The Multiply-Free Chen Transform - A Rational Approach to JPEG

James D. Allen, Steve M. Blonstein
Ricoh California Research Center
2882 Sand Hill Road, Suite 115
Menlo Park, CA 94025

Abstract

The JPEG committee of CCITT and ISO has recommended a method for the compression of continuous-tone still images. That method employs a two-dimensional Discrete Cosine Transform (DCT) followed by quantization and entropy coding. We have developed a high-performance low-cost alternative -- the Generalized Chen Transform (GCT) -- which complies with the JPEG recommendation.

1.0 Improving the Computational Ease

Our goal was to design a low-cost high-performance VLSI chip to perform image compression and decompression in a fashion compatible with the JPEG recommendation [JPEG 90]..

When the disadvantages of complicated computational procedures [Lee 84], [Wang 84], [Hou 86], [Suehiro 86], are considered, the Chen procedure [Chen 77], [Wang 83] is a good algorithm for performing the transforms required by JPEG compression. Furthermore the irrational multipliers of the Chen DCT can be adequately approximated with the simple integers of the GCT. In this fashion, JPEG compression of gray-scale images can be accomplished with 9.5 additions and a single serial multiplication per pixel.

The method also requires 3.5 constant shifts per pixel. These are essentially free in a VLSI circuit. Some of the shifts can be traded off for further additions on a general-purpose architecture where shifts are slow.

2.0 The Generalized Chen Transform for JPEG

2.1 The Chen Matrix for the Discrete Cosine Transform on Eight Points

The matrix of the Chen procedure and its inverse can be expressed in terms of internal multipliers as:

$$[GCT_8\ (a,b,c,r)] = [D(a,b,c)] \times [A(a,b,c,r)]$$

$$[GCT_8\ (a,b,c,r)]^{-1} = [A(a,b,c,R)]^T \times [D(a,b,c)]$$

where

$$2rR = 1\ ,$$

[D] is a trivial diagonal matrix required for normalization, and

$$[A] = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a & ar+r & ar-r & 1 & -1 & r-ar & -r-ar & -a \\ b & 1 & -1 & -b & -b & -1 & 1 & b \\ c & r-cr & -r-cr & -1 & 1 & cr+r & cr-r & -c \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -cr-r & cr-r & c & -c & r-cr & cr+r & -1 \\ 1 & -b & b & -1 & -1 & b & -b & 1 \\ 1 & r-ar & ar+r & -a & a & -r-ar & ar-r & -1 \end{vmatrix}$$

The matrix [A] admits an efficient decomposition into several sparse matrices; this is the essence of the Chen procedure [Chen 77] [Wang 83].

2.2 Two-Dimensional Image Transform

When the two-dimensional JPEG compression method is implemented with Chen's matrix [D] x [A], the forward transform can be expressed as a system of rank-64 matrices:

$$[JPEG] = [Q]^{-1} [D_v] [A_v] [D_h] [A_h]$$

and the inverse is:

$$[JPEG]^{-1} = [A_h]^T [D_h] [A_v]^T [D_v] [Q]$$

where:

$[A_v]$ denotes eight applications of [A] in the vertical direction

$[A_h]$ denotes eight applications of [A] in the horizontal direction

$[D_{v,h}]$ are the corresponding forms of [D]

[Q] is also a diagonal matrix and comprises the "psychovisual weights", which are tuned for the desired image quality, image type and output device.

Since the matrices $[A_v]$, $[D_h]$ are actually simple symmetric sparse forms, it can be easily derived that:

$$[A_v] [D_h] = [D_h] [A_v]$$

and this permits coalescing all the diagonal matrices:

$$[JPEG] = [Q'] [A_v] [A_h]$$

$$[JPEG]^{-1} = [A_h]^T [A_v]^T [Q'']$$

This simplification is not always possible with most of the "high performance" computational procedures for the DCT which have a structure more difficult than Chen's [Wu 89].

2.3 Rationalized Approximation ([GCT])

It has been noted [Allen 90], that the Chen procedure in the form given above operates with arbitrary parameters a, b, c, and r. In our implementation of JPEG we achieve adequate DCT closeness by using computationally simple parameters:

Chen Parameter	Standard Value(DCT)		Our Value(GCT)	
a	tan $7\pi/16$	≈ 5.0273	5/1	= 5.0000
b	tan $6\pi/16$	≈ 2.4142	12/5	= 2.4000
c	tan $5\pi/16$	≈ 1.4966	3/2	= 1.5000
r	$\sqrt{(0.5)}$	≈ 0.7071	128 /181	≈ 0.7072

We write simply [GCT] to denote this choice of parameters.

$$[GCT] = [GCT_8(5,\ 2.4,\ 1.5,\ 128/181)]$$

2.4 Computational Cost of GCT

Clearly each multiply by `a = 5' reduces to 1 shift and 1 add, and each multiply by `r = 128/181' can be rephrased as a multiply by:

$$181/128 = (1 + 1/32) * (1 + 1/4) + 1/8$$

which can be accomplished with 3 adds and 3 shifts.

The optimal arithmetic to support parameters `b = 12/5' and `c = 3/2' requires a look at matrices internal to the Chen procedure:

$$[Mb] = \begin{vmatrix} 12 & 5 \\ -5 & 12 \end{vmatrix} = \begin{vmatrix} 16 & 4 & 1 \\ -1 & 4 & 8 \end{vmatrix} \begin{vmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{vmatrix} \qquad [Mc] = \begin{vmatrix} 3 & 2 \\ -2 & 3 \end{vmatrix} = \begin{vmatrix} 4 & 1 & 0 \\ 0 & 2 & -1 \end{vmatrix} \begin{vmatrix} 1 & 0 \\ -1 & 2 \\ 0 & 1 \end{vmatrix}$$

In comparison with the Chen 8-point DCT, 2 multiplications by each of a,b,c,r are replaced respectively with 2 adds, 3 adds, 1 add, and 6 adds. For the 8x8 2D JPEG transform the total cost is 64 multiplications,224 shifts (free in hardwired circuitry), and 608 additions.

Compared with the 2D Chen DCT, 128 multiplications have been replaced with 192 additions and the shifts. The 64 multiplications comprise a simple diagonal-matrix multiply in the transform space. This structural simplicity is in contrast to other methods [Wu 89] which not only require more arithmetic, but have multiplications scattered hither-and-yon throughout an elaborate dataflow.

2.5 Mismatched Mode

Ideally, an image which is compressed with [GCT] will be decompressed with the inverse [IGCT]. In practice, sharing of files or transmissions between the equipment of different manufacturers will mean that an image compressed with [GCT] may be decompressed with the more costly [IDCT] decompressor; conversely, [DCT] images may be decompressed with our low-

cost chip. Error is introduced by this mismatch between the transforms assumed by compressor and decompressor.

Fortunately, the mismatch error associated with [IDCT][GCT] or [IGCT][DCT] is negligible at even the highest image qualities. With random pixels in the range 0 to 255, DCT/GCT mismatch will increase MSE by about 0.02, and this error will be less for real (correlated) images.

When the SNR's of JPEG images are compared with and without the mismatch, the difference is usually less than 100 microbels and even this sometimes randomly favors the mismatch!

3.0 Summary

We have developed an approach (GCT) for JPEG-compatible image compression which can be implemented in VLSI with fewer transistors than other methods. To be fully JPEG compliant, only one multiply per point is needed. No other multiplies are required.

References:

[Allen 90] Allen, J.D., "A new fast transform with energy compaction as good as the DCT", Ricoh Tech Report.

[Chen 77] Chen, W.H., Smith C.H. and Fralick, S.C., "A fast computational algorithm for the DCT", IEEE Trans. Commun. Vol COM-25, pp 1004-9 (1977)

[Hou 86] Hou, HS, "A fast recursive algorithm for computing the discrete cosine transform", SPIE Vol 697 Applications of Digital ImageProcessing (1986), pp 14-25.

[JPEG 90] Joint Photographic Experts Group of IEC/ISO and CCITT, "Draft (Revision 5.2) of the JPEG Algorithm", JPEG-8-R5.2

[Lee 84] Lee, BG, "A new algorithm to compute the DCT", IEEE ASSP-32 (1984), pp 1243-5.

[Pratt 72] Pratt, WK, "Generalized Wiener filtering computation technique", IEEE Trans. on Comput., vol C-21 (1972) pp 636-41.

[Suehiro 86] Suehiro and Hatori, "Fast ... DCT ...", Trans ASSP-34.

[Vitterli 84] Vitterli, M and Nussbaumer, HJ,"Simple FFT and DCT Algorithms", Signal Processing, Vol 6 No 4 pp 267 Aug 1984.

[Wang 83] Wang, Z., "Reconsideration of A Fast Computational Algorithm for the DCT", IEEE Trans. Commun. Vol COM-31, pp 121-3 (1983)

[Wang 84] Wang, Z., "Fast Algorithms for Discrete W and Fourier Transforms", ASSP-32, pp 803 Aug 84.

[Wang 85] Wang, Z., "On Computing the Discrete Fourier and Cosine Transforms", ASSP-33, pp 1341-1344.

[Wu 89] Wu, HR and Paolini, FJ, "A 2D Fast Cosine Transform", IEEE Conf. on Image Processing, vol I (1989)