MAPPED INVERSE DISCRETE WAVELET TRANSFORM FOR DATA COMPRESSION

Haitao Guo

Chromatic Research, Inc. Sunnyvale, CA 94089-1707

ABSTRACT

The Discrete Wavelet Transform (DWT) has been applied to data compression to decorrelate the data and concentrate the energy in a small portion of the coefficients. Compression can be achieved since most of the quantized wavelet coefficients are zeros. For the decoder, the traditional inverse discrete wavelet transform (IDWT) has a complexity proportional to the size of the data. In this paper, we propose a mapped inverse discrete wavelet transform algorithm (MIDWT) that takes advantage of the sparsity of the quantized wavelet coefficients, and significantly lowers the complexity of the IDWT to the level that is proportional to the number of non-zero coefficients. We further generalize the MIDWT to progressive decoding, and propose a realization of progressive IDWT without any run-time multiplication operations. Experiments show that our algorithms outperform the traditional IDWT for sparse coefficients, especially for progressive decompression.

1. INTRODUCTION

The discrete wavelet transform (DWT) is widely used for data compression [10, 9]. Wavelet based still image compression methods not only out-perform traditional methods in the rate-distortion sense, but also possess built-in scalability such that single bit stream can be transmitted progressively and decoded from coarse to fine resolution. Scalability is also very important for video compression, thus wavelet based method is incorporated in the proposed MPEG4 standard [3].

In many situations, the data are first compressed *once*, then the compressed bit streams are transmitted over the communication media like the Internet, or stored on the storage media like the digital video disk (DVD). Finally, copies of the same bit stream are decompressed *many times* at different places and different times. Over all, the inverse discrete wavelet transform (IDWT) used in the decoder is performed much more often than the DWT used in the encoder. Progressive decoding of embedded bit stream, where IDWT is used many times to reconstruct images at different resolutions [10, 9], serves as an example which could benefit greatly from a fast IDWT algorithm.

The wavelet transform decorrelates the data and concentrates the energy into a few coefficients. After quantization, a large number of the wavelet coefficients are zero. Consequently, after the inverse quantization, the inputs to the IDWT have lots of zero value coefficients as well. Although many compression schemes have been studies to take advantage of the sparsity in the DWT coefficients, the same sparsity has been ignored by IDWT. The goal of this paper is to propose and study fast IDWT algorithms that take advantage of the sparse inputs.

The discrete cosine transform (DCT) has also been extensively used for data compression. Similar to the DWT case, the quantized DCT coefficients also have plenty of zeros for most natural data. Several inverse discrete cosine transform (IDCT) algorithms have been proposed to take advantage of the sparsity to speed up the IDCT, e.g. the mapped IDCT (MIDCT) by McMillan and Westover [6], and the further improvement by Hung and Meng [4]. The basic idea is to treat inverse transform as a weighted sum of basis functions, and ignore those where the weight is zero. Applying similar idea of the MIDCT, we propose the mapped IDWT (MIDWT) which could significantly reduce the number of operations when decompressing most natural data. We further develop MIDWT techniques specific to wavelet transform. One of the situations that is unique to some of the embedded wavelet based compression algorithms is that it is possible to progressively decompress data from coarse to fine resolution. Straight forward realization of the progressive IDWT requires one IDWT per resolution. By generalizing our basic mapped IDWT, we develop the progressive mapped IDWT that does not require any multiplications.

The paper is organized as follows. We first briefly review the basic ideas of the mapped IDCT. The main result of this paper is developed in section 3, where we apply and generalize the idea of MIDCT to the wavelet transform. We further extent our algorithm to the progressive reconstruction from embedded bit stream, and develop a multiplication-free algorithm in section 4. Examples and performance analysis are shown in section 5.

2. REVIEW OF MAPPED IDCT

The DCT has many nice properties [8], and many compression standards are based on DCT [7, 3]. Figure 1 is a simplified diagram for DCT based compression and reconstruction. The signal X is first transformed into frequency domain by the DCT. The DCT coefficients D are further quantized and reconstructed to obtain coefficients C, whose components only takes on values in a discrete set. Error is usually introduced in the quantization step, i.e. C is an approximation of D. Thus the result of IDCT Y is an approximation of X.

For both JPEG and MPEG, type II [8] 2D 8×8 DCT is used. Since a 2D IDCT applied to an 8×8 block of image is equivalent to an 8 point 1D IDCT applied to rows, followed by an 8 point 1D IDCT applied to columns, here we concentrate on 1D algorithms. The basis functions for the 8 point DCT are

$$b_i(j) = \mu_i \cos\left(\frac{(2j+1)\pi i}{16}\right) \quad j = 0, 1, \dots, 7,$$
 (1)



Figure 1: Diagram for DCT based compression and reconstruction.

where $\mu_0 = 1/\sqrt{8}$, and $\mu_i = 1/2$ for i > 0. There are only 8 basis functions, so i = 0, 1, ..., 7. The reconstruction **Y** can be written as a weighted sum of these basis functions as

$$y(j) = \sum_{i=0}^{7} c_i b_i(j), \quad j = 0, 1, \dots, 7,$$
 (2)

or, in vector form

$$\mathbf{Y} = \sum_{i=0}^{7} c_i \mathbf{B}_i.$$
(3)

Traditional fast DCT and IDCT algorithms take advantage of the structure of the basis functions to reduce the numerical operations [8]. However, the mapped IDCT proposed in [6] takes a rather different, and somewhat straight forward approach. The key observation on equation (3) is that we do not need to weigh the basis function if the corresponding coefficient c_i is zero, and we do not need to cumulate that weighted basis function either. That is, equation (3) is equivalent to

$$\mathbf{Y} = \sum_{i:c_i \neq 0} c_i \mathbf{B}_i.$$
(4)

The 2D version of the mapped IDCT is proposed in [6], where there are $8 \times 8 = 64$ coefficients and corresponding basis 2D functions. There are two main reasons that make the MIDCT a practical algorithm. First, only a few out of the 64 coefficients are non-zero. Second, even if the coefficients are not zero, we do not need 64 multiplications to weigh the basis functions since many values in the 64 point basis function are duplicated, thus we only need one multiplication for each *unique* value in the basis function.

Symmetry property of the basis function is also observed and taken advantage of in [4] to further speed up the IDCT. It has been found that the symmetric mapped IDCT is faster than traditional IDCT when the number of non-zero coefficients are small. The exact cross-over point depends on the computer architecture.

3. MAPPED IDWT

The DWT is another very powerful tool for signal analysis and synthesis, representation and compression. The general wavelet theory is very rich, and warrants extensive treatment [2, 12, 1]. In this paper, we only look at the wavelet theory from the discrete basis function point of review.

For a given DWT, we have a pair of filters h (lowpass), and g (high pass). The scaling function S and the wavelet function W at different scale can be obtained as

$$\mathbf{S}_{j+1} = up(\mathbf{S}_j) * \mathbf{h}, \quad \mathbf{W}_{j+1} = up(\mathbf{W}_j) * \mathbf{h}, \tag{5}$$

with

$$\mathbf{S}_1 = \mathbf{h}, \quad \mathbf{W}_1 = \mathbf{g}, \tag{6}$$

where up() denotes up-sampling, and * denotes convolution. Similar to the IDCT case in section 2, we can write the IDWT as a weighted sum of basis functions as

$$\mathbf{Y} = \sum_{i=0}^{N} c_i \mathbf{B}_i,\tag{7}$$

where *N* is the data length, \mathbf{B}_i denotes the wavelet basis function \mathbf{W}_j on various scales, and the scaling basis function \mathbf{S}_j on the coarsest scale. Unlike the DCT case where the basis functions are on the same scale and have the same support, the basis functions for the DWT have increasing support at coarser scales. Also, the basis functions on the same scale are merely shifts of one another. By grouping the coefficients according to scales and relabeling them, we can rewrite equation (7) as

$$\mathbf{Y} = \sum_{j=1}^{J} \sum_{i=0}^{\frac{N}{2^{j}}-1} c_{j,i} \mathbf{W}_{j,i} + \sum_{i=0}^{\frac{N}{2^{j}}-1} d_{J,i} \mathbf{S}_{J,i},$$
(8)

where j is used for scale, and i is used for position within the scale, J is the total number of scales of decomposition. Although equation (8) has more detail information and is more widely used, equation (7) is a more convenient representation for our purpose. Clearly, the idea of MIDCT leads us from equation (7) to the following key equation for the mapped IDWT,

$$\mathbf{Y} = \sum_{i:c_i \neq 0} c_i \mathbf{B}_i. \tag{9}$$

If the number of non-zero coefficients are small, equation (9) yields an efficient algorithm obviously. Also, if the basis functions have duplicated values, we can further reduce the number of operations by only performing multiplication with the unique values. This allows us to cut the number of multiplications by half for those symmetrical wavelets often used for image compression.

The number of operations per coefficient is proportional to the length of the corresponding basis function. For DCT, the length is fixed, 8 for 1D, and 64 for 2D. But for DWT, the length of the basis functions gets longer as the scales get coarser. Let len(C) denote the length of the vector C. Equations (5) and (6) imply,

$$len(\mathbf{S}_{j+1}) = 2len(\mathbf{S}_j) + len(\mathbf{h}) - 2, \tag{10}$$

$$len(\mathbf{W}_{j+1}) = 2len(\mathbf{W}_j) + len(\mathbf{g}) - 2, \tag{11}$$

with

$$len(\mathbf{S}_1) = len(\mathbf{h}), \quad len(\mathbf{W}_1) = len(\mathbf{g}).$$
 (12)

So the length of the basis function approximately doubles at each increasing scale. Thus the worst case complexity of the MIDWT, when all the coefficients are non-zero, is O(JN).

4. PROGRESSIVE MAPPED IDWT

For embedded compression algorithms [10, 9], the wavelet coefficients are coded and transmitted from the most significant bitplane to the least significant bitplane, so we can reconstruct data from coarse to fine resolution as more and more parts of the compressed bit stream are available. Let s_i be the sign of c_i , and $p_{i,k}$ be the bit value (either 0 or 1) of c_i on the k_{th} bitplane, we can write c_i as

$$c_{i,K} = s_i \sum_{k=0}^{K-1} p_{i,k} \frac{q}{2^k},$$
(13)

where q is the reconstruction level for the most significant bitplane, and K is the total number of bitplanes. The reconstructed signal using the first K bitplanes is

$$\mathbf{Y}_{K} = \sum_{i=0}^{N-1} c_{i,K} \mathbf{B}_{i} = \sum_{i=0}^{N-1} s_{i} \sum_{k=0}^{K-1} p_{i,k} \frac{q}{2^{k}} \mathbf{B}_{i}.$$
 (14)

Let \mathbf{Z}_K be the refinement for going from bitplane K to K + 1, i.e.

$$\mathbf{Z}_{K} = \mathbf{Y}_{K+1} - \mathbf{Y}_{K} = \sum_{i=0}^{N-1} s_{i} \ p_{i,K} \frac{q}{2^{K}} \mathbf{B}_{i}.$$
 (15)

Again, only non-zero $p_{i,K}$ need to be kept, so

$$\mathbf{Z}_{K} = \sum_{i:p_{i,K}\neq 0} s_{i} \ p_{i,K} \frac{q}{2^{K}} \mathbf{B}_{i}.$$
 (16)

Since the value of q is always transmitted first, we can per-calculate the basis functions at each bitplane as

$$\mathbf{B}_{i,K} = \frac{q}{2^K} \mathbf{B}_i. \tag{17}$$

They only need to be calculated once per scale and per bitplane. Since the basis function at lower bitplane is half of that at the next higher bitplane, it can be calculated from the basis function at the higher bitplane very efficiently by right shifting for the fix point implementation, or by reducing the exponents for floating point implementation. Also, since $p_{i,K}$ is either 0 or 1, we can simplify equation (16) as

$$\mathbf{Z}_{K} = \sum_{i:p_{i,K}=1} s_{i} \mathbf{B}_{i,K}$$
(18)

$$= \sum_{i:s_i=l, p_{i,K}=l} \mathbf{B}_{i,K} - \sum_{i:s_i=-l, p_{i,K}=l} \mathbf{B}_{i,K}.$$
 (19)

Thus we avoid any multiplications for the run-time progressive MIDWT. Note that $p_{i,K}$ is the bit value of the coefficient c_i at bitplane K, even if c_i has non-zero bit at bitplane higher than K, $p_{i,K}$ could still be zero, thus leads to operation reduction.

5. EXAMPLE

We compare the complexity of the traditional IDWT, the MIDWT and the progressive MIDWT for the two signals shown in figure (2). Figure (3a) shows that the percentage of non-zero coefficients is small and increases as the number of bitplanes increases. Our MIDWT outperforms the traditional IDWT when the number of bitplanes is small, as shown in figure (3b). Since the numbers of multiplications and additions are similar, we only show one of them in the plots. For progressive decompression, the percentage of non-zero bits for each bitplane is shown in figure (4a). The percentages are also small but are not necessarily monotonously increasing. Our progressive MIDWT does not require any multiplication, and the numbers of additions needed to progressively decompress data based on all the previous bitplanes are shown in figure (4b). Much smaller number of operations is achieved over the traditional IDWT, which requires similar number of multiplications as additions. Hence, our progressive MIDWT is highly efficient.



Figure 2: Test signals.

6. SUMMARY AND DISCUSSION

In this paper, we propose a mapped realization of the inverse discrete wavelet transform that takes advantage of the large number of zero wavelet coefficients often generated by quantization. The complexity of our algorithm is proportional to the number of nonzero coefficients, and experiments show that it significantly reduces the number of operations when reconstructing from sparse coefficients. Thus our algorithm is especially suitable for decompression of natural signals. We also generalize our algorithm to progressive decompression so that unnecessary computations corresponding to zero-value coefficients are avoided at every resolution and no multiplication operation is required at run-time. Examples confirm the efficiency of our algorithms. Our algorithm easily generalizes to separable 2D DWT in the row-column fashion. It is also possible to formulate direct 2D mapping as done for DCT [6].

The usual IDWT has the tree structure [5], and equation (9) corresponds to flattening the tree. The basic idea for MIDCT and MIDWT can be directly applied to the tree structure, since the convolution is nothing but a weighted summation. Thus, we can guarantee the complexity to be less than that of the standard implementation. Furthermore, we can combine our mapped idea with other efficient IDWT implementation, such as the lattice structure [12] and the ladder/lifting structure [11] to further reduce the complexity.



Figure 3: (a) The percentage of non-zero coefficients increases as the number of bitplanes increases. (b) Comparison of the number of operations for traditional IDWT and the MIDWT for the test signals.

7. REFERENCES

- [1] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and the Wavelet Transform*. Prentice-Hall, 1997.
- [2] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992. Notes from the 1990 CBMS-NSF Conference on Wavelets and Applications at Lowell, MA.
- [3] International Organization for Standardization. Coding of moving pictures and associated audio information. *MPEG97/N1796*, 1997.
- [4] A. C. Hung and T. H.-Y. Meng. Statistical inverse discrete cosine transforms for image compression. In *Proceedings of* SPIE Symposium of Electronic Imaging 2187, 1994.
- [5] S. Mallat. A theory for multiscale signal decomposition: The wavelet representation. *IEEE Trans. on Pattern and Machine Intelligence*, 11(7):674–693, 1989.
- [6] L. McMillan and L. Westover. A forward-mapping realization of the inverse discrete cosine transform. In *Date Compression Conference*, 1992.



Figure 4: (a) The percentage of non-zero bits at the corresponding bitplane. (b) Comparison of the number of operations for traditional IDWT and the progressive MIDWT (PMIDWT) for the test signals.

- [7] W. B. Pennebaker and J. L. Mitchell. JPEG Still Image Data Compression Standard. Van Nostrand Reinhold, New York, 1993.
- [8] K. R. Rao and P. Yip. Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press, Boston, 1991.
- [9] A. Said and W. A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circ. and Syst. Video Tech.*, 1996.
- [10] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. SP*, 41(12):3445–3462, December 1993.
- [11] Wim Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic analysis*, 3(2), 1996.
- [12] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1992.