Image and Video Compression Laboratory Outline of implemented Techniques

Yanchen Huang

Technical University of Munich

Matrikel-Nr: 03715208

Email: yanchen.huang@tum.de

Chang Liu

Technical University of Munich

Matrikel-Nr: 03707882

Email: ge29ney@tum.de

I Introduction

Optimization methods: PCA for color space conversion, deblocking filter, DWT are implemented in our project and explained in detail in this outline.

II Color Space Conversion: PCA

Conversion between RGB and YCbCr usually follows ITU-R BT.601 standard, which is a fixed space conversion matrix. However, different images have different color properties, to preserve more color information for each image, a better conversion algorithm such as PCA is needed. Based on the reference paper, "A better Color Space Conversion Based on Learned Variances For Image Compression[1]", an optimal RGB-YCbCr convert matrix for each image is defined in equation (1) as T_{enc} and $Offset_{enc}$.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ z1 & z2 & z3 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} Y\_offset \\ Cb\_offset \\ Cr\_offset \end{bmatrix} \tag{1}$$

Each row in T_{enc} is the direction of the new YCbCr space's base axis, to find the optimal axis, PCA is applied. Firstly, the whole picture is divided into 8*8 grids, each pixel in the grid should minus the mean value within the same grid. Secondly, covariance matrix of size 3*3 is computed and PCA is utilized to find eigenvalues and eigenvectors. Stack the eigenvectors based on the descend order of the corresponding eigenvalues to form our T_{pca} as follow.

$$\begin{bmatrix} x_p1 & x_p2 & x_p3 \\ y_p1 & y_p2 & y_p3 \\ z_p1 & z_p2 & z_p3 \end{bmatrix}$$

Thirdly, based on the reference paper, according to YCbCr's range restrictions, scaling is done based on following equation(2).

$$\begin{aligned} &[x1, x2, x3] = L1\_normalize([x_p1, x_p2, x_p3]) * 219/255 \\ &Scale_{Cb} := 224/255/(|y_p1| + |y_p2| + |y_p3|) \\ &[y1, y2, y3] = [y_p1, y_p2, y_p3] * Scale_{Cb} \\ &Scale_{Cr} := 224/255/(|z_p1| + |z_p2| + |z_p3|) \\ &[z1, z2, z3] = [z_p1, z_p2, z_p3] * Scale_{Cr} \end{aligned} \tag{2}$$

$Offset_{enc}$ is defined as equation(3).

$$\begin{aligned} &Y\_offset = 16 \\ &Cb\_offset = -1 * sum\_negative(y1, y2, y3) * 255 + 16 \\ &Cr\_offset = -1 * sum\_negative(z1, z2, z3) * 255 + 16 \end{aligned} \tag{3}$$

In decoding, instead of using T_{enc}^{-1}, T_{dec} is computed through least square method, i.e. linear regression. The convert formula is shown in equation(4).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = T_{dec} * (\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} Y\_offset \\ Cb\_offset \\ Cr\_offset \end{bmatrix}) \tag{4}$$

To further optimize the result, polynomial curve fitting is applied to the converted RGB image, i.e. use converted RGB image and original RGB image to train coefficient for a polynomial mapping of degree 3 to find the best match.

III Deblocking filter

Deblocking filter is applied to reduce blocking distortion by smoothing the block edges. Based on H.264/AVC Loop filter[2], we first take vertical and horizontal edges of 8*8 macroblock as depicted in Fig. 1.

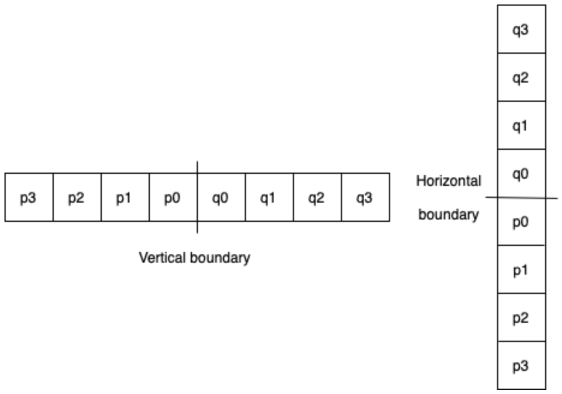

Figure 1: *Pixels adjacent to vertical and horizontal boundaries*

Secondly, to determine the true boundary for applying deblock filter, two conditions shown as follow should be fulfilled, notice that in our project, thresholds α and β are prefixed.

- $|p0 - q0| < \alpha$
- $|p1 - q1| < \beta$

Thirdly, a 4-tap linear filter is applied with inputs p1, p0, q0, q1 if the filter is applicable. Output value would be assigned to position p0 and q0 as the new value for the filtered boundary.

References

[1] Li, Manman. A Better Color Space Conversion Based on Learned Variances For Image Compression. CVPR Workshops (2019).

[2] H.264/AVC Loop Filter. https://www.vcodex.com/h264avc-loop-filter/. Accessed 02.02.2020.