FAST IMAGE DECOMPRESSION FOR TELEBROWSING OF IMAGES

Shaou-Gang Miaou and Julius T. Tou

Department of Electrical Engineering
University of Florida, Gainesville, FL 32611

Abstract. Progressive image transmission (PIT) is often used to reduce the transmission time of an image telebrowsing system. A side effect of the PIT is the increase of computational complexity at the viewer's site. This effect is more serious in transform domain techniques than in other techniques. Recent attempts to reduce the side effect are futile as they create another side effect, namely, the discontinuous and unpleasant image build-up. Based on a practical assumption that image blocks to be inverse transformed are generally sparse, this paper presents a method to minimize both side effects simultaneously.

1. Introduction

One important evaluation criterion for a telebrowsing system is the response time which is the time elapsed from the moment a retrieval request is issued until the desired information is actually displayed on the monitor [1]. The response time can roughly be divided into three major parts. The first part is the searching time for the system to locate the desired information. The second part is the transmission time to send the information through a channel. The third part is the display time for the information to be displayed on the monitor. The early studies of the telebrowsing systems were concentrated on the efficient retrieval of pure text information [2,3]. In this case, the searching time is the only major concern. However, for modern telebrowsing systems where multimedia information, including text, audio, image, and video, is considered, the transmission time and the display time become a significant part of the response time because of huge amount of data involved in still images and video (a sequence of images).

To reduce the transmission time of an image telebrowsing system, a well known scheme called progressive image transmission (PIT) is often used. PIT allows an approximate reconstruction of an image whose fidelity is built up gradually until the viewer decides either to abort the transmission sequence or to allow further reconstruction. This scheme increases the effective compression ratio because usually only a small part of the compressed data needs to be sent for browsing purpose.

With PIT techniques, the transmission time can be greatly reduced. However, it also creates a side effect, that is, it increases the processing time at the viewer's site because an inverse PIT process is required. Since the major task of the inverse PIT process is the image decompression given part of the compressed data, the research is aimed at the development of fast image decompression schemes for the inverse PIT process.

The rest of the paper is organized as follows. First, the PIT schemes and their computational complexities are briefly addressed. Then, the drawbacks of recent attempts to reduce the computational complexities are discussed. Next, the demonstration of a new approach is given. Finally, a performance comparison between the new approach and the recent ones is made.

2. PIT Schemes and Their Computational Complexities

There are many PIT schemes. According to Tzou's classification, they are divided into three major categories, namely, spatial domain, transform domain, and pyramid-structured, based on where the progression takes place [4]. Each category can be further divided into several classes of techniques. The classification is shown in Figure 1. Note that not all of the PIT schemes will produce a considerable amount of computational overhead in the inverse PIT process. For instance, the spatial domain schemes only require a very low computational effort in the inverse PIT process. In pyramid-structured PIT schemes, only successively filtered pyramid techniques require high computational complexity in the inverse PIT process. Even for the successively filtered pyramid techniques, however, the complexity to process the first few levels of a pyramid from the top remains low. From a practical point of view, the processing of the first few levels of the pyramid may suffice the purpose of image browsing. On the other hand, transform domain techniques usually take considerable amount of computation in the inverse PIT process, since the inverse transforms have to be carried out with about the same computational effort for every stage of image reconstruction.

- Spatial Domain Techniques
    - Bit-plane method
    - Tree-searched vector quantization
    - Progressively quantized DPCM
- Transform Domain Techniques
    - Scanning pattern techniques
    - Transform domain multistage quantization
    - Bit-slicing method
- Pyramid-structured progressive transmission
    - Tree-structured pyramid
        - Binary tree
        - Quadtree
    - Successively filtered pyramid

Figure 1. Tzou's classification of PIT schemes

In transform domain PIT schemes, the transform coefficients are first quantized and then divided into segments. Only one segment of quantized coefficients is sent for one stage of image reconstruction. The only differences among all transform domain techniques are the ways to determine the segments and the order in which they are sent. One common feature among them is that the transform coefficients are only "partially" encoded where in a non-PIT or sequential scheme they are said to be "fully" encoded.

For a transform domain non-PIT scheme, one M x N inverse transform for an M x N image block is needed. However, for a transform domain PIT scheme, r times of M x N inverse transform are needed for the image block, where r is the number of stages of image reconstruction. The lower bound for r is 1 but its upper bound depends on the image, the viewer, and the PIT scheme. Therefore, the computation load for inverse transform is r times heavier in PIT schemes than in non-PIT schemes.

One transform domain scheme using discrete cosine transform (DCT) receives great attention, since the DCT has the energy packing capabilities and also approaches the statistically optimal transform (i.e. Karhunen-Loeve transform) in decorrelating a signal governed by a Markov process [5]. In addition, it is part of the recently approved JPEG standard [6,7]. The JPEG standard has brought a tremendous impact on the image-coded related industry. However, as far as implementation of the standard is concerned, the standard provides only a guideline. How to implement the standard efficiently for certain application still relies on the ingenuity of designers. For example, JPEG has chosen to specify neither a unique forward DCT (FDCT) algorithm or a unique inverse DCT (IDCT) in its recommendation. This is because research in fast DCT algorithms is ongoing and no single algorithm is optimal for all implementations [7]. For the application of inverse PIT, we will show that traditional fast two dimensional (2-D) IDCT algorithms can be accelerated to reduce the processing time at the viewer's site.

3. Previous Approaches and Their Drawbacks

To relieve the computation burden of IDCTs in inverse PIT, the following approaches have been used.

Approach 1: Use traditional fast algorithms for IDCT. The computational complexity is reduced from $O(N^4)$ by the definition of IDCT to $O(N^2\log_2N)$ by traditional fast algorithms, where N x N is the block size. There are many fast algorithms available for IDCT. For an 8 x 8 IDCT, one of the best algorithms reported so far takes 96 multiplications and 466 additions [8].

Approach 2: Use a fast progressive reconstruction method. It is a combination of a special scheme and the use of approach 1. This approach was first proposed by Takikawa to perform fast progressive reconstruction for discrete Fourier transformed and Walsh-Hadamard transformed images [9]. Later, Miran and Rao followed the similar derivation by Takikawa and developed a fast progressive reconstruction for DCT images [10]. The basic idea of approach 2 is to decompose the N x N transformed block into $\log_2N + 1$ sparse matrices, each of which can be inverse transformed by 1 x 1, 2 x 2, 4 x 4,..., and N x N fast inverse transform algorithms.

Approach 2 has some advantages over approach 1. First, the computational complexity is lower. For example, consider a 4-stage image reconstruction and an 8 x 8 image block. Approach 1 takes four 8 x 8 IDCTs while approach 2 requires only one 1 x 1 IDCT, one 2 x 2 IDCT, one 4 x 4 IDCT, and one 8 x 8 IDCT. The computational saving is obvious. Secondly, the delay time is reduced. The delay time is the time to wait for all the elements in a transformed block before an inverse fast transform can be performed.

However, approach 2 has a serious problem, that is, it has a poor and discontinuous image build up. The reason is that the order in which the sparse matrices are formed and sent is not in the order of visual significance. In general, a DCT coefficient with higher variance (or energy) tends to be more visually significant than that with lower variance. It is well known that the DCT coefficient variances are highly correlated along the zig-zag scan [11]. Approach 2 has a fixed transmission pattern that does not even close to the zig-zag scan. This problem has been confirmed experimentally by Miran and Rao [10]. They ascribed the drawback to not having low frequency terms immediately adjacent to DC components in the intermediate stages of reconstruction. Another drawback of approach 2 is that it still requires all elements of the sparse matrices to start computing the inverse transform. Thus, the delay time is reduced but not eliminated.

One more drawback for both approaches 1 and 2 is the computational redundancy of traditional fast algorithms in inverse PIT. If IDCT is used in image decompression, its input block contains only a few nonzero coefficients. In addition, if a PIT scheme is used, the input matrix to IDCT contains even fewer nonzero elements. To visualize the redundancy, consider the signal flowgraph of a fast IDCT algorithm. Since a zero presented at an input node contributes nothing to the output, the paths between a zero input node and output nodes are trivial or redundant.

To get a picture on how many spatial frequencies retained on the average after the quantization, many 512 x 512 8-bit greyscale images and RGB components of color images were tested. In the test, JPEG's coding scheme, including a recommended quantization table, was used. Part of the test result is presented in column 3 of Table 1. It is shown, even for a very busy image such as baboon image, no more than a quarter of quantized coefficients are nonzero. Even with so many spatial frequencies set to zero, the decompressed images and their originals are perceptually indistinguishable. Next, consider the case when a small visible image degradation is allowed. To produce a small image degradation, the same test was repeated except that the round-off operation in JPEG's scheme was replaced by truncation. The new numbers are shown in column 4. The decompressed images have only minor degradation, for it does not diminish our capability to recognize meaningful objects in the images. In fact, the image quality is good enough to be the last stage of PIT. The test shows that the number of nonzero quantized DCT coefficients decreases sharply at the minor expense of image quality. In the inverse PIT process, the average number of nonzero elements in an 8 x 8 matrix does not need to be higher than that in column 4.

Table 1. Average # of Nonzero Quantized DCT Coefficients in an 8 x 8 Block

Images	Image Activity	Round-off	Truncation
Lena	Low	6.13	4.01
Boat	Medium	9.20	6.00
Baboon	High	15.50	9.80

How much of the matrix must be zero for it to be considered sparse depends on the applications. Generally, a matrix is called sparse if there is an advantage in exploiting its zeros [12]. It is well known that exploiting the sparsity can lead to enormous computational savings in many applications such as solving simultaneous equations with Gaussian elimination method. Inspired by this fact, it is curious to see if the sparsity of the input matrix can also be exploited to compute IDCT efficiently in the environment of inverse PIT. Since the characteristic of an input image block to IDCT is generally not considered in traditional algorithms, a nonconventional approach must be adopted to exploit the sparsity of the input matrix. The proposed approach will be presented in the following manner. First, we describe the goal to be accomplished by the approach. Then, the rationale of the approach is discussed. Next, based on the rationale, two methods are presented -- one is too slow to be useful, the other is its fast version. The fast version is shown to be good enough for the practical use.

4. The Proposed Approach

In the inverse PIT process, computation burden of IDCT and computation redundancy associated with traditional algorithms are two major problems. The inherent drawbacks in Takikawa's or Miran and Rao's approach present another problem in the inverse PIT process. In view of all these problems, our approach should meet the following goals. First, it must be fast and efficient. Second, it must allow a scanning pattern that can conform to the visual significance. Finally, it must have practically no delay time.

For the ease of discussion, several terms are defined first. A target matrix is an image block consisting of the quantized DCT coefficients that are partially encoded for PIT. Performing an IDCT on a target matrix results in a matrix called goal matrix. The result of processing one nonzero element in the target matrix is called the partial contribution to the goal matrix. Throughout this paper, the partial contribution is treated as a matrix or all its elements depending on the context.

Based on the definition of 2-D IDCT, only nonzero elements in the target matrix can contribute to the goal matrix. In fact, the value of each nonzero element can affect the values of all elements in the goal matrix. The idea of our approach is to completely ignore the zero elements in the target matrix and process each nonzero element separately and efficiently. The goal matrix is then updated periodically by adding the partial contribution. Therefore, the computation of IDCT is divided into two tasks, i.e., the computation of partial contribution and the update of the goal matrix. The idea adapts particularly well to the scheme where DCT coefficients are run-length coded (such as JPEG's).

The definition of 2-D IDCT is

$$g_{xy} = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u)c(v)F_{uv} \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{1}$$

where x=0, ..., M-1, y=0, ..., N-1, and

$$c(k) = \frac{1}{\sqrt{2}} \quad if\ k=0,$$
$$=1 \quad otherwise.$$

The coefficient in front of the double summation of equation 1 is only a scale factor which requires essentially no computation (except a register shift operation) in practical applications where M=N and M=4, 8, or 16 are often used. Thus, it is usually neglected when comparing the computational complexity among the fast algorithms of IDCT. By taking the scale factor out, equation 1 becomes

$$f_{xy} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u)c(v)F_{uv} \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{2}$$

where

$$f_{xy} = \frac{\sqrt{MN}}{2} g_{xy}$$

If $[f_{xy}]_{uv}$ is defined as the partial contribution to the goal matrix $[f_{xy}]$ due to F_{uv} alone, then

$$[f_{xy}]_{uv} = c(u)c(v)F_{uv}\cos\left(\frac{(2x+1)u\pi}{2M}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{3}$$

The partial contribution can be obtained by the use of definition in equation 3 directly. Assume the values of cosine functions for different combinations of x and u are precalculated and stored as a table. The table can also be used as the values of cosine functions for different combinations of y and v with x and u replaced by y and v, respectively. Let Q be the number of multiplications required to find the partial contribution due to F_{uv}. Then, Q=2MN if both u and v are not zero, Q=3MN if u=0 or v=0 but not both, and Q=0 if u and v are both zero. For an M x N target matrix with n (>1) nonzero F_{uv}, where n << MN, the number of multiplications to get the goal matrix is from 2(n-1)MN to 3nMN. With this naive approach, no addition but 128 to 384 multiplications are required if M=N=8 and n=2. This is not good enough, since an 8 x 8 fast IDCT can take as low as 96 multiplications [8]. Therefore, a better way to compute the partial contribution is needed.

Equations 2 and 3 are equivalent if only one term in the double summation of equation 2 is nonzero. So the traditional fast algorithms for equation 2 can be applied to equation 3 as well. However, the direct use of them to compute the partial contribution is not desirable since they contain high computational redundancy. We found that with a systematic reduction rule for the signal flowgraphs of traditional fast algorithms, a much faster way to compute the partial contribution than the naive approach is possible. The rule is based on the two attributes associated with the partial contribution, which we call the mirror effect and the reducible property.

From equation 3, it can be readily shown that

$$\begin{aligned} [f_{x'y}]_{uv} &= (-1)^{u}[f_{M-1-x',y}]_{uv} \\ [f_{xy'}]_{uv} &= (-1)^{v}[f_{x,N-1-y'}]_{uv} \\ [f_{x'y'}]_{uv} &= (-1)^{u+v}[f_{M-1-x',N-1-y'}]_{uv} \end{aligned} \tag{4}$$

where x' and y' are particular values of x and y, respectively. Equation 4 indicates that the partial contribution exhibits high degree of symmetry or mirror effect. Note that only possible sign changes are involved in equation 4 and practically no addition or multiplication is required. The significance of this result is that only a quarter of the partial contribution needs to be determined through additions and multiplications. The rest of them can be determined by simple copy operations and possible sign changes.

The reducible property can be stated as follows. The M x N partial contribution due to F_{uv} is equivalent to that of (M/m) x (N/n) partial contribution due to $F_{u/m,v/n}$, where cd(M,u) = m, cd(N,v) = n, and cd(a,b) is a common divisor between non-negative integers a and b. This statement can be proved easily by noting that

$${}_{M,N}[f_{xy}]_{u,v} = {}_{M/m,N/n}[f_{xy}]_{u/m,v/n} \tag{5}$$

where ${}_{M,N}[f_{xy}]_{u,v}$ is the $[f_{xy}]_{uv}$ defined in equation 3. If cd(M,u)=1 only, it is said to be an irreducible partial contribution in row. Similarly, if cd(N,v)=1 only, it is said to be irreducible in column. If for some m>1 or n>1, the partial contribution is said to be reducible. Note that the reducible property is separable, i.e, the reduction in row size and column size can be processed separately. The largest reachable reduction for F_{uv} happens when gcd(M,u)=m and gcd(N,v)=n, where m>1 and n>1, and gcd(a,b) is the greatest common divisor between non-negative integers a and b. For example, if M=N=8 ,u=6 ,and v=4, the 8 x 8 partial contribution due to F_{64} is equivalent to 4 x 2 partial contribution due to F_{31}, since cd(8,6) can be 2 and cd(8,4) can be 4. It is also a maximum reducible case for F_{64} since gcd(8,6)=2 and gcd(8,4)=4. Note that gcd(a,0) = a. Therefore, if u and v are both zero, M x N partial contribution due to F_{uv} is simply a 1 x 1 partial contribution due to F_{00}, which is always $F_{00}/2$ no matter what the values of M and N are.

Combining the mirror effect and the reducible property can lead to a great saving in computation of partial contribution. Consider the following example: We want to compute the 8 x 8 partial contribution due to F_{44}. Since gcd(8,4)=4, it can be reduced to the 2 x 2 partial contribution due to F_{11}. By using the mirror effect, only 1 x 1 of the 2 x 2 partial contribution needs to be determined explicitly, which is $f_{00}=F_{44}/2$. It can then be expanded to 2 x 2 partial contribution by the use of mirror effect:

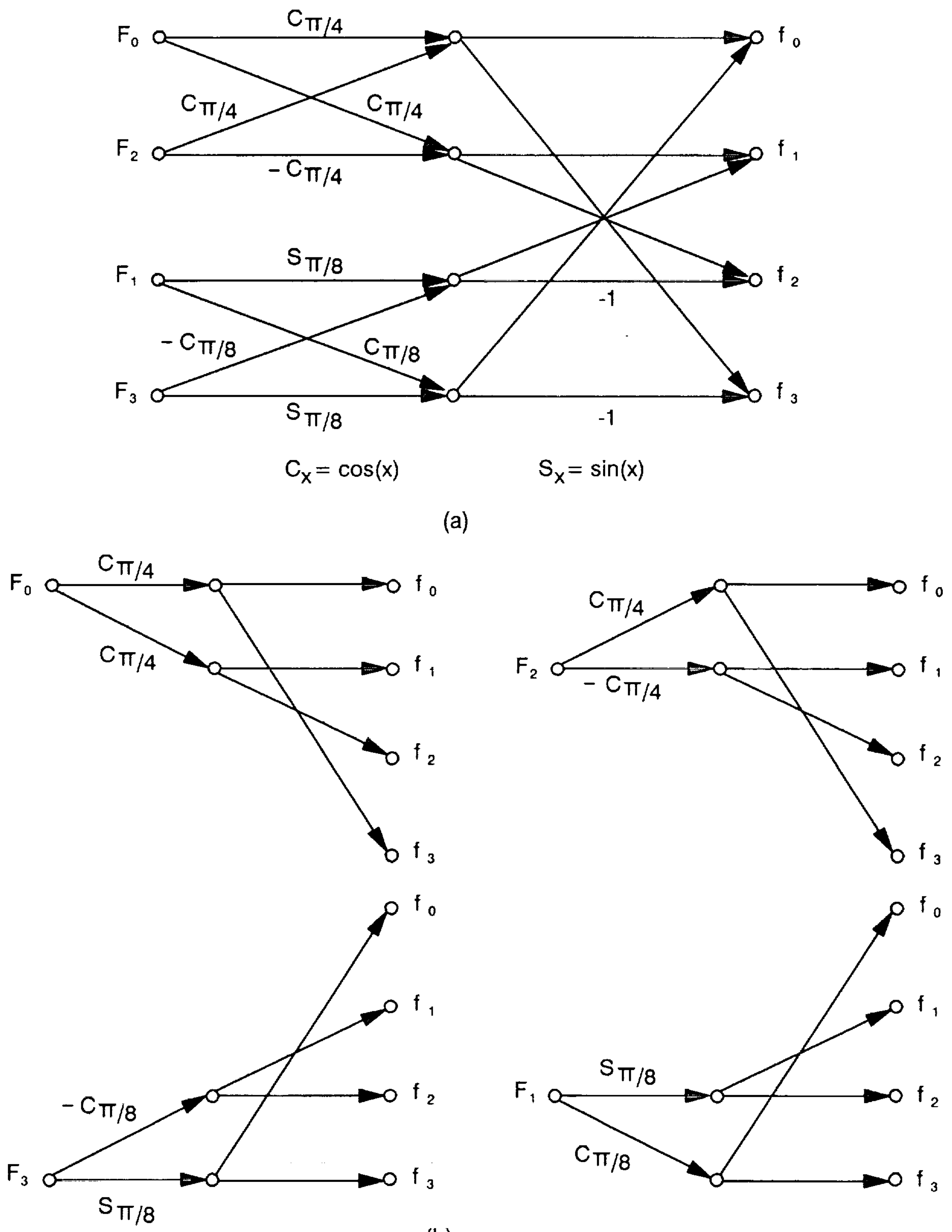
$$\begin{matrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{matrix}$$

where $f_{01}=-f_{00}$, $f_{10}=-f_{00}$, and $f_{11}=f_{00}$. Similarly, by using the mirror effect for F_{11}, we expand the partial contribution to 4 x 4 :

$$\begin{matrix} f_{00} & f_{01} & f_{02} & f_{03} \\ f_{10} & f_{11} & f_{12} & f_{13} \\ f_{20} & f_{21} & f_{22} & f_{23} \\ f_{30} & f_{31} & f_{32} & f_{33} \end{matrix}$$

where $f_{02}=-f_{01}$, $f_{03}=-f_{00}$, $f_{12}=-f_{11}$, $f_{13}=-f_{10}$, $f_{20}=-f_{10}$, $f_{21}=-f_{11}$, $f_{30}=-f_{00}$, $f_{31}=-f_{01}$, $f_{22}=f_{11}$, $f_{23}=f_{10}$, $f_{32}=f_{01}$, $f_{33}=f_{00}$. Since cd(8,4) can be 2, the 8 x 8 partial contribution due to F_{44} is equivalent to the 4 x 4 partial contribution due to F_{22}. By using the mirror effect for F_{22}, we can expand the partial contribution of 4 x 4 to the desired result of 8 x 8. Note that no multiplication is required to determine the 64 elements of partial contribution due to F_{44}.

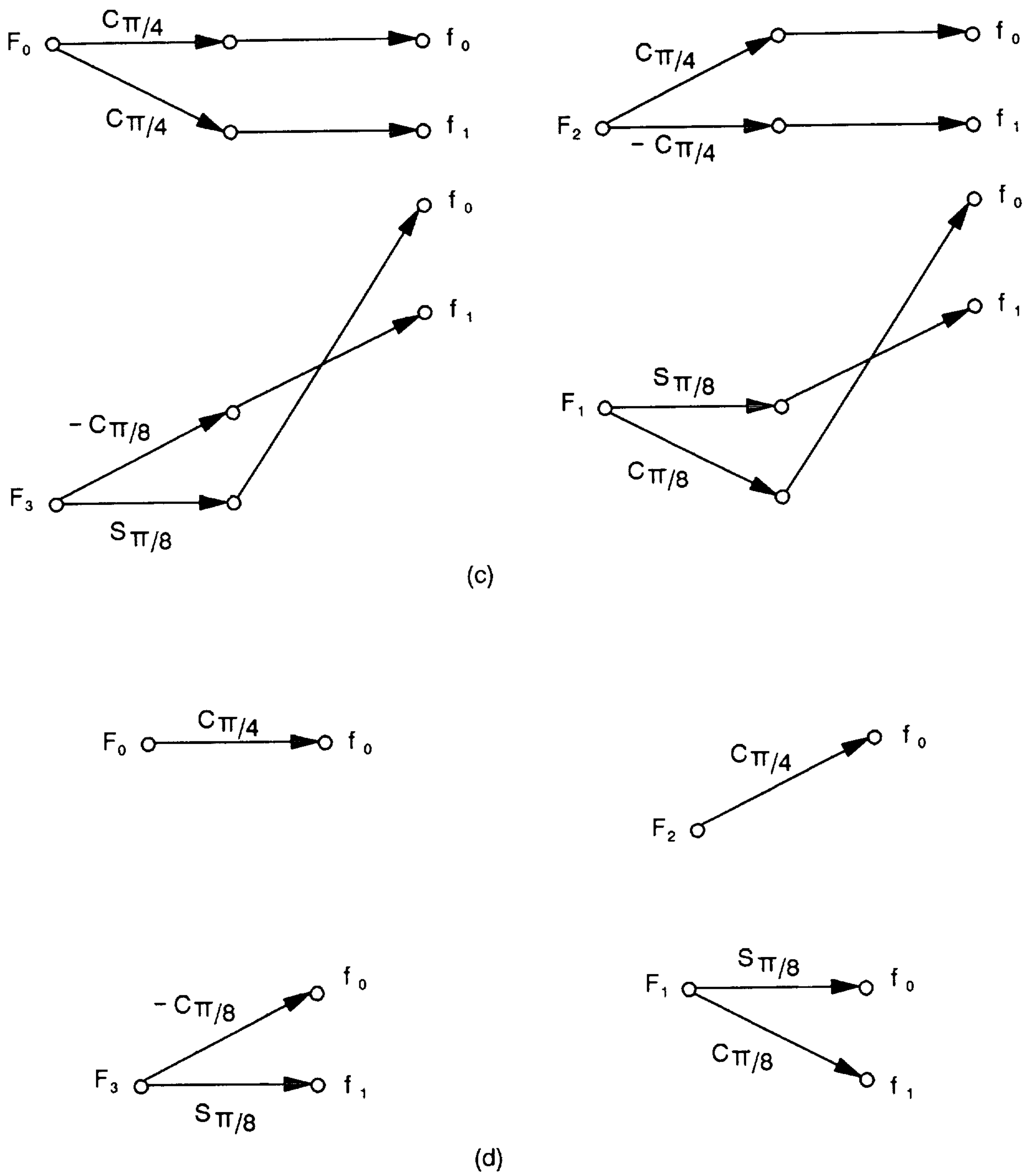
The basic principle to reduce the signal flowgraph of a traditional algorithm is by retaining only the nontrivial paths. This concept is demonstrated by an example. Consider the row-column or indirect approach of a fast 2-D IDCT for a 4 x 4 target matrix. Chen's algorithm is chosen here because it is simple and well recognized [13]. Normally, 8 4-point 1-D IDCTs are needed to accomplish the task (with very complicated data reordering, 4 4-point IDCTs are enough [8]). However, in our case at most 3 4-point IDCTs are necessary (1 along the rows (or columns) of the target matrix to get an intermediate matrix and 2 along the columns (or rows) of the intermediate matrix to get a 2 x 2 submatrix of the partial contribution). The other three 2 x 2 submatrices can be derived automatically by the use of mirror effect. Furthermore, each 4-point IDCT can be done efficiently since only one input data out of 4 is nonzero. Consider the signal flowgraph for a 4-point IDCT shown in Figure 2(a). The outputs of the 4-point IDCT (denoted by f_0, f_1, f_2, and f_3) can be treated as linear combinations of the 4 inputs (denoted by F_0, F_1, F_2, and F_3). Since only one of the inputs is nonzero, Figures 2(a) and 2(b) are functionally equivalent. The signal flowgraph in Figure 2(b) can be further simplified by retaining only two of the four outputs (f_0 and f_1) as shown in Figure 2(c) because the other two outputs can be derived by the use of mirror effect. Since the reducible property is separable, it can be used here to further reduce some of the subgraphs in Figure 2(c). Specifically, the subgraphs with input F_0 and F_2 are reducible. The



(a)

(b)

Figure 2. (a) A 4-point IDCT (b) signal decomposition of (a)
(c) simplified version of (b) (d) irreducible subgraphs of (a)



(c)

(d)

Figure 2. (Continued)

final irreducible subgraphs are shown in Figure 2(d). For convenience, the subgraphs shown in Figure 2(d) are said to be in their primitive forms. In other words, they can not be reduced or simplified any more. The above procedure can be extended easily to 8-point or higher order cases.

The primitive subgraph with input F_u and the one with input F_v can be cascaded as a signal flowgraph to compute part of the partial contribution due to F_{uv}. The connection rule is: at each output of the first subgraph, the second subgraph is cascaded. Which subgraph should be the first is immaterial as far as the result is concerned. However, the computational complexity may be different.

The complexity of computing part of the partial contribution due to F_{uv} can be examined by checking the primitive subgraphs with input F_u and F_v. The two primitive subgraphs are cascaded in the way described earlier. If the first subgraph takes P multiplications and the other requires Q multiplications, then the total number of multiplications required to obtain part of the partial contribution would be P+PQ multiplications. Alternatively, P and Q are also the number of output nodes for one subgraph and another, respectively. So P and Q can be obtained by counting the number of output nodes of the irreducible subgraphs. Since $P+PQ = P(1+Q)$, a fast way to tell the required number of multiplications is to take the product of P and Q+1. P+PQ multiplications will also be the complexity to compute the full size partial contribution since no addition operations are involved and the expansion of partial contribution to its full size adds no complexity. Suppose the two subgraphs are cascaded in reverse order, the complexity becomes Q+QP. But $Q+QP \neq P+PQ$ if $P \neq Q$. Thus, the order of the subgraphs is relevant to the complexity. If $P<Q$, P+PQ is always smaller than Q+QP. Therefore, the order selection should be such that the first one requires less complexity than the second one. The numbers of multiplications required for different combinations of u and v are shown in Table 2 and Table 3 for 4 x 4 and 8 x 8, respectively. Note that for u=0 or M/2 and v=0 or N/2, no multiplication is required (except a left shift operation by one bit).

Table 2. The Number of Multiplications Associated with Fuv (4 x 4)

u\v	0	1	2	3
0	0	3	0	3
1	3	6	3	6
2	0	3	0	3
3	3	6	3	6

Table 3. The Number of Multiplications Associated with Fuv (8 x 8)

u\v	0	1	2	3	4	5	6	7
0	0	5	3	5	0	5	3	5
1	5	20	10	20	5	20	10	20
2	3	10	6	10	3	10	6	10
3	5	20	10	20	5	20	10	20
4	0	5	3	5	0	5	3	5
5	5	20	10	20	5	20	10	20
6	3	10	6	10	3	10	6	10
7	5	20	10	20	5	20	10	20

According to Table 3, we can estimate the average number of multiplication required for a nonzero element in the 8 x 8 case. Assume that the chance of a nonzero element falling in any u-v pair is equally likely. Then the average will be the 1/64 of the sum of all the numbers shown in Table 3. The result is 9.5 multiplications per nonzero element. This is about 7 to 20 times faster than the naive approach mentioned earlier. Similarly, from Table 2, we will get 3 multiplications per nonzero element for the 4 x 4 case.

The update of a goal matrix is straightforward because it involves only additions of the corresponding elements in each partial contribution. The total number of additions is (n-1)MN for the update of the goal matrix, where n is the number of nonzero elements in an M x N target matrix.

5. Performance Comparison of the Approaches

The advantages of the proposed approach are as follows:

(1) It has essentially no delay time and computational redundancy.
(2) It allows any scanning or transmission patterns, including the zig-zag scanning pattern. Note that the zig-zag scanning pattern is generally good for many images. However, a better or optimal scanning pattern for a particular image may deviate from the zig-zag scanning pattern [14]. Furthermore, it can be different from one image to another. Therefore, it is critical to be adaptive to different scanning patterns.
(3) In inverse PIT, it has lower computational complexity than traditional fast algorithms.

Both advantages 1 and 2 are due to the separate processing of the element in the input matrix. The performance comparison of approach 1, approach 2, and the proposed approach are summarized in Table 4.

Table 4. Performance Comparison of the Approaches

	Approach 1	Approach 2	Proposed
Scanning Pattern	Adaptive	Fixed and Poor	Adaptive
Delay Time	High	Low	Lowest
Computational Redundancy	High	Low	Lowest
# of Multiplication*	384	114	95
# of Additions*	1864	740	576

*The fast algorithm in [8] is used for approaches 1 and 2. The number of nonzero elements in an 8 x 8 target matrix is assumed to be 10 on the average. In addition, 4 stages of image reconstruction are assumed.

6. Conclusion

This paper presents a new and promising solution to the problem of heavy computation of IDCTs in inverse PIT process. When approach 2 was shown to have poor and discontinuous image build up in 1990 [10], the research in fast progressive reconstruction for transform domain PIT schemes seems to be hopeless. With the proposed approach, both the fast progressive reconstruction and the pleasant image build up can be achieved simultaneously. This is an encouraging result for the research of transform domain fast progressive reconstruction.

References

[1] N. Dal Degan, P. Migliorati, S. Pozzi, V. Trecordi, "IMAGINE: Effective Retrieval from a Remote Image Database, " Proceedings of IEEE Global Telecommunications Conference, Vol. 1, pp. 306-311, 1990.

[2] Julius T. Tou, Robert W. Depree, Paul B. Lin, "Telebrowsing of Science Information Via a Minicomputer, " Proceedings of NSF Seminar on Scientific and Technical Information Dissemination, Washing, D.C., 1976.

[3] Julius T. Tou, Robert W. Depree, "Telebrowsing System and Its Applications, " Proceedings of the European Computing Congress, 1978.

[4] Kou-Hu Tzou, "Progressive Image Transmission: A Review and Comparison of Techniques, " Optical Engineering, Vol. 26, No. 7, pp. 581-589, July 1987.

[5] K. R. Rao, P. Yip, Discrete Cosine transform -- Algorithms, Advantages, and Applications, Academic Press, Inc., 1990.

[6] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, June 1992.

[7] Gregory K. Wallace, "The JPEG Picture Compression Standard, "IEEE Transactions on Consumer Electronics, Vol. 38, No. 1, pp. xviii-xxxiv, Feb. 1992.

[8] Nam Ik Cho, Sang Uk Lee, "Fast Algorithm and Implementation of 2-D Discrete Cosine Transform, " IEEE Transactions on Circuits and Systems, Vol. 38, No. 3, pp. 297-305, March 1991.

[9] Kei Takikawa, "Fast Progressive Reconstruction of a Transformed Image, " IEEE Transactions on Information Theory, Vol. 30, No. 1, pp. 111-117, January 1984.

[10] M. Miran, K. R. Rao, "Fast Progressive Reconstruction of Images Using the DCT, " in Signal Processing V: Theories and Applications, pp. 897-900, edited by L. Torres, E. Masgrau, and M. A. Lagunas, Elsevier Science Publishers B. V., 1990.

[11] A. G. Tescher, R. V. Cox, "An Adaptive Transform Coding Algorithm, " International Conference on Communications, pp. 47-20 through pp. 47-23, Philadelphia, PA, July 14-16, 1976.

[12] I. S. Duff, A. M. Erisman, J. K. Reid, Direct Methods for Sparse Matrices, Clarendon Press, Oxford, 1986.

[13] Wen-Hsiung Cheng, C. Harrison Smith, S. C. Fralic, "A Fast Computational Algorithm for the Discrete Cosine Transform, " IEEE Transactions on Communications, Vol. 25, No. 9, pp. 1004-1009, September 1977.

[14] Bowonkoon Chitprasert, K. R. Rao, "Human Visual Weighted Progressive Image Transmission, " IEEE Transactions on Communications, Vol. 38, No. 7, pp. 1040-1044, July 1990.