An Image Compression Framework with Learning-based Filter

Heming Sun[†] [14], Chao Liu[‡] [2], Jiro Katto[§13] and Yibo Fan[¶2]

[1]Waseda Research Institute for Science and Engineering, Waseda University, Tokyo Japan
[2]State Key Lab of ASIC and System, Fudan University, Shanghai P.R. China
[3]Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo Japan
[4]JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama, Japan

Abstract

In this paper, a coding framework VIP-ICT-Codec is introduced. Our method is based on the VTM (Versatile Video Coding Test Model). First, we propose a color space conversion from RGB to YUV domain by using a PCA-like operation. A method for the PCA mean calculation is proposed to de-correlate the residual components of YUV channels. Besides, the correlation of UV components is compensated considering that they share the same coding tree in VVC. We also learn a residual mapping to alleviate the over-filtered and under-filtered problem of specific images. Finally, we regard the rate control as an unconstraint Lagrangian problem to reach the target bpp. The results show that we achieve 32.625dB at the validation phase.

1. Introduction

The image/video lossy compression performance has been continuously improved with the development of the coding draft/standard. JPEG, JPEG2000, BPG for image and H.264/AVC, H.265/HEVC, H.266/VVC for video have been published in decades. Some learning-based end-to-end image compression methods [1, 2, 3] have been proposed as well. The intra(image) coding architecture of H.26x includes the prediction, transformation, entropy coding and loop filtering. It uses the neighbor pixels to predict the unknown blocks. The difference between prediction and original pixels are transformation and send to entropy coding. Finally, the post-filtering is utilized to reduce the artifacts in the reconstructed samples, such as blocking, ringing and blurring. However, the VTM built-in filters are not satisfying and the artifacts caused by the lossy image compression decrease the human perceptual quality. Recently, many learning-based methods [4, 5, 6, 7] are proposed to decrease those artifacts and achieve significant improvement in both objective and subjective evaluation.

Video coding standard H.26x is mainly designed to compression video or image in YUV color space, which has a more concentrated variance then RGB space. ITU-R BT.601[8] is widely used to convert the raw RGB image into YUV space. However, this conversion can't fully eliminate the correlation between different components. The optimal linear de-correlation transformation is principal component analysis (PCA)[9], its coefficients are not the same for different images. In the process of PCA, the images need to minus their mean value firstly. Those residual of RGB is used to calculate the covariance matrix. The eigenvalue and eigenvector of the covariance matrix represent the variance proportion and the basis of conversion space, respectively. For the color space conversion from coded YUV images to RGB, directly using the inverse transform matrix of PCA may be viable but not the best. The reconstructed samples are always accompanied by some noise, which will be transformed as well and result in unbearable square errors. A better solution to solve this question is using the least squares measurement (LSM)[10] to distribute the noise into all samples and achieves an overall best trade-off of the whole image.

Considering what mentioned above, we propose a framework based on VimicroABCnet[5] and some changes are adopted to improve its coding performance. Firstly, we modify the PCA to make the VTM[11] could compress the transformed YUV components at a higher compression rate. To decouple the difficult filtering problem of three components to three independent ones, we build our filter in YUV color space. This kind of filtering can use the unintentional pre-processing from PCA and reduce the noise of inputs to LSM. Finally, the learned residual mapping is proposed

[†]hemingsun@aoni.waseda.jp
[‡]chaoliu18@fudan.edu.cn
[§]katto@waseda.jp
[¶]Corresponding author, fanyibo@fudan.edu.cn

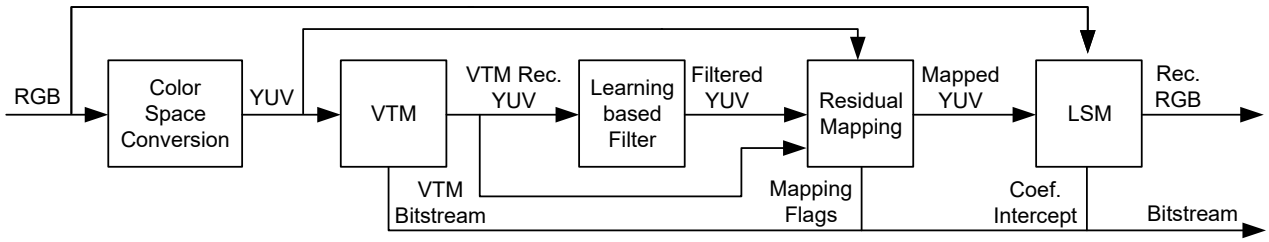

Figure 1. The proposed coding framework.

to improving the generalization ability of neural network-based filter, we use the linear regression that has only fewer parameters to learn a transformation between the filtered residual and distortion, which helps a single neural network to filter different images adaptively.

2. Proposed Methods

2.1. Framework Design

Our codec framework is shown in Fig. 1. The image with RGB format is first converted to YUV by using color space conversion. After that, the converted samples are sent to the VTM codec. Both the VTM reconstruction and bits are obtained after VTM coding. In the residual mapping phase, we map the learned residual to the distortion between original and VTM reconstructed samples. Finally, the mapped YUV is converted to RGB by using LSM.

During the encoding process, the input RGB and converted YUV need to be sent to LSM and residual mapping to obtain the coefficients for the transformation in these modules. Both the coefficients and VTM bits constitute the coded bitstream. For the decoding process, this bitstream is sent to VTM, residual mapping and LSM, respectively. And we can decode the reconstructed YUV and RGB sequentially.

2.1.1 Color Space Conversion

Similar to ABC color space conversion[10], the PCA-like algorithm is used to replace the standard color space conversion BT.601. The difference mainly consists of two parts, one is that the prediction value is used to replace the mean value in PCA and the other one is the chroma components rotation. The experiment shows that this method achieves 0.006dB gain in CLIC2020s valid dataset.

As we know, PCA can effectively reduce the pixel's correlation between different components. However, it is the residual between prediction and input pixels that accounts for the main proportion of coded bitstream in VTM. That is to say, if the residuals of different components have some correlation between each other, the transformed residuals may have correlated frequency components, which produces the redundancy in the bitstreams. Therefore, a better method for the average calculation in PCA should consider the VTM prediction, which better estimates the actual residuals. The RGB to YUV conversion can be written as follow.

$$x \begin{bmatrix} R \\ G \\ B \end{bmatrix} + y = \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{1}$$

$$x \begin{bmatrix} R - R_x \\ G - G_x \\ B - B_x \end{bmatrix} = \begin{bmatrix} Y \\ U \\ V \end{bmatrix} - (x \begin{bmatrix} R_x \\ G_x \\ B_x \end{bmatrix} + y) \tag{2}$$

where both x and y are unknown matrices for this transform. The shape of x is 3×3 and that of y is 3×1. And y is repeated in the second axis to match the size of input image. We assume the x are the eigenvectors of covariance matrix of $[R - R_x, G - G_x, B - B_x]^T$. So the left item of all input samples is de-correlated in the Eq. 2. We let the $[Y - Y_{pred}, U - U_{pred}, V - V_{pred}]^T$ equal to this item, which means the correlation in the residual is removed as well. The subscript "pred" in Eq .3 indicates the prediction samples in VTM.

$$\begin{bmatrix} Y - Y_{pred} \\ U - U_{pred} \\ V - V_{pred} \end{bmatrix} = \begin{bmatrix} Y \\ U \\ V \end{bmatrix} - (x \begin{bmatrix} R_x \\ G_x \\ B_x \end{bmatrix} + y) \tag{3}$$

$$x \begin{bmatrix} R_x \\ G_x \\ B_x \end{bmatrix} + y = \begin{bmatrix} Y_{pred} \\ U_{pred} \\ V_{pred} \end{bmatrix} \tag{4}$$

From Eq. 1, we approximate that,

$$x \begin{bmatrix} R_{pred} \\ G_{pred} \\ B_{pred} \end{bmatrix} + y \approx \begin{bmatrix} Y_{pred} \\ U_{pred} \\ V_{pred} \end{bmatrix} \tag{5}$$

Substituting Eq. 5 into Eq. 4,

$$\begin{bmatrix} R_x \\ G_x \\ B_x \end{bmatrix} \approx \begin{bmatrix} R_{pred} \\ G_{pred} \\ B_{pred} \end{bmatrix} \tag{6}$$

So $[R_x, G_x, B_x]^T$ is obtained to calculate the unknown x and y in PCA. For simplicity, we only use the DC prediction mode in our experiment.

For the chroma components rotation, we rotate the chroma axis to let them have a correlation with each other. This is because the two chrominance components share the same coding tree in VTM. And we scale them to ensure the sum of variance of chroma components unchanged.

$$x = \begin{bmatrix} x_y \\ x_u \\ x_v \end{bmatrix} \rightarrow x' = \begin{bmatrix} x_y \\ x_{u'} \\ x_{v'} \end{bmatrix} = \begin{bmatrix} x_y \\ \frac{x_u + x_v}{\sqrt{2}} \\ \frac{x_u - x_v}{\sqrt{2}} \end{bmatrix} \tag{7}$$

2.1.2 The Proposed Filtering

The main difference between the proposed framework with VimicroABCnet is that we construct our learning-based filter in YUV space instead of RGB space. The reasons for the design of the framework include three aspects. Firstly, the images in YUV space is converted by PCA-like algorithms. As we know, PCA that has a powerful capability to pre-process the dataset can control the mean and variance of the dataset effectively, which is quite helpful for the training of our neural networks. In our cases, we set the mean and range of YUV components are the same with BT.601 by using the proposed PCA. So the YUV datasets with the same mean and range are sent to the training of neural networks, while the RGB dataset has different means and ranges. Secondly, the artifacts for the learning-based filter to remove is produced by lossy VTM codec, which coding images in YUV color space. If we convert the lossy YUV image into RGB space, those artifacts will be converted as well and the distortion from different components may overlap each other. Compared with RGB space, the YUV space is decoupled. So we can train three individual models for YUV components, respectively. Those models can handle the artifacts independently and effectively. Moreover, we choose models with different complexity for luma and chroma to reduce complexity. The relative simple model is enough for the filtering of chroma components in our experiments. Another important reason for choosing YUV space is to make the LSM less affected by those artifacts. In the color space conversion from RGB to YUV, the transform matrix and intercept are the same for all pixels in one image. So if we prefer to convert the color space before filtering, the different levels of distortion in a specific image may make the optimal transform matrix and intercept for different pixels are not the same. Compared with VTM reconstructed images, the filtered images in YUV space have fewer artifacts and is more close to the original images in YUV space. In other words, this filtering process reduces the noise in the inputs to LSM and make it less affected by the artifacts.

2.1.3 Learned Residual Mapping

Due to the uneven distribution of the training dataset, the training process of a neural network is a trade-off between different training samples. For individual images, the learning-based filter may be over-filtered or under-filtered, which is unsatisfied with the human perceptual quality. To solve this question and further improve the generalization ability of the trained model, we use regression to map the learned residual of the learning-based filter to the distortion between original images and VTM reconstructed images. Consequently, this kind of over-filtered or under-filtered can be alleviated. This method balances the detail texture and overall smoothness in terms of objective evaluation.

The output of VTM is denoted by $\hat{x}$, so the reconstructed images $\tilde{x}$ of the learning-based filter can be calculated by the sum of the learned residual $\tilde{\varepsilon}$ and $\hat{x}$.

$$\hat{x} + \tilde{\varepsilon} = \tilde{x} \tag{8}$$

Correspondingly, the original images x can be formulated as the sum of distortion ε and $\hat{x}$.

$$\hat{x} + \varepsilon = x \tag{9}$$

From the perspective of enhancing subjective quality, minimizing the mean square error(MSE) between $\tilde{x}$ and x is equivalent to map $\tilde{\varepsilon}$ to ε with an accurate regression. In this case, we use the linear regression $f(\cdot)$ to map the $\tilde{\varepsilon}$. The reason for choosing linear regression is that it only needs 2 parameters for each component, named coefficients w and intercept b.

$$f(\tilde{\varepsilon}) = w\tilde{\varepsilon} + b \tag{10}$$

Considering only the intercept term in linear regression, it is equivalent to the sample adaptive offset (SAO) in VTM. Both of them are aimed at compensating offsets by using the bias term. However, the bias term needs some extra bits which are redundant if we use the SAO and the intercept term of linear regression at the same time. Furthermore, the whole image shares the same parameters to reduce the bits and avoid artificial imprints in our design. For most images, its reconstructed output is always unbiased, so we drop the intercept term and quantify coefficients into fixed points to save its consumed bits. Specifically, we set 16 and 8 different levels for luminance and chrominance, respectively. The number of required bits is reduced from 192(2 floats per component) to 10 (4 3 3 for YUV) per image. So the overhead of this method is only $10/2000000 = 0.000005$bpp for a general image with 2000×1000. At the same time, the case of not performing residual mapping is retained in the candidate level. Therefore, the residual mapping could ensure the coding gain with negligible rate overhead.

2.2. Rate Control

In this subsection, we describe our rate control algorithm, which is mainly based on the Lagrangian method. The distortion and rate for different QPs and images are indicated as D_{ij}, R_{ij}, where i and j are the index of QPs and

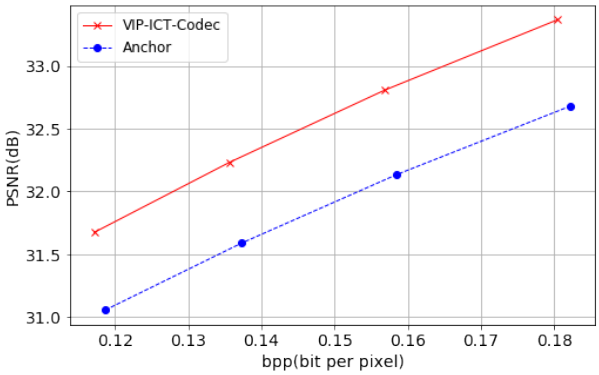

Figure 2. PSNR comparison around 0.15bpp.

images, respectively. So the rate control problem with target bits constraint C is

$$\min \sum_j D_{ij} \tag{11}$$

$$\text{subject to:} \sum_j R_{ij} \leq C \tag{12}$$

This constraint optimization problem can be converted into an unconstraint Lagrangian function.

$$L = \sum_j D_{ij} + \lambda(\sum_j R_{ij} - C) \tag{13}$$

$$= \sum_j (D_{ij} + \lambda R_{ij}) - \lambda C \tag{14}$$

Furthermore, those images are independent with each other, which means that minimizing the overall loss L is equivalent to minimize the loss $D_{ij} + \lambda R_{ij}$ of every single image. So we fix the hyper-parameter λ and save the candidate QP with the smallest loss of all images as a preliminary selection. Finally, we fine-tune the preliminary selection to make its bits closer to the target constraints.

3. Experiment

The structure of the learning-based filter is similar to the one in [5]. Differently, the depth and the number of feature maps for luminance's network are extended to 56 and 128, respectively. CLIC2020 training set and DIV2K [12] with patch size of 64×64 are used in the training phase. We use the VTM 7.1 without built-in filters as the anchor to produce the training samples. Fig. 2 compared the PSNR between the proposed model and the anchor. It can be found that our proposed model achieves 32.625dB at 0.15 bpp.

4. Conclusion

In this paper, an image compression framework with a neural network-based filter is proposed for CLIC 2020 challenge. This framework mainly consists of color space conversion, VTM, learned residual mapping and LSM. First, we introduce the detail in our color space conversion. The VTM prediction and chroma rotation were utilized to modify the PCA. This color space conversion produces the frames that have a higher VTM compression ratio than using standard PCA directly. Different from previous works, we construct our filter in YUV space instead of RGB space. To describe the reason for filtering in YUV space, we provide a detailed explanation from three different aspects, including the relationship between the filtering module and PCA-like conversion, filtering module itself and LSM. After that, we proposed a novel module, called residual mapping. It is aimed at solving the over-filtered or under-filtered for individual images while using the same trained model. By using some additional bits, this module can further improve the generalization ability of the trained model. By using the Lagrangian method, a rate control method is designed to find the best image set with the smallest mean square errors finally. Experimental results show our proposed framework achieves about 32.625dB for CLIC 2020 validation dataset.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61674041, in part by Alibaba Group through Alibaba Innovative Research (AIR) Program, in part by the STCSM under Grant 16XD1400300, in part by the pioneering project of academy for engineering and technology and Fudan-CIOMP joint fund, in part by the National Natural Science Foundation of China under Grant 61525401the Program of Shanghai Academic/Technology Research Leader under Grant 16XD1400300, the Innovation Program of Shanghai Municipal Education Commission, in part by JST, PRESTO Grant Number JPMJPR19M5, Japan.

References

[1] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv e-prints*, p. arXiv:1802.01436, Jan. 2018.

[2] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep residual learning for image compression," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[3] L. Zhou, Z. Sun, X. Wu, and J. Wu, "End-to-end optimized image compression with attention mechanism," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[4] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, "Variational autoencoder for low bit-rate image compression," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[5] M. Li, "Vimicroabcnet: An image coder combining a better color space conversion algorithm and a post enhancing network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[6] M. Lu, T. Chen, H. Liu, and Z. Ma, "Learned image restoration for vvc intra coding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[7] C. Liu, H. Sun, J. Chen, Z. Cheng, M. Takeuchi, J. Katto, X. Zeng, and Y. Fan, "Dual learning-based video coding with inception dense blocks," in *2019 Picture Coding Symposium (PCS)*, Nov 2019, pp. 1–5.

[8] I. T. Union, "Bt.601 : Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios."

[9] I. Jolliffe, *Principal Component Analysis, second edition (Springer)*.

[10] M. Li, "A better color space conversion based on learned variances for image compression," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[11] S. K. J. Chen, Y. Ye, *Algorithm description for Versatile Video Coding and Test Model 7 (VTM 7)*, document JVET-P2001, Joint Video Experts Team (JVET), Oct 2019.

[12] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.