpolation method with some constrain strategies during the training phase. Both of these two learning-based fractional interpolation methods are use the reference frame as input.

In this paper, we present a dual-input convolutional neural network based fractional interpolation approach for HEVC inter prediction. The purpose of fractional interpolation in HEVC inter prediction is to generate more accurate predic-

tion value for current coding block, and the fractional samples do not really exist in the reference frame. Instead of treating the interpolation process as a task similar to super-resolution like before, we regard this process as image generation. Thus, we utilize our interpolation method to generate prediction of current coding block from reference frame directly in a block-by-block manner, rather than generating an enlarged reference frame. Further, because the reference frames in video coding are just results from simple addition of its corresponding prediction part and residual part, sometimes followed by clip operation, we propose to input the prediction and residual parts separately instead of simply using the reference block as before [10],[11]. This dual-input scheme can make better use of different frequency information and the nonlinear way to recombine them by convolution operation in CNN is also more flexible than linear addition.

The rest of this paper is organized as follows: Section 2 presents our proposed method in detail. Experimental results compared with HEVC baseline are shown in Section 3. And Section 4 concludes this paper.

2. THE PROPOSED CNN-BASED INTERPOLATION METHOD

As illustrated in Fig.1, the real integer pixel values at the reference frame are represented by blue capitalized *A*, while other fractional samples described with lowercase letters are not real existed and have to be interpolated with DCT-based filters in HEVC. In this paper, we propose a dual-input CNN-based method to perform this interpolation process. In our approach, instead of interpolating the whole reference frame to a large one, the interpolation is processed in block level. We predict and generate the fractional sample block which has the same size as the reference block directly.

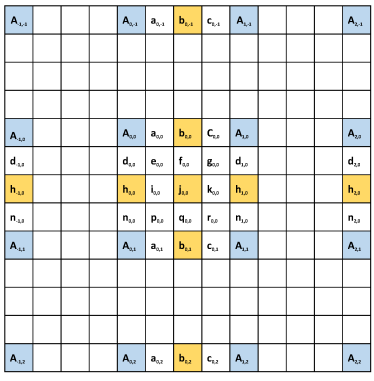


Fig. 1. Example of Integer and Fractional samples in HEVC luma component

2.1. Proposed Network Architecture

In this paper, we propose a six-layer convolutional neural network with two inputs to perform the fractional interpolation process. The network structure is depicted in Fig.2. In order to extract information from both prediction and residual parts of the relative reference block, we use separate convolution layers to handle them. After several convolution layers, the feature maps of prediction and residual channels are concatenated to form the input of following layers. The following convolution operation gets nonlinear combination of prediction and residual instead of the simple linear addition as in HEVC. Our network is a fully convolutional network, and the rectified linear unit (ReLU) [12] is adopted as the activation function. Thus the first five layers can be formulated as

$$F\left(X_i\right) = max\left(0, W_i * X_i + b_i\right) \tag{1}$$

where $F\left(X_i\right)$ and X_i represent the output and input of the i-th layer. W_i and b_i are the weight and bias of the i-th layer respectively. In our network, each layer except the last one consists of 64 filters with size 3×3. The last layer used to combine previous feature map and generate output contains a single 3×3 filter.

We also adopt residual learning strategy in [13], which just learns the difference between input and output. The residual learning strategy proves to enable faster converge and sometimes shows superior performance. In residual learning scheme, the input of network need to be added to the output of the last layer so as to generate the final output. Considering the sum of our two input prediciton and residual is the reference value which is used to generate the fractional samples in original HEVC, we add these two input together with the output of 6-th layer to get the predicted fractional samples.

2.2. Training

In order to improve the objective quality of generated fractional samples, Euclidean loss function is used during the training phase. Given a training dataset $\left\{X^i, Y^i\right\}_{i=1}^N$, the Euclidean loss can be formulated as

$$L\left(W, b\right) = \frac{1}{N}\sum_{i=1}^{N}\left\|F\left(X^i; W, b\right) - Y^i\right\|^2 \tag{2}$$

where $F\left(X^i; W, b\right)$ is the final output of our network and Y^i is the original block label, W, b is the parameter set to be learned. Due to the residual learning strategy, the final output is :

$$F\left(X^i; W, b\right) = \left(W_6 * X_6^i + b_6\right) + X_{pred}^i + X_{resi}^i \tag{3}$$

where $\left(W_6 * X_6^i + b_6\right)$ is output of the sixth layer, X_{pred}^i and X_{resi}^i are the input prediction and residual parts respectively. In this paper, $L\left(W, b\right)$ is optimized with stochastic gradient descent.

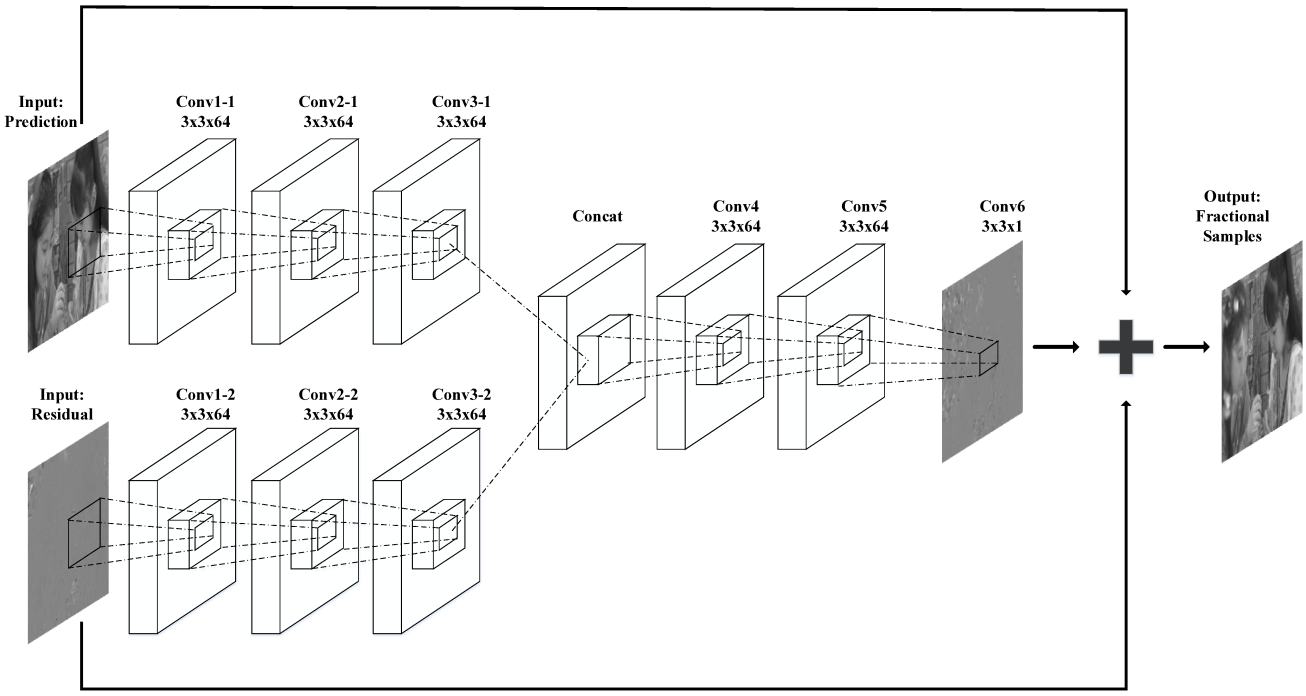


Fig. 2. Architecture of our proposed dual-input network for fractional interpolation

2.3. Training Data Preparation

Since our network is designed to predict the fractional samples for current coding block in a block-by-block manner instead of generating them for the whole reference frame, our training data preparation process is different from previous work. The training data can be derived at the decoder side of HEVC directly. Firstly, all the fractional motion vectors are recorded. Because these motion vectors' relative reference blocks are located in fractional positions, which are not really existed and need to be generated using interpolation filters from integer pixel value. The corresponding integer locations of these fractional reference blocks are calculated according to the fractional value of motion vectors at the second stage. These integer position blocks are termed as real reference block in our method. After we get the integer position of reference block, it's prediciton and residual parts are extracted respectively as the input of our network. Since the original value of current coding block is required as the label of network, the spatial locations of current block are recorded and used to crop label blocks from original sequences. In order to generate output with the same size of input block, zero padding operation is applied for all convolution layers except the first one. Because the input prediction and residual block of our network are usually very small, we use the neighboring real pixel as padding value in the first layer. The prediction and residual blocks we extracted from decoder side already contain padding margin.

It should be noticed in our proposed scheme, we train different CNN models for different fractional positions. In order to achieve this, the training input prediciton and residual data is generated according to the fractional value of relative motion vector.

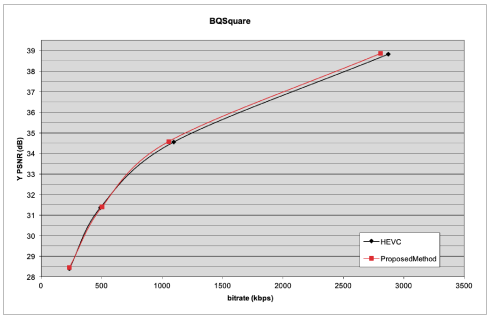
3. EXPERIMENTAL RESULTS

3.1. CNN Training

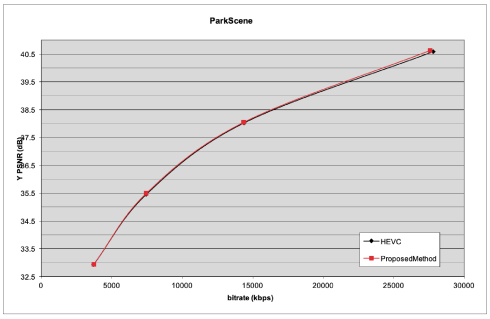
Our CNN network is trained on Caffe platform [14]. As mentioned in Section 2, the training data can be derived conveniently at the decoder side. Thus we extract fractional motion vectors and the relative prediction and residual parts of there corresponding reference blocks from the encoded bitstream, which is coded under the low delay P configuration. The sequences we used to generate training data are two 4K sequences *TrafficFlow*, *CampfireParty* [15], and one test sequence of HEVC *BlowingBubbles*. With all these sequences we can get hundreds of thousands of training data pairs. We train separate network for different QP values - 22, 27, 32, 37, and network with the closest QP to current slice will be adopted when conducting the encoding process.

3.2. Comparison with HEVC

The proposed dual-input CNN-based fractional interpolation method is implemented in HM-16.7 to conduct a comprehen-



(a) BQSquare



(b) ParkScene

Fig. 3. Rate-Distortion curves of two typical sequences.

sive comparison. Currently, we only replace the interpolation process for half-pel samples of luma component. Simulations are also conducted under the low delay P configuration and the BD-Rate metric is utilized to evaluate the performance. The overall comparison results with original DCTIF based interpolation method in HEVC are shown in table.1. It can be observed that our proposed scheme can achieve 0.9% bitrate reduction for luma component on average, and the highest coding gain can be up to 2.1%. To demonstrate the performance of our proposed method intuitively, several rate-distoration curves are shown in Fig.3.

In order to further verify our proposed dual-input interpolation scheme, we also compare this approach with our previous work [11], which is a representative whole frame based interpolation method. The convolutional neural network in [11] is used to generate an enlarged reference frame and the input of network is a reconstructed reference frame. The average bitrate saving of previous work is 0.4%, which validates the efficiency of the new method.

4. CONCLUSION

In this paper, we present a novel CNN-based interpolation method for HEVC inter prediction. Our network takes dual inputs, which are the prediction and residual parts of current coding block's reference block. With this dual-input CNN network, we can predict and generate the fractional samples that are much closer to current block. Compared with using reference block as input, separately processing its prediction and residual parts helps extract more information from different frequency components. And using convolution operation to re-combine them is also superior to the direct linear addition. Comprehensive experiments have been conducted to verify the efficiency of our method. The experimental results show this novel approach achieves great bitrate reduction. We apply the same network model to all sizes prediction units in this paper. The proposed method is expected to obtain better results if we use individual model for different size prediction units respectively and use a larger and more diverse training dataset.

Table 1. Overall BD-Rate Results on common test sequences

Classes	Sequences	BD-Rate[%]		
		Y	U	V
ClassB 1080p	Kimono	0.2	0.2	0.1
	ParkScene	-0.9	0.1	0.4
	Cactus	-0.7	0.2	-0.3
	BasketballDrive	-1.4	-0.3	0.0
	BQTerrace	-0.4	0.0	0.4
ClassC WVGA	BasketballDrill	-0.9	-0.2	0.0
	BQMall	-1.2	0.1	-0.3
	PartyScene	-0.8	0.2	0.4
	RaceHorses	-1.2	0.1	0.3
ClassD WQVGA	BasketballPass	-1.1	-0.4	0.1
	BQSquare	-2.1	0.1	-0.1
	BlowingBubbles	-1.3	0.1	0.4
	RaceHorses	-0.9	-0.3	-0.1
ClassE 720p	FourPeople	-1.3	-0.2	-0.2
	Johnny	-0.9	0.2	-0.4
	KristenAndSara	-0.7	0.4	0.5
ClassF	BasketballDrillText	-0.6	0.7	0.3
	ChinaSpeed	0.3	0.1	0.3
	SlideEditing	-0.9	0.0	0.2
	SlideShow	-1.3	-0.3	-0.1
Average	Class B	-0.6	0.0	0.1
	Class C	-1.0	0.0	0.1
	Class D	-1.3	-0.1	0.1
	Class E	-0.9	0.1	0.0
	Class F	-0.6	0.1	0.2
Overall	All	-0.9	0.0	0.1

5. ACKNOWLEDGMENTS

This work was supported by NSFC (61671296 and 61527804), the 111 Project (B07022 and Sheitc No.150633) and STCSM 18DZ2270700.

6. REFERENCES

[1] Kemal Ugur, Alexander Alshin, Elena Alshina, Frank Bossen, Woo-Jin Han, Jeong-Hoon Park, and Jani Lainema, "Motion compensated prediction and interpolation filter design in h. 265/hevc," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 946–956, 2013.

[2] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay 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.

[3] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, et al., "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.

[4] Thomas Wedi, "Adaptive interpolation filter for motion compensated hybrid video coding," in *Picture Coding Symposium (PCS 2001), Seoul, Korea*, 2001.

[5] Bernd Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604–612, 1993.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[7] Simon Niklaus, Long Mai, and Feng Liu, "Video frame interpolation via adaptive convolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, vol. 1, p. 3.

[8] Yuanying Dai, Dong Liu, and Feng Wu, "A convolutional neural network approach for post-processing in hevc intra coding," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[9] Zhenghui Zhao, Shiqi Wang, Shanshe Wang, Xinfeng Zhang, Siwei Ma, and Jiansheng Yang, "Cnn-based bi-directional motion compensation for high efficiency video coding," in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*. IEEE, 2018, pp. 1–4.

[10] Ning Yan, Dong Liu, Houqiang Li, and Feng Wu, "A convolutional neural network approach for half-pel interpolation in video coding," in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–4.

[11] Han Zhang, Li Song, Zhengyi Luo, and Xiaokang Yang, "Learning a convolutional neural network for fractional interpolation in hevc inter coding," in *Visual Communications and Image Processing (VCIP), 2017 IEEE*. IEEE, 2017, pp. 1–4.

[12] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[13] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[15] Li Song, Xun Tang, Wei Zhang, Xiaokang Yang, and Pingjian Xia, "The sjtu 4k video sequence dataset," in *Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on*. IEEE, 2013, pp. 34–35.