ISSN 2464-4617 (print)
ISSN 2464-4625 (DVD)
Computer Science Research Notes
CSRN 3201
WSCG 2022 Proceedings

Region of interest in JPEG

David Barina Ondrej Klima

Centre of Excellence IT4Innovations
Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2, Brno
Czech Republic
{ibarina,iklima}@fit.vutbr.cz

Abstract

This paper describes three methods to encode a region of interest (ROI) of arbitrary shape in the JPEG standard. The ROI is part of the image encoded with higher quality than the rest of the image. Two of the described methods are based on thresholding of DCT coefficients. The last method is based on the early termination of the entropy coding phase. All methods are fully compatible with existing JPEG decoders. Using ROI, it is possible to reduce the bitrate to only a fraction of the original value. The results in this paper show that thresholding of original DCT coefficients provides superior performance in terms of the PSNR-B index.

Keywords

JPEG, region of interest, image compression

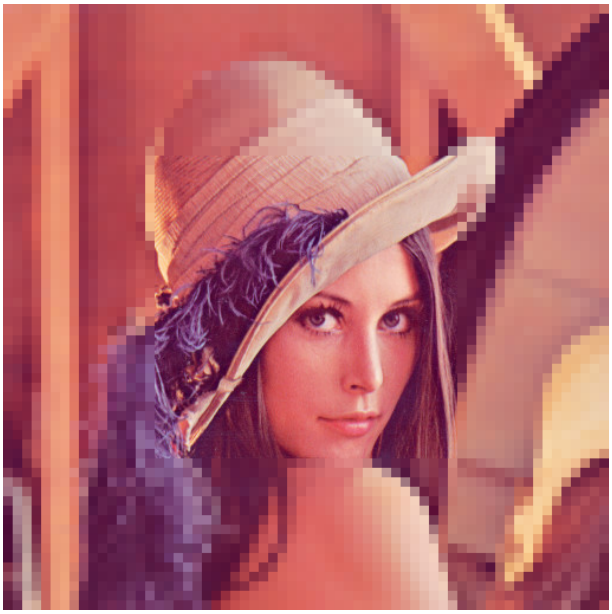


Figure 1: The image was compressed using the JPEG method. The face in the image has been selected as a region of interest.

1 INTRODUCTION

In some applications, certain areas of the image are more important than the rest. We can imagine, e.g., human faces for face recognition (Figure 1) or license plates for automatic license plate recognition system. Then it may be desirable to compress these areas with higher quality than the rest of the image. In image processing, these important parts of images are often referred to as the regions of interest (ROI).

JPEG [3] is the most widely used image format in the world. The ROI functionality is already included in some newer image compression standards, such as the JPEG 2000. However, the functionality is missing in the original JPEG format (ITU-T T.81 / ISO/IEC 10918-1). This paper describes three methods compatible with the original JPEG standard for encoding images with a ROI. The principle of these methods is to drop less important details in non-ROI areas. All three methods use only one common quantization matrix (per component). The resulting bitstream fully conforms to the JPEG standard (no modification to existing decoders is required).

The rest of the article is organized as follows. Section 2 familiarizes the reader with the necessary background for understanding the rest of this article. Section 3 presents the three proposed methods for encoding a ROI in the JPEG standard. Section 4 evaluates these methods using the PSNR-B quality assessment index. Finally, Section 5 summarizes the paper.

2 BACKGROUND

Because this article discusses the JPEG format's internalities, it is necessary to explain how this format works. This 1992 standard, commonly referred to as JPEG, is primarily intended for lossy image compression. For lossy compression, JPEG supports sequential and progressive image data transfer. The input image is processed in the YCbCr color model. Individual components can be subsampled horizontally and vertically. The chromatic components Cb and Cr are typically subsampled in one or both directions in half (4:2:2 and 4:2:0). The Y, Cb, and Cr components of the color model are further processed separately. The following text describes

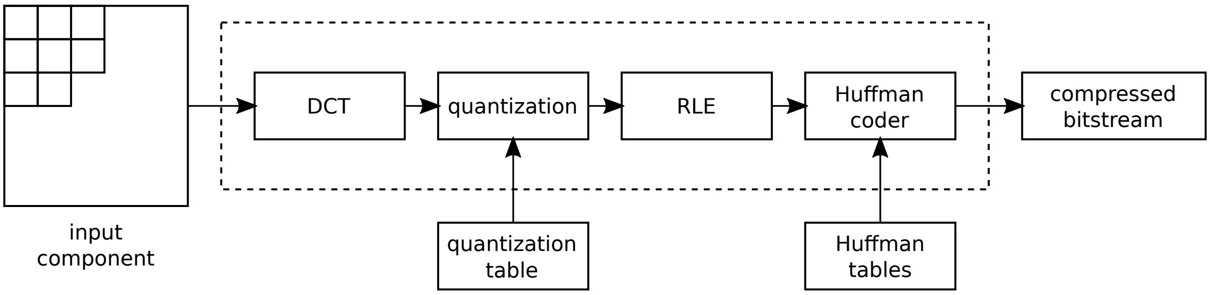


Figure 2: Simplified JPEG encoder scheme. The same scheme is applied to each image component.

the scheme shown in Figure 2. Each component is divided into blocks of 8×8 samples. This division is the main reason for the block artifacts in images with a high degree of compression. The 8×8 block is the fundamental data unit of the JPEG format. On each such block, its discrete cosine transform (DCT) [1] is then computed. The result gives 8×8 DCT coefficients. The DCT coefficient at position (0,0) indicates the shift from zero, and it is called the DC coefficient. The remaining 63 coefficients are referred to as the AC coefficients. They indicate the weights with which the corresponding two-dimensional cosine function is present in the block. The procedure just described is fully invertible. No data is lost. The next step is to quantize the coefficients. To do this, both the encoder and decoder must receive a so-called quantization table. This is a table with the integer values by which the corresponding DCT coefficient is to be divided during compression. Subsequently, each coefficient is rounded to the nearest integer. During decompression, the coefficients are in turn multiplied by these values. Because the coefficients' values have been rounded to integers, the reconstructed coefficient is quantized to several levels. The quantization table determines how much data is lost during compression. Higher values mean coarser quantization, i.e., more information loss. Therefore, the values in the table are created concerning the target quality and the human psychovisual model. A different table is used for the luminance component Y and the chromatic components Cb and Cr. The reason for this is the greater sensitivity of the human eye to luminance than to chromatic components. Quantized coefficients are further processed in the zig-zag scan. In the beginning, this scan processes coefficients with lower frequencies. These typically have a higher amplitude and are therefore more likely to remain non-zero even after quantization. Conversely, the coefficients at the end of the scan are more likely to be zero. A sequence of linearized coefficients is further subjected to a variant of RLE coding (only zeros are considered). The last step of the compression is Huffman or arithmetic coding. Both variants work based on supplied tables.

ROI provides an overall improvement in perceived image quality compared to conventional coding at the same bitrate. Alternatively, ROI is a way to save some bitrate on less essential parts of the image. In general, a ROI can be of arbitrary shape. However, in connection with the JPEG standard, this shape is constrained by 8×8 block nature of the JPEG coding. Outside of the JPEG environment, these constraints may differ. For example, in the JPEG 2000, a direct successor to the JPEG format, the ROI can be defined on a pixel basis.

The method used in the JPEG 2000 (Part 1) [2] for encoding images with a ROI is called the Maxshift method. When an image is encoded using JPEG 2000, a wavelet transform [1] of the image is initially computed. This step is followed by entropy coding of resulting coefficients, which processes the coefficients bitplane-by-bitplane. Maxshift method shifts up the wavelet coefficients so that the bits associated with the ROI are encoded in higher bit-planes than the bits associated with the background. In contrast to the methods proposed in this paper, an encoder can place those bits in the bitstream before the bits corresponding to the background.

3 REGION OF INTEREST IN JPEG

This section presents the three strategies we have chosen to implement ROI in JPEG format. For compatibility, all of these strategies must encode ROI and non-ROI blocks in a single interleaved scan with a single quantization table.

3.1 Baseline process

The JPEG transforms an input image (each component) using disjoint 8×8 blocks of pixels. Particularly, the nth image block is transformed from image domain s^n to frequency domain S^n using

$$S^n_{v,u} = \sum_{y=0}^{7} \sum_{x=0}^{7} s^n_{y,x} \, g_{v,u,y,x}, \tag{1}$$

where the

$$\left\{ g_{v,u,y,x} = C_v C_u \frac{1}{4} \cos\left(\frac{v\pi(2y+1)}{16}\right) \cos\left(\frac{u\pi(2x+1)}{16}\right) \right\}_{0 \le v,u < 8} \tag{2}$$

is set of DCT basis vectors, and

$$C_u = \begin{cases} 1/\sqrt{2} & \text{if } u = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

The next step in the process is a quantization of the coefficients. The JPEG uses uniform scalar quantization

$$S^{q,n}_{v,u} = \text{round}\big(S^n_{v,u}/Q_{v,u}\big). \tag{4}$$

The quantized coefficients $S^{q,n}$ are then encoded by entropy coder using zig-zag scan $\zeta_{v,u} : \mathbf{N}_0 \times \mathbf{N}_0 \to \mathbf{N}_0$. Our implementation uses adaptive Huffman coding. The Huffman coding uses two Huffman tables per component, one for the DC coefficients $(u, v = 0)$, and the other for AC coefficients $(u, v \neq 0)$. The Huffman table for AC coefficients is indexed by the pair (zero-run length, non-zero coefficient). The special pair (0, 0) is used if all coefficients up to the end of the current block are zero. This symbol is known as the end-of-block symbol (EOB).

3.2 Coefficient thresholding

This section aims to encode a region of interest (foreground) with higher quality than the rest of the image (background). Let $\mathscr{B}$ be a set of background block indices. Let $\mathscr{F}$ be a set of foreground block indices. Note that $\mathscr{B} \cup \mathscr{F}$ covers all block indices in an image.

Our first idea was to alter the DCT coefficients. To suppress the background details, we apply the following nonlinear transform

$$\hat{S}^n_{v,u} = \begin{cases} \rho_\lambda\left(S^n_{v,u}\right) & \text{if } n \in \mathscr{B}, \\ S^n_{v,u} & \text{if } n \in \mathscr{F}, \end{cases} \tag{5}$$

where

$$\rho_\lambda(x) = \begin{cases} x & \text{if } |x| > \lambda, \\ 0 & \text{if } |x| \leq \lambda \end{cases} \tag{6}$$

is hard thresholding operator [1] with the threshold λ. The process continues with quantization

$$S^{q,n}_{v,u} = \text{round}\big(\hat{S}^n_{v,u}/Q_{v,u}\big) \tag{7}$$

as usual. Effectively, this procedure uses uniform scalar dead-zone quantization [5] for background blocks. The transformed coefficients inside interval $[-\lambda, +\lambda]$ are quantized to zero. The interval $[-\lambda, +\lambda]$ is called the "dead zone". The procedure just described will hereinafter be referred to as Coefficient thresholding.

3.3 Quantized coefficient thresholding

Our second idea was to threshold quantized DCT coefficients. This method has the advantage that the entropy coder does not have to encode many symbols for background blocks (since zeros have replaced some). Formally, this step can be described as

$$S^{q,n}_{v,u} = \begin{cases} \rho_\lambda\left(\text{round}\big(S^n_{v,u}/Q_{v,u}\big)\right) & \text{if } n \in \mathscr{B}, \\ \text{round}\big(S^n_{v,u}/Q_{v,u}\big) & \text{if } n \in \mathscr{F}. \end{cases} \tag{8}$$

Also this procedure effectively uses uniform scalar dead-zone quantization for background blocks. However, the choice of λ threshold differs from that described in the previous section. This will from now on be called Quantized coefficient thresholding.

3.4 Cutting of coefficients

Our last idea was to modify the Huffman coding so that the EOB symbol is inserted prematurely for background blocks. In other words, we cut the coded sequence after some fixed number of coded coefficients (however, the DC coefficient is always fully encoded). This corresponds to resetting the coefficients

$$S^{q,n}_{v,u} = \begin{cases} \sigma_{\mu,v,u}\left(\text{round}\big(S^n_{v,u}/Q_{v,u}\big)\right) & \text{if } n \in \mathscr{B}, \\ \text{round}\big(S^n_{v,u}/Q_{v,u}\big) & \text{if } n \in \mathscr{F}, \end{cases} \tag{9}$$

where

$$\sigma_{\mu,v,u}(x) = \begin{cases} x & \text{if } \zeta_{v,u} < \mu, \\ 0 & \text{if } \zeta_{v,u} \geq \mu, \end{cases} \tag{10}$$

in the zig-zag sequence $\zeta_{v,u}$ starting at some fixed position μ. Note that the encoder could have also inserted the EOB symbol before this fixed position. The disadvantage of this procedure is the absence of high frequencies in the decoded image. The procedure will be called Cutting of coefficients.

4 EVALUATION

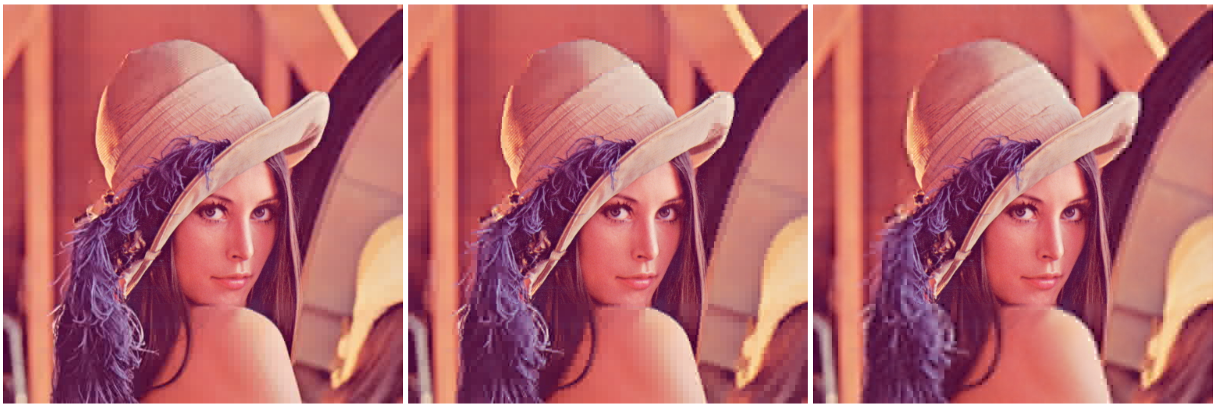
Since methods proposed in the previous section effectively increase quantization step size, blocking artifacts generally become more visible. Yim and Bovik [4] proposed a block-sensitive quality assessment index, named PSNR-B. The PSNR-B modifies PSNR by including a blocking effect factor. We use the PSNR-B index to evaluate the qualitative performance of the proposed methods. The index for reference image $\mathbf{x}$ and decompressed image $\mathbf{y}$ is defined as

$$\text{PSNR-B}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \frac{255^2}{\text{MSE-B}(\mathbf{x}, \mathbf{y})} \tag{11}$$

where

$$\text{MSE-B}(\mathbf{x}, \mathbf{y}) = \text{MSE}(\mathbf{x}, \mathbf{y}) + \text{BEF}(\mathbf{y}) \tag{12}$$

where BEF is the blocking effect factor. Further details are given in [4].



(a) (b) (c)

Figure 3: The Lenna with 3 bpp bitrate and ROI fixed at quality $q = 100$. From left: (a) Coefficient thresholding, (b) Quantized coefficient thresholding, and (c) Cutting of coefficients.

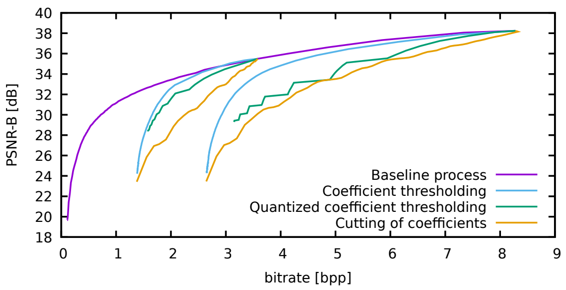


Figure 4: Results of the experiment on the Lenna image with her face used as the ROI. Two ROI quality was fixed at (i) $q = 95$ and (ii) $q = 100$.

In our evaluation, we compute quantization matrices based on different quality q—an integer in the interval $[1, 100]$. We compute the scaling factor

$$\beta = \begin{cases} 5000/q & \text{if } q < 50, \\ 200 - 2q & \text{if } q \geq 50. \end{cases} \tag{13}$$

The quantization matrices are then computed as

$$Q_{v,u} = \max\left(\min\left(\frac{\beta \cdot Q^{\text{ref}}_{v,u} + 50}{100}, 255\right), 1\right), \tag{14}$$

where Q^{ref} is the reference luminance or chrominance quantization matrix listed in the JPEG standard.

The performance of the individual methods from the previous section is demonstrated on a standard Lenna test image. Image size is 512×512 pixels. First, we compress the test image without ROI for each of the q in $[1, 100]$. This gives us an idea of what quality we could achieve at the most using ROI. In the next step, Lenna's face was used as the ROI (size 256×256 pixels, thus the ROI occupies exactly 25 % of the image area). For the experiment, we fixed the ROI quality to (i) $q = 95$ and (ii) $q = 100$. Next, we tested all permissible parameterizations of all three methods from the previous section. This gives us three PSNR-B dependences on the bitrate for (i) and three for (ii). All should be dominated by the curve from the beginning of the experiment. The result is shown in Figure 4. The (i) is shown in the left part and (ii) in the right part of the figure. It is clear that the Coefficient thresholding method dominates the other two methods. Furthermore, we see that using this method, we can conveniently halve the bitrate while maintaining full quality in the ROI. The Cutting of coefficients method proves to be the worst of the three. The extreme case of this experiment is shown in Figure 1. In this case, we kept only the DC coefficient in the non-ROI area. Methods discussed in this paper show the same behavior also on other images with a different ROI. The smaller the ROI area to the rest of the image, the more bitrate we can save.

To give an idea of the artifacts caused by the particular methods, the experiment for $q = 100$ and 3 bpp is shown in Figure 3. The original image at $q = 100$ without ROI has an 8.279 bpp bitrate.

We have released the software (complete implementation of the JPEG baseline encoder and decoder) used in this article as open source.[1] We want to emphasize that the JPEG files created in this way are fully compatible with existing decoders.

5 CONCLUSIONS

This paper proposed three novel methods for encoding a region of interest (ROI) in the original JPEG standard. Two of the methods are based on the thresholding of DCT coefficients in the non-ROI blocks. Similarly, the third method is based on the early termination of the entropy coding. The proposed methods allow for a reduction of the bitrate to only a fraction of the original value. The methods are fully compatible with existing JPEG decoders, and the resulting bitstream fully complies with the JPEG standard. The software used in this paper was released as open-source.

[1] `https://github.com/xbarin02/jpeg`

Acknowledgements

This work has been supported by the Technology Agency of the Czech Republic (TA CR) project Safe and Secure Traffic Systems of New Generation (no. VB01000048).

REFERENCES

[1] Mallat, S. *A Wavelet Tour of Signal Processing: The Sparse Way. With contributions from Gabriel Peyre*. Academic Press, 3 edition, 2009. ISBN 9780123743701.

[2] Taubman, D. S. and Marcellin, M. W. *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer, 2004. ISBN 978-1-4613-5245-7. doi: 10.1007/978-1-4615-0799-4.

[3] Wallace, G. K. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–44, Apr. 1991. ISSN 0001-0782. doi: 10.1145/103085.103089.

[4] Yim, C. and Bovik, A. C. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, 2011. ISSN 1941-0042. doi: 10.1109/TIP.2010.2061859.

[5] Yu, J. Advantages of uniform scalar dead-zone quantization in image coding system. In *2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914)*, volume 2, pages 805–808, 2004. doi: 10.1109/ICCCAS.2004.1346303.