Further Lossless Compression of JPEG Images

Ingo Bauermann and Eckehard Steinbach
Institute of Communication Networks, Technische Universität München, Munich, Germany
{ingo.bauermann,eckehard.steinbach}@tum.de

ABSTRACT

In this paper we present an algorithm to further compress JPEG images without loss. The main contribution of this work is a sorting transform that allows reducing inter-block redundancy not exploited by the standard JPEG still image compression algorithm. Inter subband redundancy is also reduced by grouping of coefficients with similar statistics and entropy encoding in adaptive contexts. Tested on randomly selected JPEG images from the Internet, our approach reduces image file size on average by 14%.

1. INTRODUCTION

Further compression of JPEG images is motivated by the fact that there is an interest in storing and archiving images on portable devices like USB-memory sticks and other memory cards efficiently and with no deviation from the originally captured version. Still images taken with digital cameras are commonly stored in standard JPEG format [1]. The rate-distortion optimization used in such devices is constrained by the computational complexity of common optimizing algorithms.

However, for post-processing tasks and archiving, PC workstations are used, where further and more complex optimizations can be performed. Most optimization algorithms proposed in the literature use requantization techniques to reduce the rate but they also hurt image quality (see e.g. [2]). To achieve lossless further compression of JPEG images, lossless encoding steps on the DCT transform coefficients are needed. One common technique is the arithmetic coding option proposed by JPEG [1], as most JPEG images are encoded using Huffman coding. A reduction of file size of about 7-10% can be achieved for lossless rate optimization. The approach in [3] also achieves an average gain of about 7%. The authors group coefficients according to some heuristic rules and recursively encode the groups using adapted huffman tables.

Our approach is designed to reduce inter-block redundancy caused by independently encoding eight-by-eight pixel blocks and to exploit redundancy between frequency bands within one block. This is achieved by encoding transform coefficients with similar neighborhood properties using the same context for adaptive arithmetic coding.

Assume all luminance coefficients reordered to 64 subimages, each containing coefficients from the same of the 64 subbands corresponding to the 2D DCT basis functions. Figure 1 shows a part of the DCT transform coefficient image (luminance component) of the well known ‘lena’ test image. Nine subimages are shown from the lowest three frequencies in vertical and horizontal direction. Dark values denote large absolute values. Bright values denote small coefficient values.

As can be seen from Figure 1 two main types of redundancy between coefficients can be exploited after the DCT-Transform and quantization.

- Between coefficients representing the same subband in neighboring blocks typically a significant statistical dependency exists. Plane areas or connected areas with a high variance of the coefficients can be identified. Standard JPEG exploits this redundancy to some extent only for the DC coefficients by differential encoding.

- Also between coefficients representing the same image block in neighboring subbands a strong statistical dependency exists. The structure of the image can be identified in all subbands (see Figure 1). Standard JPEG exploits this redundancy by using the end of block symbol (EOB) for zero coefficients and run level coding for non zero coefficients. No statistical modeling is done.

Our approach tries to further compress JPEG-encoded images by exploiting the remaining redundancy. The main idea is to group coefficients with similar statistics and encode those groups in the same adaptive context. To determine these neighborhood properties and corresponding contexts, a sorting transformation is used. A drawback of our approach is that blocks can not be decoded separately anymore.

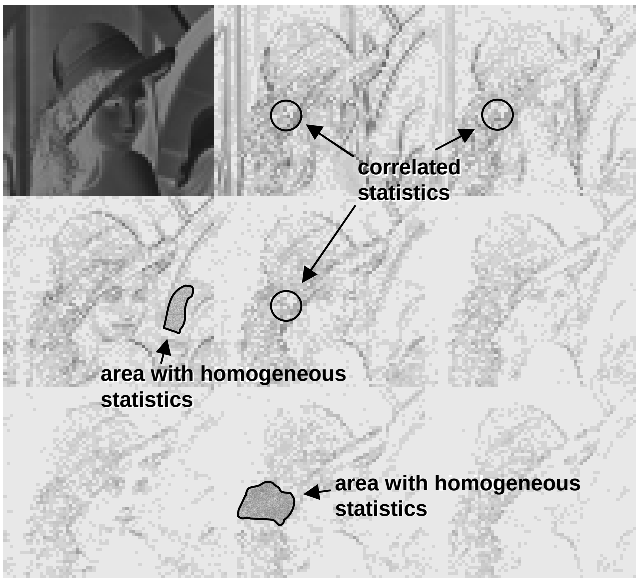

Figure 1. The upper left part of the luminance coefficient image of 'lena'. Marked areas denote examples of interblock areas with similar statistics (gray) and examples of intrablock areas with correlated statistics (circled).

The remainder of the paper is structured as follows. In Chapter 2 the sorting transform used in our algorithm is described. In Chapter 3 the algorithm to further losslessly compress JPEG images is investigated. In Chapter 4 results using our approach are presented. Chapter 5 gives conclusions and closes the paper.

2. SORTING TRANSFORMATION

The transcoder from standard JPEG to the format we propose in this work processes the transform coefficients retrieved from the original JPEG bitstream. The coefficients are scanned in zig-zag order within each block. The resulting row vectors are concatenated in block-wise raster scan order to a representation as illustrated in Figure 2.

Block-Nr.	DC	AC1	AC2	...	AC63
1	-3	0	0	...	0
2	-10	2	-1	...	0
3	-10	1	0	...	0
4	-5	1	0	...	0
5	-1	0	-1	...	0
6	1	1	0	...	0
7	0	0	0	...	0
8	0	2	0	...	0
9	1	0	-1	...	0
10	0	0	0	...	0
...	...	...	...	...	...

Figure 2: Reorganized transform coefficients for 10 blocks (raster scan order) and selected coefficients ('lena' at medium quality).

The absolute values in each column are now sorted in ascending order. This order is not applied to the coefficients in the column itself but to the coefficients of the next higher order (zig-zag scan index). This means that each column in Figure 2 is sorted according to the order of the column to the left. Figure 3 illustrates this procedure:

1. All coefficients from one subband are concatenated in one column.

DC	AC1	AC2
3	0	0
-10	2	-1
-10	1	0
-5	1	0

DC	AC1	
3	0	
5	1	
10	2	
10	1	

2. The absolute magnitude of AC1 is ordered in the ascending order of the DC-coefficients (magnitude).

3. AC2 (column 3) is ordered in the ascending order of AC-coefficient 1 (column 2) and so on.

	AC1	AC2
	0	0
	1	0
	1	0
	2	-1

...

DC	AC1	AC2
3	0	0
-10	1	0
-10	2	0
-5	1	-1

4. The differentially encoded DC coefficients are stored in their original order. All AC coefficients are stored in their new order.

Figure 3: Sorting transformation used to exploit inter-block redundancy.

The mapping can be simply inversed, as the first column is transmitted without reordering. This allows for determining the inverse mapping for the second column. Once the second column is reconstructed the inverse mapping for the third column can be determined. And so on.

The sorting transform produces areas where ideally large and small coefficients (in practice: variance of absolute values) are accumulated in connected areas. Figure 4 shows a grayscale representation of the magnitude of transform coefficients of the luminance component before and after the sorting transformation.

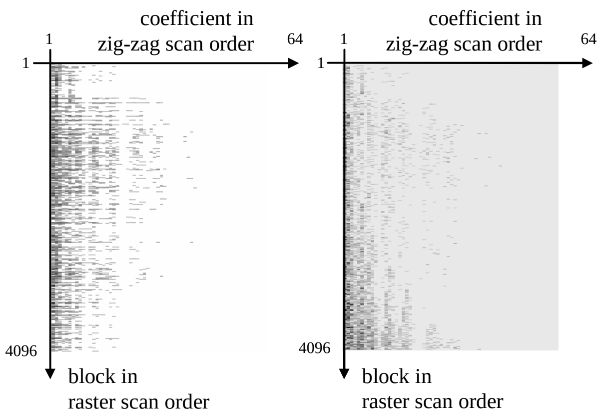

Figure 4: (left) Reorganized coefficient image of ‘lena’ without ordering. (right) Reorganized coefficient image of ‘lena’ after the sorting transform. For the latter case transform coefficients with large magnitude are accumulated in the lower left corner. Coefficients with large magnitude are denoted as dark values.

3. PARTITIONING AND ENTROPY CODING

To exploit the new order of the coefficients a partitioning step is performed to separate areas with similar statistical properties. One approach for this partitioning is illustrated in Figure 5. For each of the partitions maximum values are stored with the bit stream to minimize loss in bit rate due to inaccurate finite precision modeling. A special “zero-partition”-symbol is used to indicate that there are no non-zero coefficients in one particular partition. Partitions containing non-zero values are encoded using a context adaptive arithmetic coding scheme.

This scheme exploits inter block redundancy as coefficients in neighboring blocks and blocks with similar statistics are grouped and encoded in the same adaptive context. Redundancy between subbands of a particular block is reduced in the same way.

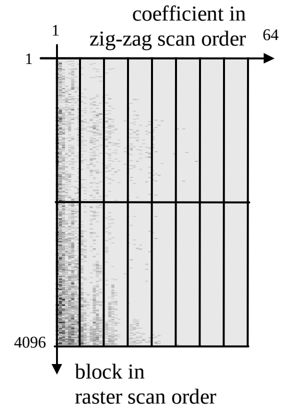

Figure 5: Example of partitioning performed in our approach for statistical modeling and entropy coding. Every block bounded by black lines is encoded separately using context adaptive arithmetic coding.

4. RESULTS

In the first experiment we tested various linear partitioning configurations. Figure 6 shows compression results for various horizontal and vertical subdivisions and for the testimage ‘lena’. The best result is achieved for partitions containing two neighboring subbands each containing one half of the ordered blocks.

No. of partitions / coeff. / blocks	64	32	16	8	4
1	83.6%	83.1%	84.3%	86.0%	90.5%
2	84.2%	**82.6%**	83.5%	85.0%	89.6%
4	87.5%	83.7%	83.4%	84.4%	88.9%
8	94.0%	87.5%	85.0%	84.7%	88.9%

Figure 6: Transcoded file size compared to original file size for different linear partitioning algorithms on test image ‘lena’ (medium quality). Horizontally the number of divisions for DCT-coefficient numbers is shown. The rows denote the number of divisions for the block numbers of all blocks of the test image.

The second experiment was performed to determine the dependency of the proposed algorithm on the quality of the original JPEG-image. Figure 7 shows the rate-distortion plot for the test image 'sail' encoded using standard JPEG and further compressed using our algorithm. Header information is not included for determining the results. The reduction is about 12% for all rates.

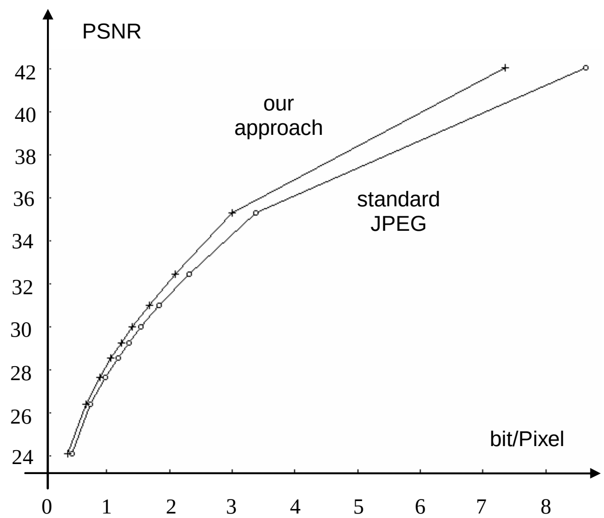

Figure 7: Rate distortion plot for standard JPEG (circles) and our approach (crosses) for the test image 'sail'. Our approach achieves a file size reduction of about 12% at all rates.

In the third experiment our approach is compared to standard JPEG. For a large number of test images an average of 14% smaller file size is achieved for 200 randomly selected JPEG files from the Internet. The result is illustrated in Figure 8. The algorithm performs badly on large multiple-pass optimized high quality JPEG-files using custom quantization and custom huffman tables. Header information was about 1% on average.

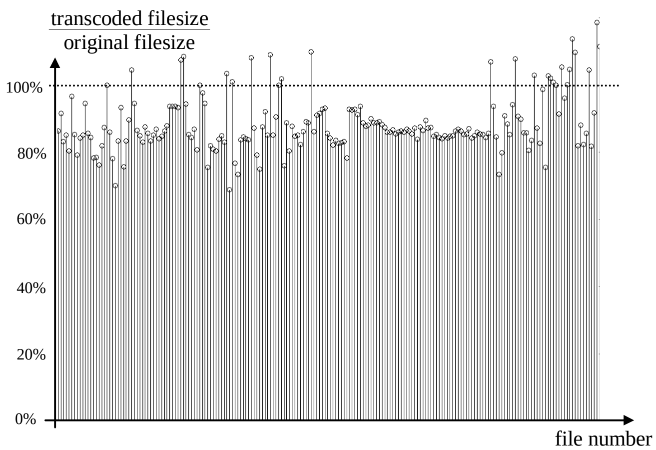

Figure 8: Transcoding gain for a test set of 200 JPEG-images with various sizes and qualities.

For the test image "smandril" a rate-distortion analysis is performed. The result is illustrated in Figure 9. The D-R curve for JPEG2000 [4] is also plotted for comparison reasons. Note that this rate-distortion curve results from directly encoding the source image, not from lossless transcoding of already encoded JPEG images.

The test image has about twice the file size compared to JPEG2000 for lower bit rates. Our lossless optimization approach leads to a further compression gain over the entire rate range and significantly decreases this gap. The observed gains are larger at low rates for this test image. Note that corresponding optimization pairs for JPEG and our approach have the same PSNR while the rate required is reduced.

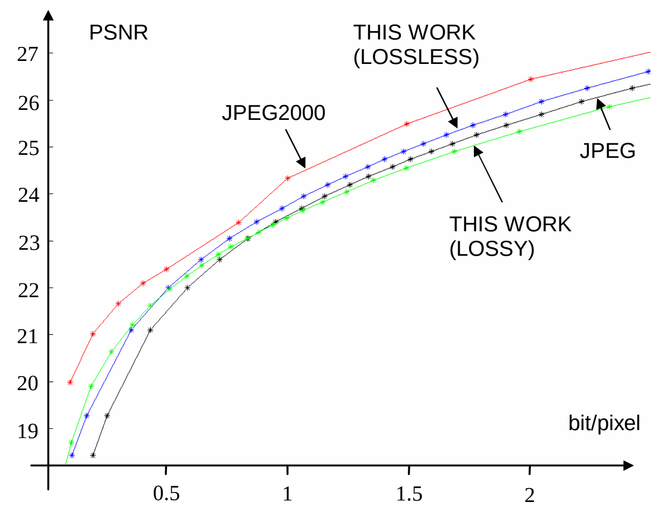

Figure 9. Rate-distortion plots for test image "smandril" (or "baboon"). Standard JPEG, JPEG2000, our lossless algortihm, and our lossy extension are compared. For JPEG2000 direct encoding of the source image is used. Whereas our lossless and lossy approach further compresses the JPEG bitstream.

A last experiment was performed, investigating a lossy extension for our optimization approach. In this experiment, a requantization is performed. Selected coefficients are set to zero to optimize the performance of the partitioning and arithmetic coding step. The result for simply introducing a deadzone for requantization is also shown in Figure 9. The lossy extension approach performs very well for low bitrates.

5. CONCLUSIONS

In this paper we investigate further lossless compression of JPEG-images using a sorting transform. This sorting transform is able to segment DCT transform coefficients to groups with similar statistical properties. Arithmetic coding is used as entropy coding scheme. Gains of up to 30% and an average of 14% less file size on randomly selected images from the Internet are achieved. Our algorithm works best on low and high quality images.

6. REFERENCES

[1] Joint Photographic Experts Group, "Various Documents and Links," http://www.jpeg.org.

[2] H.H. Bauschke, C.H. Hamilton, M.S. Macklem, J.S. McMichael, and N.R. Swart: "Recompression of JPEG images by Requantization," IEEE Transactions on Image Processing 12(7), pp. 843-849, 2003.

[3] Karl Skretting, John Håkon Husøy and Sven Ole Aase, "Improved Huffman coding using recursive splitting," Proc. of Norwegian Signal Proc. Symp., Asker, Norway, Sep. 1999.

[4] David S. Taubman and Michael W. Marcellin, "JPEG 2000: Image Compression Fundamentals, Standards and Practice," Kluwer International Series in Engineering and Computer Science, Secs 642, Nov. 2001.