논문 98-01-08

Image Browse for JPEG Decoder

Ui-Pil Chong*
(鄭 義 必*)

Abstract

Due to expected wide spread use of DCT based image/video coding standard, it is advantageous to process data directly in the DCT domain rather than decoding the source back to the spatial domain. The block processing algorithm provides a parallel processing method since multiple input data are processed in the block filter structure. Hence a fast implementation of the algorithm is well suited. In this paper, we propose the JPEG browse by Block Transform Domain Filtering(BTDF) using subband filter banks. Instead of decompressing the entire image to retrieve at full resolution from compressed format, a user can select the level of expansion required($2^N \times 2^N$). Also this approach reduces the computer cpu time by reducing the number of multiplication through BTDF in the filter banks.

요 약

DCT 변환을 기반으로 하는 비디오 코딩은 많은 사용자와 더불어 급속한 기술 발전을 하게 되었다. 공간영역내에서 디코딩을 수행하는 것보다 DCT 영역에서 직접 데이터를 처리하는 것이 계산속도 면에서 빠르다. 그리고 블록처리 알고리듬은 병렬처리에 기초하므로 데이터 처리속도가 빠른 하드웨어로 구성되어질 수 있다. 본 논문에서는 서브밴드의 필터뱅크에서 블록변환영역 필터링을 이용한 JPEG브라우저를 제안한다. 디코딩시에 압축된 파일로부터 전체 영상을 디코딩하는대신 사용자가 원하는 크기의 영상을 브라우징 할 수 있다. 한편 DCT 영상 데이터가 입력으로 사용될 경우 제안된 블록변환 필터링은 일반적인 서브밴드 필터링보다 필터뱅크내에서의 곱셈 수를 줄임으로서 계산속도면에서 빠른 결과를 얻을 수 있다.

I. Introduction[1]

Direct transform domain processing removes the necessary of inverse transform and reduces the amount of computation comparing to time domain processing. Block processing algorithm provides a parallel processing method since multiple input data are processed in the block filter structure. Block Transform Domain Filtering(BTDF)[1][4][5] provides an approach suitable for processing of JPEG coded image and other block base coded video. This paper focuses on JPEG browsing using BTDF in subband filter banks. The purpose of BTDF is to process data directly in the DCT domain. Since the number of nonzero elements after quantization is significantly smaller than in spatial domain, we can save on the number of multiplications as well as computer cpu time. The DCT coefficients of input data for filter banks are retrieved through proper inverse quantization at JPEG decoder[2]. We briefly review the BTDF in section 2. In section 3, we analyze the computational complexity of subband filter banks using the BTDF. In section 4, the BTDF based approach to

* 蔚山大學校 컴퓨터情報通信工學部
(School of Computer Eng. and Information Technology, Ulsan Univ.)
接受日: 1998年4月8日, 修正完了日: 1998年7月21日

the subband decomposition and reconstruction are introduced. Computer simulation results are discussed and conclusions are given in section 5.

II. Block Transform Domain Filtering(BTDF)

The BTDF can utilize any block transform data for equivalent spatial domain processing. In the JPEG browser application, DCT coefficients are input to the filter banks and outputs are the spatial domain data. Since the number of nonzero DCT coefficients are small, the amount of computation is significantly smaller than the equivalent spatial domain filtering. For the completeness of the presentation, we briefly introduce BTDF. The transfer function of an N-th order FIR filter is represented as

$$H(z) = b_0 + b_1 z^{-1} + \cdots + b_N z^{-N} = B(z) \triangleq \frac{Y(z)}{X(z)} \quad (1)$$

The Z-domain relationship of the input and output signals is $Y(z) = B(z)X(z)$ or the time domain representation is $y(n) = b(n) * x(n)$, where $b(n) = b_n$, $n = 0,1, \cdots ,N$ are real coefficients of $B(z)$. The block matrix equation[1] is given by

$$\underline{y}_k = \boldsymbol{B}_0\underline{x}_k + \boldsymbol{B}_1\underline{x}_{k-1} + \cdots + \boldsymbol{B}_M\underline{x}_{k-M}, \quad (2)$$

where $\underline{x}_k$, $\underline{y}_k$ and $\boldsymbol{B}_i$ are the block input, the block output and the block filter coefficients respectively. Also, M is the block filter order. Applying the Z-transform to the above equation with respect to the block index k, a Z-domain relationship between input and output vectors is obtained as

$$\underline{Y}(z) = (\boldsymbol{B}_0 + \boldsymbol{B}_1 z^{-1} + \cdots + \boldsymbol{B}_M z^{-M})\underline{X}(z) \quad (3)$$

The block transfer function in the sampled z-domain is given by

$$\boldsymbol{H}(z) = \boldsymbol{B}_0 + \boldsymbol{B}_1 z^{-1} + \cdots + \boldsymbol{B}_M z^{-M} \quad (4)$$

The block FIR filter equation (4) can be represented using the relation $\underline{X} = \mathbf{T}\underline{x}$ or $\underline{x} = \mathbf{T}^{-1}\underline{X}$, where $\mathbf{T}$ is the forward DCT transform kernel. Direct substitution gives the block transform domain transfer function, $\underline{y}_k = \dot{\mathbf{H}}(z)\underline{X}(z)$ where $\dot{\mathbf{H}}(z) = \boldsymbol{H}(z)\mathbf{T}^{-1}$. The matrix coefficients of the BTDF are given as

$$\dot{\mathbf{B}}_m = \mathbf{B}_m\mathbf{T}^{-1}, \ m = 0,1, \cdots ,M \quad (5)$$

Finally, we get the desired BTDF as follow.

$$\underline{y}_k = \dot{\mathbf{B}}_0\underline{X}_k + \dot{\mathbf{B}}_1\underline{X}_{k-1} + \cdots + \dot{\mathbf{B}}_M\underline{X}_{k-M} \quad (6)$$

The direct form structure in the block transform domain is shown in Fig. 1.

Equation (6) means that the DCT input data, $\underline{X}$ are directly processed in the BTDF without the inverse DCT of input data.

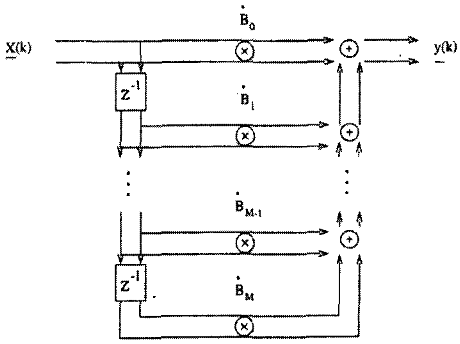

Fig. 1. FIR direct form of BTDF

III. Computational Complexity of Wavelet Transform

The 1-D BTDF developed in section 2 can be directly applied to 2-D image blocks for wavelet transforms. A single 2-D forward wavelet transform of an image is accomplished by two separate horizontal and vertical 1-D transform. The input 2-DCT image $V_0(x,y)$ as shown Fig. 2 from the JPEG/MPEG decoder is first blocked by a blocking mechanism and then filtered along the x dimension, resulting in a low pass image and a high pass image. Both images are then filtered along the y dimension, resulting in four subimages. The way to count the number of multiplications for BTDF in this section is as follows; Let us consider a p × p input DCT image and an N-tap separable FIR filter with block size L. The number of multiplications for column and row filtering in one band are given by

$$T_s = L * \sum_{q=1}^{p} [(\sum_{k=0}^{nc} (n_{q,k,0} + n_{q,k-1,1} + \cdots + n_{q,0,M}) + \sum_{l=1}^{nb} (n_{q,l,0} + n_{q,l-1,1} + \cdots + n_{q,0,M}))] \quad (7)$$

where,

L : block size

M : Block filter order

p : p × p DCT input image size

n_c, n_b : the number of blocks per column and row

$n_{q,k,i}, n_{q,l,i}$: the number of nonzero coefficients of k-th block image in q-th column and l-th blocked image in q-th row.

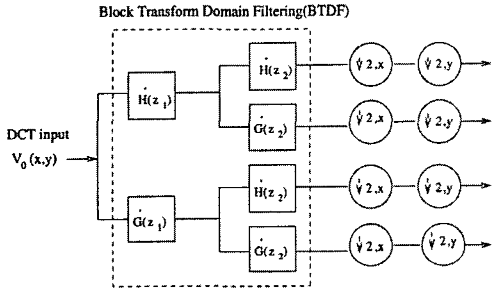

Fig. 2. 2-D forward wavelet transform using BTDF

IV. Image Browse using Subband Decomposition

In this section, we explain how to get the browsing image from the DCT compressed format. We can get the DCT coded source between the Dequantizer and the IDCT from JPEG decoder for the DCT input data of filter bank as shown Fig. 3.

After BTDF and unblocking signals, we can downsample each of the filtered images in the x dimension by 2 and also y dimension by 2 without loss of information, resulting in four subimages. Note that all filtering process except the BTDF of first decomposition stage is same as the conventional 2-D block filtering process in subband filter banks. Also, the filters in the filter banks are satisfied by a paraunitary perfect reconstruction conditions. We can see that the decomposition provides subimages corresponding to different resolution levels. We get the browser image ($2^N \times 2^N$) at synthesis part whenever we need it as shown Fig. 3. The user can select up to the entire image at full resolution. Note that since the filter banks guarantee the perfect reconstruction, the full resolution image is same as the output of JPEG decoder.

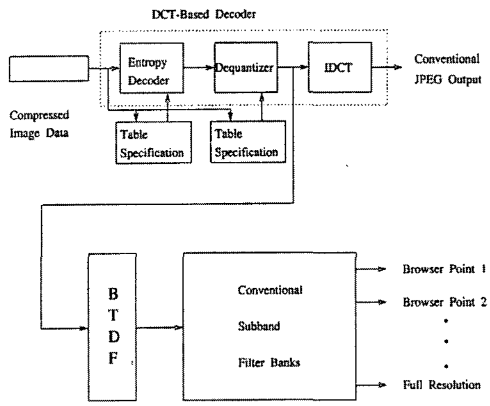

Fig. 3. Structure of JPEG Browse

V. Simulation Results and Conclusions

The experiments are presented to test the proposed algorithm. The filters used to compute the wavelet coefficients are Daubechies 4-tap filters[3]. Test images, 256 x 256 "lena" and "airplane" are used. In this example, the block size L is equal to 8, the block order M is equal to 2, and the number of block for column n_c and row n_b are equal to 32. Table 1 shows the relative computational complexity for BTDF based on the number of multiplications of the spatial domain block filtering. The number of multiplications in the BTDF are less than those of spatial domain block filtering at same quantization scale, Q-i(i=1, 2, 3) which means the degree of quantization in the JPEG coder for the original image. For example, at Q-1 quantization scale, 0.19 bpp for lena image case, the columns of the blocked DCT image are filtered with $\dot{H}(x)$ in Fig. 2. The number of multiplications per column filtering are 35200. Then the rows of the blocked intermediate signals are filtered with $\dot{H}(y)$. The number of multiplications per row filtering are 233792. So, the total number of multiplications for one band by equation (7) are 268992. Similarly, repeating it for other bands, we get 1.075M multiplications in Table 1. for overall bands. The number of multiplications in the spatial domain block filters are similarly done including the inverse transform. Since we have quantized DCT data for storage or transmission purposes, we can save multiplications through the BTDF. Also, we measured the computer cpu execution time comparing BTDF to spatial domain block filtering on SUN 4 unix systems. The results are shown in Table 2. Especially, we could save about 10 seconds(25%) for Q - 1 scale quantizer. As shown in Fig. 4, we can get the different browse size at the end of synthesis part. We can see a difference in the blocking artifacts from various sizes of browse images because as the browser get smaller, we can barely see the blocks. Also the proposed algorithm can be applied to wireless communication through the progressive transmission scheme of wavelet transcoding of JPEG or MPEG.

Table 1. Block time domain filtering and BTDF for quantized image(M : mega)

	spatial domain block filtering			BTDF (relative complexity %)		
	Q-1	Q-2	Q-3	Q-1	Q-2	Q-3
lena	8.491M	8.684M	9.100M	1.075M (12.7)	2.836M (32.7)	5.946M (65.3)
air plane	8.521M	8.759M	9.111M	1.121M (13.2)	2.850M (32.5)	5.876M (64.5)

Table 2. Computer cpu execution time(seconds)

test image	spatial domain block filtering			BTDF		
	Q-1	Q-2	Q-3	Q-1	Q-2	Q-3
lena	39.35	39.98	41.17	29.65	33.28	39.80
air plane	39.51	40.22	41.42	29.80	33.47	39.55

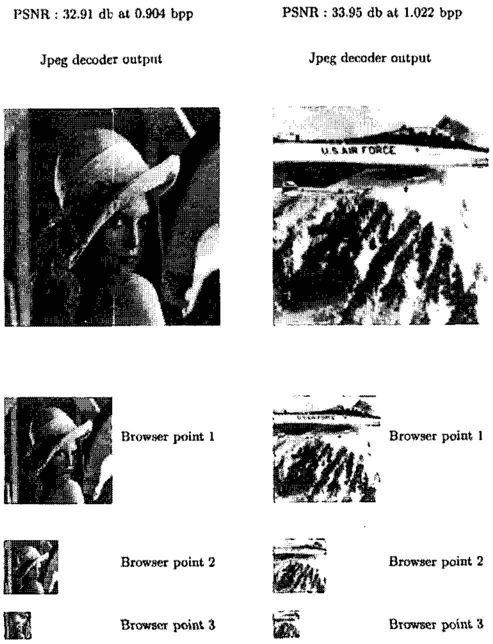

Fig. 4. Examples of JPEG Browse

References

[1] Y.Jang and S.P. Kim, " Block Digital filter Structures and Their Precision Responses, " *IEEE Trans. Circuits and Systems*, vol. 43, no. 7, pp. 495-506, Jul., 1996.

[2] G. K. Wallace, "The JPEG still-picture compressing standard," *Comm. of ACM*, pp. 30-44, Apr., 1991.

[3] Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets," *Comm. in Pure and Applied*

Math., vol. 41, no. 7, pp. 909-996, 1998.

[4] Uipil Chong and S. P. Kim, "Wavelet Transcoding of Block DCT-Based Images Through Block Transform Domain Processing," *SPIE*, vol. 2825, Aug., 1996.

[5] Uipil Chong, "Finite Precision Implementation of Subband Filter Banks and ransform Domain Processing," *Dissertation at Polytechnic Univ.*, 1997.

저 자 소 개

鄭 義 必 (正會員)

1978년 울산대학교 전기공학과 공학사. 고려대학교 대학원 전기공학과 공학석사. Oregon State Univ. 전기및컴퓨터공학과 공학박사. 현 울산대학교 컴퓨터정보통신공학부 조교수