Improving image quality by SSIM based increase of run-length zeros in GPGPU JPEG encoding

Stefan Petersson
Department of Creative Technologies
Blekinge Institute of Technology
SE-37179 Karlskrona, Sweden
Email: Stefan.Petersson@bth.se

Håkan Grahn
Department of Computer Science and Engineering
Blekinge Institute of Technology
SE-37179 Karlskrona, Sweden
Email: Hakan.Grahn@bth.se

***Abstract*—JPEG encoding is a common technique to compress images. However, since JPEG is a lossy compression certain artifacts may occur in the compressed image. These artifacts typically occur in high frequency or detailed areas of the image. This paper proposes an algorithm based on the SSIM metric to improve the experienced quality in JPEG encoded images. The algorithm improves the quality in detailed areas by up to 1.29 dB while reducing the quality in less detailed areas of the image, thereby increasing the overall experienced quality without increasing the image data size. Further, the algorithm can also be used to decrease the file size (by up to 43%) while preserving the experienced image quality. Finally, an efficient GPU implementation is presented.**

I. Introduction

JPEG encoding is a very common technique to compress images. JPEG encoding is adaptable in the sense that one can adjust the compression ratio, i.e., a low compression leads to high quality images but large files sizes and high compression leads to lower quality images but small file sizes. Thus, there is a trade-off between the image quality and the file size of the image. One problem with JPEG encoding is that certain artifacts may occur in high contrast or high frequency areas, as shown in Figure 1 and addressed in e.g. [10]. The problem can be reduced by increasing the JPEG quality factor, but at the cost of increased data size.

Fig. 1: JPEG artifacts are common in areas where abrupt changes in contrast are present.

Several approaches have been proposed to enhance JPEG encoding for different situations, e.g., using adaptive regions [15] or based on the human perception [3], [8], [9], [11]. The Structural SIMilarity (SSIM) index [12] measures the similarity between two images and has been used to improve video encoding [13], [14] and also for image compression [4]. The standard SSIM equation used to compare images is presented in Equation 1. Often an unnecessary amount of high frequency information is preserved when encoding low-quantized JPEG image data. Therefore, we propose an algorithm to decrease the encoded data size by removing frequency information based on an SSIM guided approach.

$$SSIM\,(x,y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1)$$

In this paper, we present an SSIM-based algorithm to

(i) decrease the compressed image file size by maintaining a high quality in high contrast areas, e.g., text, thin lines, etc., and decreasing the quality in the rest of the image to a lower but still acceptable level, or
(ii) iteratively increase the JPEG quality factor in high contrast or high frequency areas and lower the quality in low-frequency areas, while maintaining the same file size.

Our algorithm uses the SSIM index on 8x8 luminance pixel blocks to identify high contrast blocks, and then reduce the compression ratio of those blocks. Similarly, for low contrast blocks our algorithm increase the compression rate by increasing run-length zeros, and thus reduce the quality of those blocks.

We have implemented our algorithm by extending a high performance GPU-based implementation of a baseline JPEG encoder [6]. The baseline GPU encoder execution performance in comparison to the CPU based libjpeg-tugbo encoder is shown in Figure 2. An image is encoded targeting a specific JPEG quality and the algorithm reduces the data size based on metric information and is not able to increase quantitative metric results.

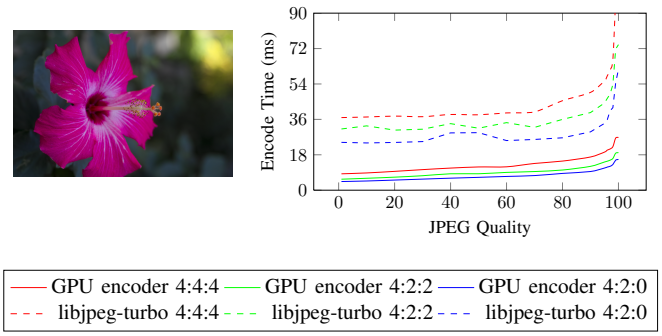


Fig. 2: JPEG encoder benchmark results in [6].

We evaluate the algorithm on all images in the Kodak Lossless True Color Image Suite [5] and on two other test images. Our results show that we can reduce the compressed image size by up to 43% on a 2268x1512 pixel image as compared to a standard JPEG encoder with the same quality level (Q90). Our results also show that we can increase PSNR by up to 1.29 dB in high contrast areas in the images in the Kodak suite as compared to a standard JPEG encoder at JPEG quality 90.

II. Why encode using SSIM?

The SSIM index is a full reference metric, which means that it relies on having a non-distorted source image when compared to, for example, a lossy compressed version of that source image. Different full reference metrics exist but only a few of them are designed with focus on human eye perception. The design goal of our algorithm is to reduce image quality in areas where the human eye would not see a noticeable difference. The SSIM metric is usually calculated using luminance information only [12].

When encoding JPEG data, luminance and chrominance planes are often separated and thereafter transformed to a frequency domain using the Discrete Cosine Transform (DCT). Previous work have shown that the SSIM index can be calculated in DCT space [1], [7], making it even more suitable for JPEG encoder integration since the already transformed coefficients can be used directly without further processing. In our algorithm the SSIM index is calculated using the formula found in [7], which is presented in Equation 2. The input signals $\{\mathbf{x},\mathbf{y}\}$ to the calculation are the DCT coefficient which, in the formula, are represented by $X(k)$ and $Y(k)$. C1 and C2 are pre-calculated constants needed to stabilize the calculation when means and variances get close to zero. In our algorithm, the two constants are based on the number of bits needed to represent the block DC coefficient.

$$SSIM\left(\mathbf{x},\mathbf{y}\right) = \left(1 - \frac{\left(X(0)-Y(0)\right)^2}{\left(X(0)^2+Y(0)^2+N\cdot C_1\right)}\right) \times \left(1 - \frac{\frac{\sum_{k=1}^{N-1}\left(X(k)-Y(k)\right)^2}{N-1}}{\frac{\sum_{k=1}^{N-1}\left(X(k)^2+Y(k)^2\right)}{N-1}+C_2}\right) \tag{2}$$

III. JPEG block simplification algorithm

When encoding baseline JPEG images, pixels are grouped into blocks of 8x8 pixels. When computed using GPU hardware, all 64 group pixels are computed in parallel. To calculate the SSIM index in the space domain, Equation 1 is used. The SSIM index can be computed from DCT coefficients [1], [7], which is a major benefit when encoding JPEG data using the proposed algorithm. In our implementation the formula in Equation 2 was used.

The JPEG encoding algorithm consists of the following steps, where entries highlighted in bold are additions proposed in this paper to increase run-length zeros:

1) Apply color transform and level shift.

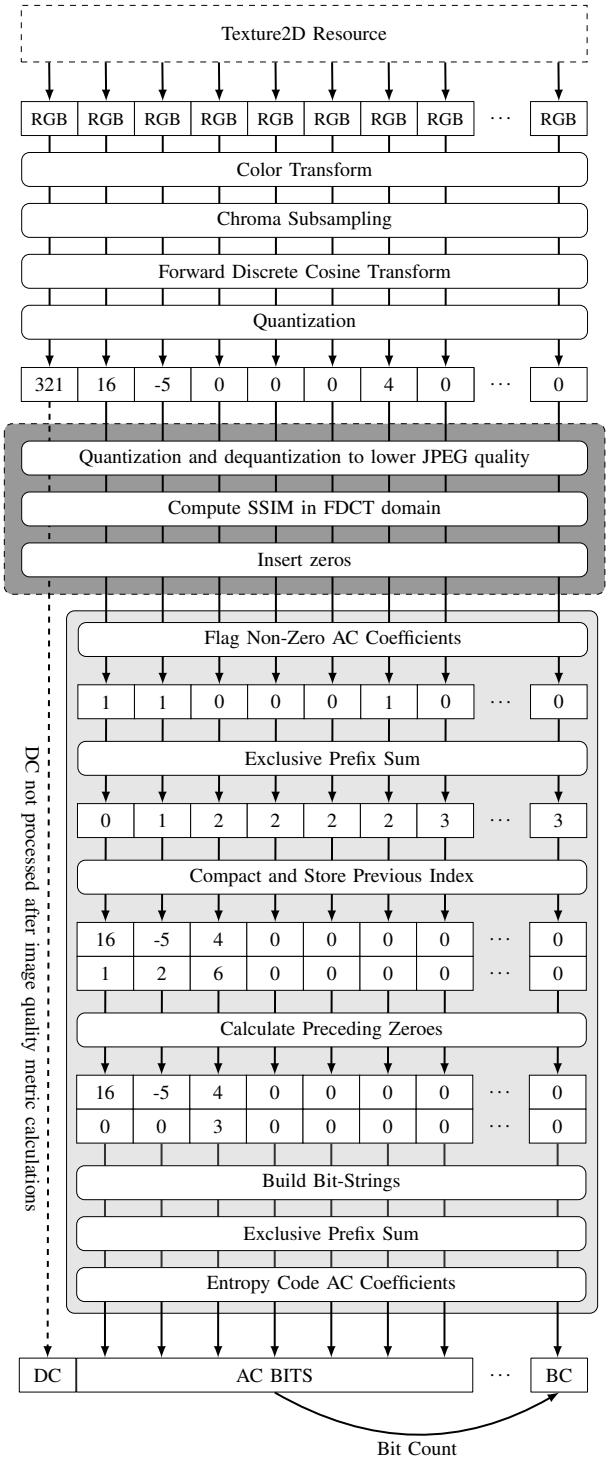


Fig. 3: An overview of the GPU implementation of our JPEG encoder. The highlighted three steps in the middle are modified as compared to [6].

2) Transform colors using the DCT.
3) **Quantize DCT coefficients using target JPEG quality and round to nearest integer value.**
4) **To simulate decoding; de-quantize and round all DCT coefficients to the nearest integer value.**

5) **Calculate a reference SSIM index from the DCT coefficients.**
6) **Iteratively identify the SSIM index below threshold boundary by applying binary search of the requested JPEG quality factor.**
    a) **Quantize and round current DCT coefficients to nearest integer value using the current iteration JPEG quality factor.**
    b) **De-quantize DCT coefficients and round to nearest integer value.**
    c) **Calculate SSIM index from the new DCT coefficients.**
    d) **Compare SSIM index with reference SSIM.**
    e) **If comparison stays above threshold value, then insert zeros in destination DCT matrix where zeros are present in the current iteration DCT matrix.**
7) Quantize destination DCT matrix.
8) Entropy code DCT coefficients and output to final JPEG data stream.

Figure 3 shows an overview of our GPU implementation of the proposed algorithm. The three highlighted steps in the figure are modified as compared to [6] as follows:

- Iteratively evaluate each 8x8 luma pixel block.
- Identify where to insert zeros by decreasing JPEG quality (simplification is guided by JPEG standard quantization matrix).
- Insert zeros until metric value stays above threshold value (using either an absolute or percentage difference).
- When below threshold, continue standard encoding scheme.

A standard JPEG image with quality 50 will have better quantitative metric results compared to an image encoded with quality 50 where SSIM simplification is enabled. To increase image quality, multiple higher quality encodes have to be done, until the simplified image has similar data size as the standard baseline image. Our implementation ensures that the resulting image encoding stays JPEG standard compliant, so existing decoders can be used.

IV. Experimental evaluation

We evaluate the algorithm on all images in the Kodak Lossless True Color Image Suite [5] and on two other test images. All measurements are done on a computer running Microsoft Windows 8.1 Pro x64, equipped with an Intel i7 860 CPU at 2.8 GHz, 8 GB RAM, and an AMD Radeon 7970 GPU. In all tests the chroma planes are subsampled according to the 4:2:0 subsampling scheme and the SSIM threshold ratio is set to 99%.

1) Reducing image size: Figure 4 shows how the proposed algorithm is able to reduce data size with a SSIM index preserved above the threshold ratio of 99%. This indicates that, when encoding low-quantized JPEG data, an unnecessary large amount of frequency information is stored. The PSNR value of the SSIM-based encoded image shows that the reduced image

(a) $2268x1512$ source RGB image (b) Baseline, Q90, 343,314 bytes

(c) Proposed, Q90, 197,641 bytes (d) PSNR diff (a) and {(b),(c)}.

Fig. 4: An example image comparison.

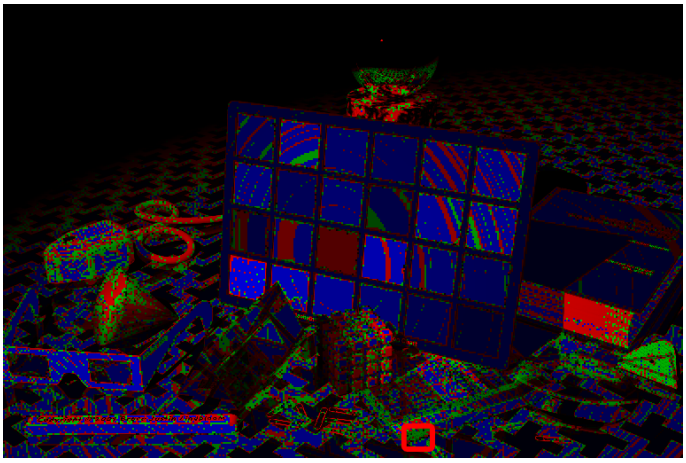
is worse as compared to the baseline encoded image. However, according to the SSIM index, the decreased quality will have low impact when presented to a human viewer. When encoding a baseline image with same size as the one in Figure 5b, the proposed technique has a higher SSIM index. Note that the SSIM-based encoded image is *significantly smaller*, i.e. 43% smaller, than the baseline encoded image.

2) Increasing image quality: When increasing the base JPEG quality the proposed encoder algorithm has the potential of maintaining that quality in sensible areas where the SSIM index has a steep decrease rate. Each 8x8 block is separately encoded and the total data size of all blocks are not known until the entire image is encoded. To redistribute image quality, the source image is re-encoded until the total data size is close or equal to the data size of the baseline encoded image.

To show the performance of the quality increasing behavior, a high-resolution (3072x2048 pixels) artificial image from www.imagecompression.info was encoded. The image was encoded using the baseline JPEG encoder and by the proposed algorithm. The encoded results, with similar data size output, are presented in Figure 5. Green areas shows areas with improved quality, red areas have decreased quality, and blue areas have the same quality in the baseline and the SSIM guided encoders.

When comparing image parts at 25x zoom level, as shown in Figure 5c and Figure 5d, we see that the proposed technique has potential to increase both qualitative and quantitative image quality in image areas that are sensible to quantization artifacts. Given the same JPEG data file size, the proposed SSIM guided algorithm could encode the image with JPEG quality 81 as compared to the baseline encoder that only could reach image quality 75.

3) Redistributing the image quality: To further evaluate the effectiveness of our simplification and quality redistribution approach, we have encoded all images from the Kodak Loss-



(a) PSNR comparison between (b) and {(c),(d)}. The green areas indicate parts of the image where the quality is improved, blue areas indicate maintained quality and red areas indicate where the image quality has been decreased.

(b) 3072x2048 source RGB image, 25x zoom

(c) Baseline, 25x zoom, Q75, 445,772 bytes

(d) Proposed, 25x zoom, Q81, 444,135 bytes

Fig. 5: An example image comparison. The green areas indicate parts of the image where the quality is improved, blue areas indicate maintained quality and red areas indicate where the image quality has been decreased.

less True Color Image Suite [5]. The suite consists of 24 images with dimension of 768x512 pixels. The encoding was done with the baseline JPEG encoder [6] and the proposed SSIM guided encoder. All images were encoded with JPEG qualities 50, 70 and 90. The encoded results were thereafter compared, using Peak signal-to-noise ratio (PSNR), to each suite reference image, respectively.

The comparison results are presented in the table of Figure 6. We start be observing that the differences between the baseline and the SSIM guided encoders increase with higher JPEG quality, as indicated by the PSNR difference for the whole JPEG image. Next, we can also observe that the variations in quality as compared to the original image decrease with higher JPEG quality, as indicated by the standard deviation.

The results indicate that the overall image quality has, according to PSNR, decreased. For example, the average PSNR value has decreased by 0.004 dB, 0.121 dB, and 1.537 dB for JPEG qualities 50, 70, and 90, respectively. However, in high contrast areas as indicated by SSIM indicated (and the human eye), the proposed algorithm improves PSNR on average by 0.172 dB, 0.311 dB, and 0.588 dB for JPEG qualities 50, 70, ad 90, respectively. In the best case, the PSNR has increased by up to 1.290 dB for image quality 90.

V. Conclusion

In this paper we have presented an algorithm for (i) maintaining or improving the quality in JPEG encoded images in high-contrast areas of the image, or (ii) significantly reducing the compressed image file size. We have implemented the algorithm in an efficient GPU-based JPEG encoder. We also present results where the potential of the simplification algorithm is shown, both with artificial and non-artificial source images. The results show that our algorithm improves the image quality in high-contrast areas, thus reducing JPEG encoding artifacts, and also increase the compression ratio in

	PSNR Baseline	PSNR Proposed	**Diff**	Baseline red area	Proposed red area	**Diff**	Baseline green area	Proposed green area	**Diff**
JPEG quality 50									
Avg dB	32,249	32,245	**-0,004**	32,491	32,194	**-0,297**	31,297	31,469	**0,172**
Min dB	27,601	27,566	**-0,096**	27,693	27,56	**-0,133**	27,582	27,628	**0,042**
Max dB	35,089	35,129	**0,083**	35,843	35,446	**-0,474**	34,261	34,582	**0,321**
Std dB	2,013	2,034	**0,050**	2,292	2,222	**0,093**	1,916	1,980	**0,088**
JPEG quality 70									
Avg dB	34,070	33,950	**-0,121**	35,055	34,276	**-0,779**	32,937	33,248	**0,311**
Min dB	29,763	29,789	**-0,39**	30,677	30,301	**-0,376**	29,408	29,572	**0,135**
Max dB	36,721	36,524	**0,146**	38,228	37,195	**-1,207**	35,898	36,357	**0,574**
Std dB	1,907	1,902	**0,123**	2,068	1,931	**0,245**	1,816	1,880	**0,152**
JPEG quality 90									
Avg dB	38,284	36,747	**-1,537**	38,323	36,042	**-2,281**	37,999	38,587	**0,588**
Min dB	35,514	33,332	**-3,073**	35,419	32,675	**-1,283**	35,815	35,911	**0,079**
Max dB	40,312	39,110	**-0,447**	40,851	38,715	**-3,630**	40,481	40,780	**1,290**
Std dB	1,466	1,635	**0,484**	1,582	1,639	**0,471**	1,452	1,484	**0,311**

Fig. 6: Encoding comparison of the Kodak Lossless True Color Image Suite, in which each image has the dimension 768x512 pixels. The PSNR is compared between the baseline JPEG encoder [6] and the proposed algorithm.

low contrast areas where we can have lower image quality without impacting the users' perception of the image.

Acknowledgment

This work is part of the research project "Scalable resource-efficient systems for big data analytics" funded by the Knowledge Foundation (grant: 20140032) in Sweden.

References

[1] S. S. Channappayya, A. C. Bovik, and R. W. Heath, "Rate Bounds on SSIM Index of Quantized Images," in *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 16241639, 2008

[2] A. Gupta, M. C. Srivastava, S. D. Pandey, and V. Bhandari, "Modified Runlength Coding for Improved JPEG Performance," in *International Conference on Information and Communication Technology (ICICT 07)*, 2007, pp. 235–237.

[3] Y. Jiang and M. S. Pattichis, "JPEG image compression using quantization table optimization based on perceptual image quality assessment," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 225–229.

[4] Y. Kai and J. Hongxu, "Optimized-SSIM Based Quantization in Optical Remote Sensing Image Compression," in *Sixth International Conference on Image and Graphics (ICIG)*, 2011, pp. 117–122.

[5] Kodak Lossless True Color Image Suite, http://r0k.us/graphics/kodak/.

[6] S. Petersson, "Real-Time JPEG Compression using DirectCompute," in *GPU Pro 4: Advanced Rendering Techniques*, Editor W. Engel, pp. 337–355, CRC Press, 2013.

[7] A. Rehman, Z. Wang, "SSIM-Inspired Perceptual Video Coding for HEVC," in *2012 IEEE Int'l Conf. on Multimedia and Expo (ICME)*, pp. 497–502, 2012.

[8] T. Richter, "Perceptual image coding by standard-constraint codecs," in *Picture Coding Symposium (PCS 2009)*,pp. 1–4, 2009.

[9] R. Rosenholtz and A. B. Watson, "Perceptual adaptive JPEG coding," in *Proc. of the 1996 International Conference on Image Processing vol 1*, pp. 901–904, 1996.

[10] G. A. Triantafyllidis, M. Varnuska, D. Sampson, D. Tzovaras, and M. G. Strintzis, "An efficient algorithm for the enhancement of JPEG-coded images," *Computers & Graphics*, **27**(4):529–534, Aug. 2003.

[11] C.-Y. Wang, S.-M. Lee, and L.-W. Chang, "Designing JPEG quantization tables based on human visual system," *Signal Processing: Image Communication*, vol. 16, no. 5, pp. 501–506, Jan. 2001.

[12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, **13**(4):600–612, April 2004.

[13] W. Wu and X. Zhang, "Code performance improvement scheme for X264 based on SSIM," in *3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 2012, pp. 396–400.

[14] C.-L. Yang, R.-K. Leung, L.-M. Po, and Z.-Y. Mai, "An SSIM-optimal H.264/AVC inter frame encoder," in *IEEE International Conference on Intelligent Computing and Intelligent Systems*, pp. 291–295, Nov. 2009.

[15] J. Zhao, Y. Shimazu, K. Ohta, R. Hayasaka, and Y. Matsushita, "A JPEG codec adaptive to region importance," in *Proceedings of the fourth ACM International Conference on Multimedia*, pp. 209–218, 1996.