Integer Reversible Transformation to Make JPEG Lossless*

Ying Chen Pengwei Hao
Center for Information Science, Peking University, Beijing, 100871, China
Department of Computer Science, Queen Mary, University of London, E1 4NS, UK
{chenying, phao}@cis.pku.edu.cn

Abstract

JPEG, as an international image coding standard based on DCT and Huffman entropy coder, is still popular in image compression applications although it is lossy. JPEG-LS, standardized for lossless image compression, however, employs an encoding technique different from JPEG. This paper presents an integer reversible implementation to make JPEG lossless. It uses the framework of JPEG, and just converts DCT and color transform to be integer reversible. Integer DCT is implemented by factoring the float DCT matrix into a series of elementary reversible matrices and each of them is directly integer reversible. Our integer DCT integrates lossy and lossless schemes nicely, and it supports both lossy and lossless compression by the same method. Our JPEG can be used as a replacement for the standard JPEG in either encoding or decoding or both. Experiments show that the performance of JPEG with our integer reversible DCT is very close to that of the original standard JPEG for lossy image coding, and more importantly, with our transform, it can compress images losslessly.

1. Introduction

Although wavelet-based JPEG-2000 has become the new generation of image compression standard [1, 6], many successful image compression applications still benefit from the previous standard, discrete cosine transform (DCT) based JPEG. It was proved theoretically in [3] that, for a commonly used class of source models, DCT is the optimal K-L transform in the limiting case that the adjacent element correlation tends to unity. Many transform coding standards, such as MPEG-1, MPEG-2 and H.263, are also based on DCT.

Originally, the float DCT is lossy and not integer reversible, although many applications require lossless compression. After JPEG standardization, ISO also standardized a technique as "lossless JPEG", which is very different from the DCT-based JPEG and is named JPEG-LS [2]. Yet some other applications demand both lossy and lossless compression or even progressive compression from lossy to lossless.

DCT-based JPEG is lossy because the three necessary processes in the image compression are all lossy: color transform, DCT and quantization.

* Supported by FANEDD China, under Grant 200038.

0-7803-8406-7/04/$20.00 © 2004 IEEE.

For each block of the image, if we use integer reversible DCT (RDCT) instead of the original floating-point DCT used in JPEG and then use lossless quantization, lossless compression can be achieved in the block. Multi-component images, such as color images, can be compressed losslessly with the reversible component transform (RCT).

In this paper, a system named integer JPEG (iJPEG) is proposed to implement the integer reversible JPEG. The system works the same as DCT-based JPEG but uses RDCT. Advantages of iJPEG and the compatibility between iJPEG and JPEG are also studied.

2. Integer Reversible DCT

According to paper [5], there are at most three triangular elementary reversible matrices (TERMs) or at most N+1 single-row elementary reversible matrices (SERMs) for an nonsingular $N \times N$ matrix factorization, except for a possible permutation matrix in the TERM or SERM factorization. The authors presented a factorization of DCT as an example for reversible integer mapping.

The transform matrix of the eight-point DCT used in JPEG is:

$$A=\begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1918 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & -0.3536 \\ 0.2778 & 0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

The matrix satisfies det A=1. Therefore, it can be factorized into SERMs:

$$P_L^{\ T} A P_R^{\ T} = S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$$

where $S_m = I + e_m s_m^{\ T}, m = 1,2,\cdots 8, S_0 = I - e_8 s_0^{\ T}$ and e_m is the m-th standard basis vector formed as the m-th column of the identity matrix. The vectors s_m to make SERMs are:

$$\begin{bmatrix} S_0^T \\ S_1^T \\ S_2^T \\ S_3^T \\ S_4^T \\ S_5^T \\ S_6^T \\ S_7^T \\ S_8^T \end{bmatrix} = \begin{bmatrix} -1.1648 & 2.8234 & -0.5375 & 0.6058 & -1.2228 & 0.3805 & -0.0288 & 2 \\ 0 & -1.1129 & 0.0570 & -0.4712 & 0.1029 & 0.0156 & -0.4486 & -0.4619 \\ -0.0685 & 0 & 0.2708 & -0.2708 & -0.2235 & 0.2568 & -0.3205 & 0.3841 \\ -0.0364 & -1.7104 & 0 & -1.0000 & 0.3066 & 0.6671 & -0.5953 & 0.2039 \\ 0.7957 & 0.9664 & 0.4439 & 0 & 0.6173 & -0.1422 & 1.0378 & -0.1700 \\ 0.4591 & 0.4108 & -0.2073 & -1.0824 & 0 & 0.7071 & 0.8873 & -0.2517 \\ -0.6573 & 0.5810 & -0.2931 & -0.5307 & -0.8730 & 0 & -0.1594 & -0.3560 \\ 1.0024 & -0.7180 & -0.0928 & -0.0318 & 0.4170 & 1.1665 & 0 & 0.4904 \\ 1.1020 & -2.0306 & -0.3881 & 0.6561 & 1.2405 & 1.6577 & -1.1914 & 0 \end{bmatrix}$$

The corresponding permutation matrices are:

$$P_L=\begin{bmatrix}0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&1&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\end{bmatrix}\qquad P_R=\begin{bmatrix}0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&1&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&1&0&0\end{bmatrix}$$

Suppose an integer vector is $x=(x_1,x_2,\cdots x_8)^T$, the integer implementation of $y=Ax$ can be:

$$y'=P_L S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 P_R x$$

and its inverse is:

$x'=P_R S_0^{-1} S_1^{-1} S_2^{-1} S_3^{-1} S_4^{-1} S_5^{-1} S_6^{-1} S_7^{-1} S_8^{-1} y'$, where x' is the reconstructed vector of x, $S_m^{-1}=I-e_m s_m^T$ $(m=1,2,\cdots,8)$,and $S_0^{-1}=S_0=I-e_8 s_0^T$ are SERMs

For each SERM, S_m, its reversible integer transform can be implemented in-place for *m*-th element only:

$$y'_m=x_m+\left[\sum_{n\neq m}s_{mn}x_n\right]=x_m+[b]$$

where $m=0,1,2,\cdots,8$.

Its reverse is as simple as it:

$$x_m=y'_m-\left[\sum_{n\neq m}s_{mn}x_n\right]=y'_m-[b]$$

where $m=8,7,6,\cdots,1,0$.

Actually, we also have

$$\begin{aligned}A&=P_L S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 P_R\\&=(P_L P_R)(P_R^T S_8 P_R)(P_R^T S_7 P_R)\cdots(P_R^T S_1 P_R)(P_R^T S_0 P_R)\\&=(P_L S_8 P_L^T)(P_L S_7 P_L^T)\cdots(P_L S_1 P_L^T)(P_L S_0 P_L^T)(P_L P_R)\end{aligned}$$

Therefore, we only need simpler computations than that proposed in [5]. The factorization for RDCT we can use is:

$$A=PS_{k_8}S_{k_7}S_{k_6}S_{k_5}S_{k_4}S_{k_3}S_{k_2}S_{k_1}S_{k_0}\text{ or}$$

$$A=S_{k_8}S_{k_7}S_{k_6}S_{k_5}S_{k_4}S_{k_3}S_{k_2}S_{k_1}S_{k_0}P$$

where $P=P_L P_R$, $k_0=k_N$, and $k_N,k_{N-1},\cdots,k_2,k_1$ is a permutation of $N,N-1,\cdots,2,1$.

For lossless RDCT and its lossless reverse, we have $x=x'$. We define $Error=y-y'$ as the error of the RDCT that results from rounding arithmetic operations.

3. Integer JPEG System

Our integer JPEG (iJPEG) works the same as DCT-based JPEG. The only difference between them is that RDCT takes the place of the original float DCT in the encoding phase and the inverse of RDCT replaces the IDCT in the decoding phase.

Similar as depicted in [7], our iJPEG encoder and decoder are shown in Figures 1 and 2 respectively.

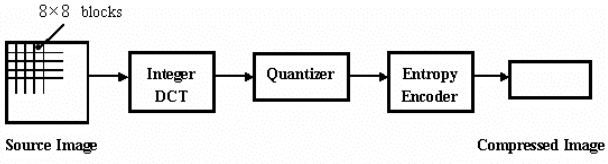

Figure 1 iJPEG Encoder

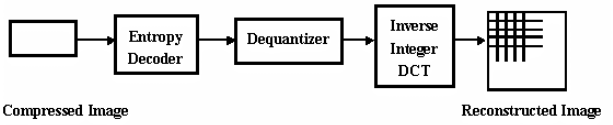

Figure 2 iJPEG Decoder

When a source image is encoded by iJPEG, each 8×8 block of a component is transformed by RDCT. Then, some insignificant information may be discarded by the quantizer. This process may be lossy or lossless. Huffman entropy coding which follows the quantization process is lossless. If the quantizer is lossless, the reconstructed image will be the same as original source image. Our lossless quantizer uses a quantization matrix whose elements are all 1's.

iJPEG uses the same file format of compressed images as that of JPEG files. Therefore, our iJPEG-compressed images can be decoded by JPEG decoder and JPEG files can also be decoded by iJPEG decoder.

The integer transform produces an error brought in by each SERM step, $y'=P\lfloor S_8\cdots\lfloor S_2\lfloor S_1\lfloor S_0 x\rfloor\rfloor\rfloor\cdots\rfloor$ ($\lfloor\ \rfloor$ denotes the rounding arithmetic). Our SERMs are carefully chosen so that the error is not so large.

For color images, a color transform is used as a pre-process before intra-component coding.

A commonly-used color transformation is from RGB to YCrCb. YCrCb is adopted by JPEG and JPEG 2000 [6]. The forward transform and its inverse formulae are:

$$\begin{bmatrix}Y\\Cr\\Cb\end{bmatrix}=\begin{bmatrix}0.299&0.587&0.114\\0.500&-0.419&-0.081\\-0.169&-0.331&0.500\end{bmatrix}\begin{bmatrix}R\\G\\B\end{bmatrix}\qquad\begin{bmatrix}R\\G\\B\end{bmatrix}=\begin{bmatrix}1.000&-0.000&1.402\\1.000&-0.344&-0.714\\1.000&1.772&0.000\end{bmatrix}\begin{bmatrix}Y\\Cr\\Cb\end{bmatrix}$$

It's the only color space to de-correlate the original tristimulus color components in JPEG. However, as a matter of fact, the determinant of above forward transform matrix is 0.2363, less than 1. Therefore, YCrCb not only de-correlates the original color components but also discards some information to get lower entropy. The discarded information cannot be recovered, so the transform is lossy.

A lossless color image compression method has to employ some RCT or use no color transform at all (RGB). Some RCTs can be used in our iJPEG for lossless color image compression, but the compatibility between iJPEG and JPEG will be weakened once other color transform is used.

An approximated reversible color transform, accepted by JPEG2000, named ORCT in this paper, is given in [4]:

$$\begin{cases} Y = \lfloor (R + 2G + B)/4 \rfloor \\ Cr = R - G \\ Cb = B - G \end{cases}$$ and inverse: $$\begin{cases} G = Y - \lfloor (Cr + Cb)/4 \rfloor \\ R = Cr + G \\ B = Cb + G \end{cases}$$

4. Experiments

We use PSNR as a metric for lossy image quality evaluation and bit rate for lossless compression assessment. For lossy compression evaluation at the same bit rate, if the PSNR of an iJPEG-compressed image is not lower than the PSNR of the JPEG-compressed image, iJPEG is at least not worse than JPEG.

In order to evaluate the image quality of the both method comparing to the original image, we apply iJPEG and standard JPEG to some 8-bit standard test images, such as, Lena, peppers, mandrill, and Barbara. Then we measure the image quality of the reconstructed images.

To measure the compatibility between iJPEG and JPEG, we use a number of JPEG-compressed images found from the internet, games and acquired with scanners, and we decompress them with both iJPEG and JPEG. For each JPEG image, the iJPEG-reconstructed and the JPEG-reconstructed versions are compared to obtain a PSNR.

As a lossless image compression system, iJPEG is also compared with other lossless systems, JPEG-LS and JPEG-2000. The assessment metric is the lossless bit rate (bits per pixel, bpp). For lossless compression, we also compare the bit rate to the best lossy performance of standard JPEG and to show the error (PSNR) between JPEG and iJPEG.

4.1. Image Quality Comparing to the Original Images

In order to compare the decompressed image with the original image, four different compression system combinations can be employed in the encoding phase and the decoding phase: (1) Both encoding and decoding are with JPEG; (2) Encoding with JPEG and decoding with iJPEG; (3) Encoding with iJPEG and decoding with JPEG; (4) Both encoding and decoding are with IJPEG.

Therefore, for a specific source image and a given quantizer or quality specification, we can have 4 reconstructed images. Then, PSNRs are measured between each reconstructed image and the original.

Experiments with Lena (Figure 3) show that the reconstructed images have nearly the same image quality. Since PSNRs of lossy experiments are all less than 50, in Figure 3, the largest PSNR of the fouth curve is actually infinity, which indicates lossless reconstruction.

Curves of PSNR versus compression ratio in our experiments with other test images are all similar. IJPEG and JPEG give very close compression performance, and more importantly, iJPEG enables lossless compression.

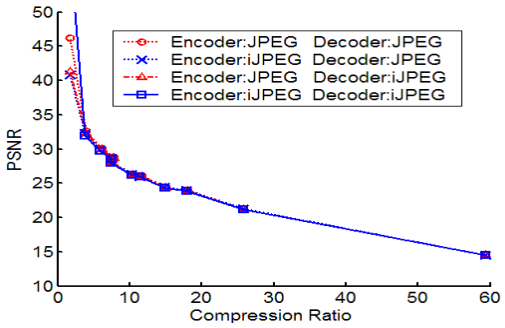

Figure 3 Compression quality comparing to the original image (Lena)

4.2. Comparison and Compatibility between iJPEG and JPEG

JPEG images are now still very popular and everywhere, so we also investigated the compatibility and difference between iPEG and the standard JPEG.

For a given quantizer, reconstructed images are compared and PSNRs are obtained to measure the error between the reconstructed images with iJPEG and JPEG.

Experiments show that PSNRs between the reconstructed images are all larger than 30dB. Thus, the error between them is very small.

As in Figure 4 with Lena, the difference measured by PSNR between iJPEG and JPEG should be acceptable. And the experiments with other test images are very similar, so the error with RDCT is very small.

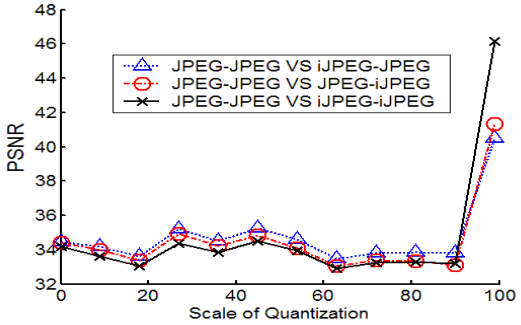

Figure 4 Comparison of reconstructed images with different codecs (Lena)

In our experiments, about 135 JPEG images are decoded by both iJPEG and JPEG. PSNRs obtained between those reconstructed image pairs are all large. Among all our experiments, the minimum PSNR is 35.54dB, the maximum is 44.63dB, and the mean PSNR is 37.99dB.

These imply that iJPEG and JPEG perform closely and the error of our iJPEG is as acceptable as that of JPEG.

4.3. Performance of Lossless Compression

Bit rates (bits per pixel, bpp) of some experiments are listed in Table 1, compared with JPEG, JPEG-LS [2] and JPEG 2000 lossless [1].

Table 2 gives the bit rates of color image experiments with different color transforms. ORCT and RGB only (without any color transform) are used in our iJPEG.

For lossless compression, although the bit rates with iJPEG are a little higher than that with JPEG-LS and JPEG-2000 lossless, most importantly, iJPEG provides both lossy and lossless compression possibilities and integrates them very well.

5. Conclusions

The integer reversible DCT (RDCT) is a very good approximation for reversible integer transform of DCT. The compression quality with RDCT is very close to that with DCT used in JPEG. The error between RDCT and DCT is small and our new compression system has good compatibility with JPEG.

Our RDCT enables lossless compression and thus integrates lossy and lossless compression in standard JPEG perfectly. Therefore, it is possible for our RDCT compression system to support lossy-to-lossless progressive coding. Compared with JPEG-LS or JPEG 2000 lossless, the performance of lossless compression is also good.

To compress color images losslessly, some reversible color transform is required. ORCT is employed in our compression system. Although ORCT is not always the best choice, such color transforms are still quite useful and may perform better for low bit rate compression.

6. References

[1] M. D. Adams, "The JPEG-2000 Still Image Compression Standard", *ISO/IEC JTC 1/SC 29/WG 1 N 2412*, Sept. 2001.

[2] R. Ansari and N. Memon. "The JPEG Lossless Compression Standards", *Handbook of Image and Video Processing*. A. Bovik, Editor, Academic Press, 2000.

[3] R. J. Clarke, "Relation between the Karhunen-Loeve and cosine transforms", *IEE Proceedings, Part F: Communications, Radar and Signal Processing*, Vol.128, No.5, pp.359-360, 1981

[4] M. J. Gormish, E. L. Schwartz, et al, "Lossless and nearly lossless compression of high-quality images", *Proceedings of the SPIE / IS&T Conference on Very High Resolution and Imaging* II, vol. 3025, San Jose, CA, pp. 62-70, February 1997.

[5] P. Hao, and Q. Y. Shi, "Matrix Factorizations for Reversible Integer Mapping" *IEEE Trans. on Signal Processing*, Vol. 49, No. 10, pp. 2314-2324, 2001.

[6] C. Christopoulos, A. Skodras, T. Ebrahimi, "The JPEG2000 still image coding system: an overview", *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, 2000, pp. 1103 -1127

[7] G. K. Wallace, "The JPEG Still Picture Compression Standard", *Communications of the ACM*, V34, N4, 30-44, 1991

Table 1. Lossless compression of gray-level images

Source Gray-Level Images			iJPEG lossless	JPEG best		JPEG-LS lossless	JPEG2000 lossless
Name	Size	bpp	bpp	bpp	PSNR	bpp	bpp
Lena	512*512	8	4.70	4.68	46.17	4.24	4.32
Barbara	512*512	8	5.11	5.08	44.51	4.73	4.67
Peppers	512*512	8	4.94	4.92	45.89	4.49	4.63
Mandrill	512*512	8	6.41	6.41	42.70	6.04	6.12
Goldhill	256*256	8	5.77	5.77	44.64	5.32	5.55
Fishingboat	256*256	8	5.24	5.22	45.05	4.80	4.89
Cameraman	256*256	8	5.13	5.08	43.73	4.31	4.57

Table 2 Lossless compression of color images

Source Color Images			IJPEG & ORCT lossless	iJPEG & RGB lossless	JPEG best		JPEG-LS	JPEG 2000 lossless
Name	Size	bpp	bpp	bpp	bpp	PSNR	bpp	bpp
Lena	512*512	24	14.30	14.78	12.34	40.11	13.60	13.60
Mandrill	512*512	24	18.67	19.71	16.73	38.62	18.51	18.09
Peppers	512*512	24	15.54	15.52	13.40	39.97	14.26	14.81
F16	512*512	24	12.93	14.02	10.88	39.99	11.84	11.55
Sail	512*768	24	12.96	17.45	10.91	39.69	15.61	10.66