On Improving the Performance of JPEG Image Compression at Low Encoding Bitrates *

Mohamed Kamel †
Quieintin Tang ‡
† Department of Systems Design Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
‡ NCR Canada
Waterloo,Ontario, Canada

Abstract

Still frame, gray and colour image compression is approaching an International Standard. Results from testing the algorithm on gray level images indicate that it performs well with respect to SNR and visual quality at sufficiently high encoding bitrates. At low bitrates, the reconstructed images suffer from visually impairing artifacts.

In this paper, we report the results of applying various preprocessing techniques to achieve an improvement in the image quality at low bitrates: Spatial Reduction, Gray Level Reduction, Block Rotation and Block Ordering. *By preprocessing, it is meant that the original image is processed in a particular manner before it is passed to the standard algorithm.*

Compression of the preprocessed images resulted in improved image quality at low bitrates as seen by the SNR and visual quality. The proposed Reduction and Block methods are transparent and compatible to the Standard's baseline algorithm.

1 Introduction

The image compression algorithm [1, 2, 3] brought forward by the Joint Photographic Experts Group (JPEG) is on its way to becoming approved as an International Standard in the area of still frame, gray and colour images. It offers advantages of reducing a digital image's storage requirement or transmission time. Practical applications include colour fax, medical imaging and image data bases. Other emerging and related standards cover binary images (JBIG) and motion pictures (MPEG).

JPEG's algorithm accommodates gray and colour images, multiple colour models, various pixel precisions and a wide range of modes of operation. A mode within JPEG's algorithm, called the Baseline, is the minimum functionality defined for all JPEG compression systems. It performs compression at a user-selectable level of compression and the degradation in image quality is usually minimal (at low compression rates) according to the observer. JPEG's algorithm is based on the Discrete Cosine Transform (DCT) [4, 5] working on disjoint 8x8 blocks within the image. The frequency coefficients produced by the DCT are quantized and subsequently encoded without loss to make up the compressed image data file. It is the quantization of the coefficients which gives rise to both compression and image degradation. The coarser the quantization, the more degraded the reconstructed image.

The decoder on the receiving side of the communication link is symmetrical and the inverse of the encoder.

A secondary source of error is in the implementation of the Forward DCT (FDCT) and the Inverse DCT (IDCT). Although one of the aims of JPEG is to have tractable computational complexity, JPEG did not restrict the FDCT nor IDCT to any particular implementation, thus permitting various fast algorithms [6, 7, 8]. Potentially, different implementations of the DCT may result in slightly different reconstructed images. Recently, Gordon and Redemann [9] proposed two test images to identify sources of inaccuracy within any DCT implementation.

This article presents the results of applying four methods for preprocessing the original input image with *Spatial Reduction, Gray Level Reduction, Block Rotation* and *Block Ordering* [10] in order for JPEG to obtain a high degree of compression yet maintain a level of image quality which is not objectionable to the viewer. An inverse process must be applied to the reconstructed image from the Standard's decoder. Compliance with the JPEG standard is maintained. Furthermore, the use of the methods is transparent to the Standard.

Section 2 discusses the motivation behind the use of Reduction and Block algorithms. Following that, Sections 3 4, 5 and 6 briefly outline the algorithms of *Spatial Reduction, Gray Level Reduction, Block Rotation* and *Block Ordering*, respectively. Results of the methods are covered in Section 7. Concluding remarks are provided in Section 8.

*This work is partially supported by the Natural Science and Engineering Research Council of Canada

2 Motivation

At compressed image representation bitrates of 1.0–2.0 bits per pixel (bpp), the reconstructed images are *excellent* or *indistinguishable* (from the original). In the range of 0.5–1.0 bpp, the images are *good.* When the bitrate is down around 0.08–0.5 bpp, the image quality is *moderate.* The compression ratio refers the bitrate back to the original 8 bits per pixel for gray level images.

Some applications may be able to make use of the low bitrate (high compression) feature but the candidates are few. The reconstructed images suffer from block artifacts at the low bitrate range because of two reasons. First, too much spatial information is lost when the high frequency FDCT coefficients are coarsely quantized. Secondly, 8x8 image pixel blocks are treated disjointly from others although the blocks may be adjacent.

The approach taken in this article is to apply preprocessing algorithms, such as *Spatial Reduction, Gray Level Reduction, Block Rotation* or *Block Ordering,* to images before inputting to the JPEG algorithm. Postprocessing of the reconstructed image is required to restore the effects of preprocessing. Whatever information may be lost in the preprocessing will be compensated by using JPEG at a compression level which preserves image quality quite well.

Reduction techniques transform an input image into a similar image which requires less bits per pixel to represent it. Obviously image degradation occurs. However, the fewer representation bits per pixel allows the JPEG algorithm to encode the reduced image at a much higher bitrate than if the image were not preprocessed. The benefits resulting from a higher bitrate are diminished block artifacts. When the input image representation is smaller than it normally is, an increase in the encoding bitrate will produce a compressed data file which is at par with the compressed data file from encoding the original image without preprocessing.

Block techniques rearrange the pixels within each 8x8 block before the block is passed to the JPEG encoder. The effect is that the arrangement of pixels produces FDCT coefficients which lead to higher image quality upon reconstruction.

3 Spatial Reduction

The approach of preprocessing using Spatial Reduction on the input image involves reducing the input size of the captured image and passing it to the encoder. The encoding process takes place according to the Standard.

On the decoder side, the compressed image data is reconstructed as normal. The reconstructed image will be smaller in size than the captured image. An inverse process (expansion) must be applied to return the reconstructed image to the full size of the original image.

The method for obtaining a smaller image size from an input image is the application of a commonly known subsampling technique. The operation is simple, fast and requires no excessive computation. Given the original image, the pixels of the subsampled image consist of every other pixel along a row or column from the original image. Subsampling from the original image must include the pixel from the upper left corner of the image by convention. In essence, the subsampled image compared with the original image is really every other pixel taken from the original.

The inverse operation to reduction is expansion; it must return the decompressed image to the original size. The method applied uses the *interpolative filter* mentioned in the JPEG standard.

For the described pair for reduction and expansion, the resulting image size passed to the encoder is 1/4 of the original. This allows the smaller image to be encoded at 4 times the desired encoding bitrate. For example, if the effective bitrate is to be 0.25 bpp, the actual encoding bitrate used by the JPEG compression algorithm can be 1.00 bpp.

4 Gray Level Reduction

This Reduction technique operates on a similar principle as Spatial Reduction. However, with Gray Level Reduction, the original image's gray level range is modified rather than its size.

The process of reducing the information[1] rate is carried out by a method called halftoning, introduced by Peleg, et al. [11, 12]. It takes the original image and essentially divides every pixel value by two after slightly altering the image. What will result is an image which is closest to the original, according to some distance measure. Halftoning can be applied successively to reduce the gray level value of each pixel by two, four, eight and sixteen.

Similar to Spatial Reduction, the altered, original image must be restored to its former structure after it has gone through compression and decompression. Expansion is simply carried out by a multiplication of the reduction factor. If, for example, halftoning is applied successively to the original image three times (dividing every pixel by eight), the expansion should multiply every pixel by a scalar value of eight. Thus, we use the terminology *3 lsb's lost* to describe the degree of Gray Level Reduction since it is like losing the information of the three least significant bits of every pixel.

With the aim of keeping the effective encoding bitrate the same as the desired rate, the use of Gray Level Reduction also increases the actual JPEG compression encoding bitrate. The increased bitrate depends upon the amount of halftoning applied to the original image. Assuming an 8 bit gray image and the degree *1 lsb lost*, the actual bitrate is $8 \div 7$ times the effective bitrate. Similarly, with *2 lsb's lost*, the actual bitrate is $8 \div 6$ times and so the pattern continues.

5 Block Rotation

Block Rotation transforms an 8x8 pixel block with a reversible operation, as opposed to Reduction schemes. The rows of the block or the columns are

[1]It is defined in [11] as the product of the number of pixel rows and pixel columns and the bits per pixel. For example, an 8 bit, 512x512 image has an information rate of 2,097,152 bits.

altered with successive applications of a primitive rotation operation.

Such a primitive takes a row or a column and moves it in its entirety to the row below or the column to the right, respectively. Rows or columns at the end of the block move to the top row or to the leftmost column of the block, respectively. The operation number is the number of times a single row or column primitive is applied. Thus an operation number of 0 means no rotation and an operation number of 7 means 7 rotations are done in succession to the same block. Other primitives can be derived from Block Rotation Right and Down; such as, rotation along the block's main diagonal and along its subdiagonal [10].

Given a default Block Rotation operation, every block in the image has the operation applied one of 0 through 7 times. Each time the block is transformed according to the operation and the number, the block is immediately passed through the FDCT, coefficient quantization, dequantization and IDCT. The SNR is recorded for that particular operation number. When all eight possible SNR values have been determined and the operation number with the highest SNR identified, the block is transformed according to that number. Encoding follows as usual but the operation number is also passed to the decoder.

The decoder will reconstruct a pixel block according to the standard. However, the associated operation number specifies an inverse Block Rotation operation to return the block to the original pixel arrangement.

The amount of additional information is minimal. The primitive operation can be assumed by default. Each block requires only 3 additional bits to identify the operation number associated with each block.

6 Block Ordering

Block Ordering is a challenging academic problem which offers dramatic improvement to JPEG's algorithm at low encoding bitrates. Unfortunately, no viable solution is known at the time of writing but the potential gains warrant further investigation. By virtue of being a Block technique, it is reversible. The transformation takes place on an 8x8 block before it is input to JPEG's algorithm.

The Block operation receives each block and sorts the 64 pixel values into descending order. All 64 original pixel locations are saved in an efficient form. The sorted values are then written into the same block from largest to the smallest value in an manner such as, for example, row major order (suitable for C implementations). Hence, the largest pixel (duplicate values permitted) is placed in the upper left corner of the block. The second largest is placed to the right of the first. The smallest pixel value would then be situated at the lower right corner.

The *sorted* block is then passed to the standard algorithm as usual. The decoder reconstructs the individual 8x8 blocks as usual. Original pixel location information received from the encoder by the decoder is used to *unsort* the pixels back to their original positions.

Block Ordering reduces the spatial frequency within the block of 64 pixels. The individual pixel values and the average pixel value are unaltered. However, by arranging the pixels in a row major order, for example, the spatial frequencies and consequently the FDCT AC coefficients are reduced. The use of FDCT coefficient quantization would force much of the higher order coefficients to zero. This property has been seen with sample images [10].

7 Results

Both Reduction techniques have been applied to a test image[2]. The results are compared against the straightforward use of the Baseline mode. It has been observed that both techniques serve to improve on the performance of the Baseline at low encoding bitrates. The improvements can be seen in the SNR. At bitrates greater than a threshold, the Baseline mode surpasses the Reduction techniques. This can be seen in the graphs of Figures 1 and 4.

Figure 2 shows an original of an image and the reconstructed image using the baseline algorithm. Examples of images processed with both Reduction schemes are given in Figures 3, 5 and 6. From a visual inspection of the Reduction preprocessed images, compared against the straight Baseline mode, Reduction methods serve to remove (or lessen) a visually annoying impairment such as block artifacts.

The results of Block Rotation showed marginal improvements over the standard algorithm. The performance plot can be seen in Figure 7.

A dramatic result can be seen in Block Ordering. For the sake of argument, each pixel's original position within its 8x8 block may require 6 bits to identify the position. The graph of Figure 8 shows the SNR of Block Ordering associated with the upper abscissa and the Baseline algorithm associated with the lower. If we can assume that the location information can be efficiently represented with a negligible amount of data, the lower abscissa scale on the graph would be appropriate for both curves and a much improved SNR is realized. In fact, the latter assumption is appropriate for highlighting the upper boundary of improvements to the SNR using Block Ordering.

8 Concluding Remarks

From the results presented, the use of Spatial Reduction and Gray Level Reduction significantly improves the SNR of the reconstructed images over the straight use of the Baseline mode. Improvement to visual quality can be seen as well. All the benefits are seen at the low, effective compression bitrates. After a certain bitrate threshold, the Baseline excels.

Spatial Reduction performs well in lessening visual impairment caused by block artifacts. The side effect is that the reconstructed images are blurry and suffer from a loss of edge definition due to the use of the interpolative filter; low pass effect on the image.

Another added advantage of Spatial Reduction is that the Baseline encodes a preprocessed image which is 1/4 of the original size. This very fact leads to a savings in execution of the Baseline mode. There is

[2] The work in [10] included two images from the JPEG test bed and a number of bank cheque images.

merely 1/4 of the computation of the FDCT, coefficient quantization, entropy encoding, entropy decoding, coefficient dequantization and IDCT.

Gray Level Reduction also results in a gain in SNR and visual quality. Again the gain is at low, effective encoding bitrates and tends to fall below the Baseline system after a certain threshold. With the degree of reduction at *1 lsb lost*, the improvement in SNR and visual quality is not very significant because of the marginal increase in the encoding bitrate. At the other end, reduction at *4 lsb's lost* leaves the reconstructed image with false contours because the reconstructed image has only 16 remaining gray levels.

The use of Gray Level Reduction does not reduce the execution time of the Baseline. In fact, added processing on top of the Baseline mode is required to perform the reduction and expansion aspects of the algorithm.

The use of Block Rotation always found the optimal improvement in SNR, according to the method, over the use of the Baseline mode. However, the gain is marginal. The other drawback is that the method is computationally intensive in the encoder in determining the optimal Block Rotation operation number for each block. The decoder's computational intensity is much less of an increase.

Block Ordering offered the most potential in improvements to the SNR and visual quality . At the time of writing, there is no efficient implementation to pass each pixel's original location from the encoder to decoder. If such information can be stored with negligible data, then the plot of the SNR, as seen earlier, defines the upper bounds on the gain in SNR.

References

[1] G. P. Hudson, H. Yasuda and Istvan Sebestyen, "The International Standardisation of a Still Picture Compression Technique," *Globecom '88*, Florida, pp. 1016–1021.

[2] J. L. Mitchell, W. B. Pennebaker and C. A. Gonzales, "The Standardization of Color Photographic Image Data Compression," *SPIE/SPSE Symposium on Electronic Imaging: Advanced Devices and Systems*, Jan. 15–20, 1989.

[3] "Digital Compression and Coding of Continuous-Tone Still Images, Part 1, Requirements and Guidelines," ISO/IEC JTC1 Committee Draft 10918-1, Mar. 1991.

[4] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Commun.*, vol. COM-23, Jan. 1974, pp. 90–93.

[5] N. Ahmed and K. R. Rao, "Orthogonal Transforms for Digital Signal Processing," *Springer-Verlag*, New York, 1975.

[6] W. Chen, C. H. Smith and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Commun.*, vol COM-25, Sept. 1977, pp. 1004–1009.

[7] M. A. Haque, "A Two-Dimensional Fast Cosine Transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, Dec. 1985, pp. 1532–1539.

[8] H. S. Malvar, "Fast Computation of the Discrete Cosine Transform and the Discrete Hartley Transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, Oct. 1987, pp. 1484–1485.

[9] H. Gordon and E. Redemann, "Minimizing Chaos-like Behavior in JPEG Implementations: The Case for Analytically Derived Test Images," *Electronic Imaging West '91*, Anaheim, California, Apr. 1991.

[10] Q. Tang, "Still Frame, Gray Level Image Compression With The Discrete Cosine Transform," Master's thesis, University of Waterloo, Oct. 1990.

[11] S. Peleg, M. Werman and H. Rom, "A Unified Approach to the Change of Resolution: Space and Gray-Level," *IEEE Trans. Patt. Rec.*, vol. PR-7, July 1989, pp. 739–742.

[12] M. Werman and S. Peleg, "Halftoning As Optimal Quantization," *Proc. Eighth Int. Conf. Pattern Recog.*, Paris France, Oct. 1986, pp. 1114–1116.

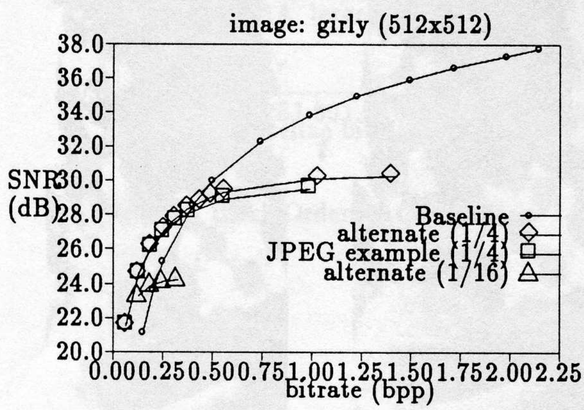

Figure 1: Spatial Reduction Comparison

(a) Original (b) Baseline

Figure 2: Original and Baseline Images

(a) JPEG 1/4 (b) Altern 1/4

Figure 3: Spatial Reduction

Effective Encoding Bitrate = 0.25 bpp

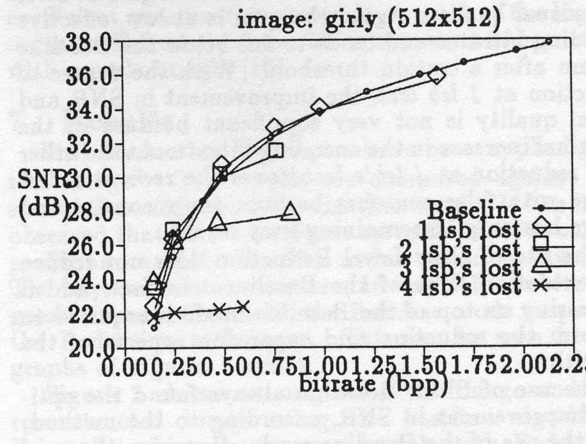

Figure 4: Gray Level Reduction Comparison

(a) 1 lsb lost (b) 2 lsb's lost

Figure 5: Gray Level Reduction (1 & 2 lsb's lost)

Effective Encoding Bitrate = 0.25 bpp

(a) 3 lsb's lost (b) 4 lsb's lost

Figure 6: Gray Level Reduction (3 & 4 lsb's lost)

Effective Encoding Bitrate = 0.25 bpp

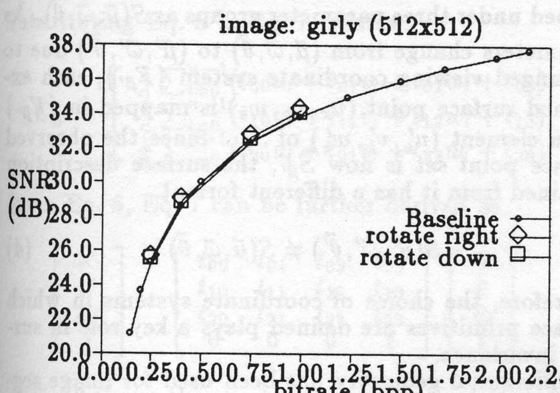

Figure 7: Block Rotation Comparison

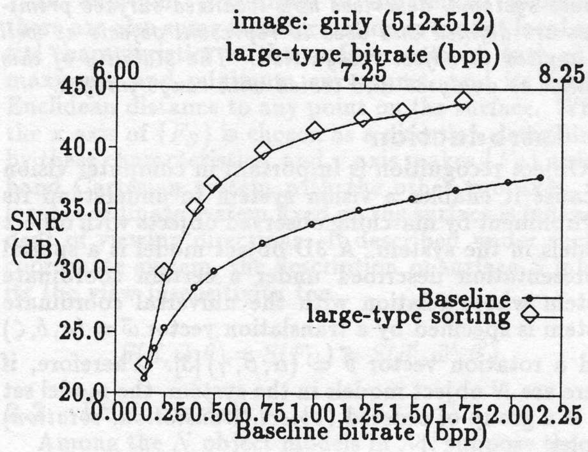

Figure 8: Block Ordering Comparison