Fast Downscaled Inverses for Images Compressed with M-Channel Lapped Transforms

Ricardo L. de Queiroz, *Member, IEEE,* and Reiner Eschbach

***Abstract*—Compressed images may be decompressed and displayed or printed using different devices at different resolutions. Full decompression and rescaling in space domain is a very expensive method. We studied downscaled inverses where the image is decompressed partially, and a reduced inverse transform is used to recover the image. In this fashion, fewer transform coefficients are used and the synthesis process is simplified. We studied the design of fast inverses, for a given forward transform. General solutions are presented for M-channel finite impulse response (FIR) filterbanks, of which block and lapped transforms are a subset. Designs of faster inverses are presented for popular block and lapped transforms.**

I. Introduction

IMAGE processing has become very popular since images were brought into the desktop computer. The resolution of digital cameras, scanners, and printers often outpaces the improvements in data storage devices in such a way that image compression is fundamental for most applications. The image is compressed, stored, and printed or displayed on one of several devices, each supporting a distinct resolution. In this fashion, one must be able to decompress the image and resize it to fit the desired resolution. It is reasonable to assume that the image will be stored at a higher resolution and downsized to fit the lower resolution devices. It is also desirable to avoid resizing the image after it is decompressed to its full resolution. One of the reasons for this is because it will require a large buffer to temporarily store the full resolution image. Another reason is the fact that it may be a very slow process, as more pixels than necessary are processed. The alternative is to decompress the image directly into its lower resolution.

A practical example is given in Fig. 1, which shows a system for processing documents in a personal computer. The images are scanned at, let us say, 300 spots per inch (spi) and to save memory space they are immediately compressed (by hardware or software). Such images may be later printed in a 300 spi color printer or immediately previewed on a 75 spi monitor. In this previewing example, the image may be decompressed at one quarter of its natural resolution in each direction. Another application arises in the use of transform coded images in web browsers, where the client application may scale the incoming image to fit the viewing resolution and size.

Manuscript received October 24, 1995; revised November 19, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ping Wah Wong.

The authors are with Xerox Corporation, Webster, NY 14580 USA (e-mails: queiroz@wrc.xerox.com; Reiner_Eschbach@wb.xerox.com).

Publisher Item Identifier S 1057-7149(97)03907-9.

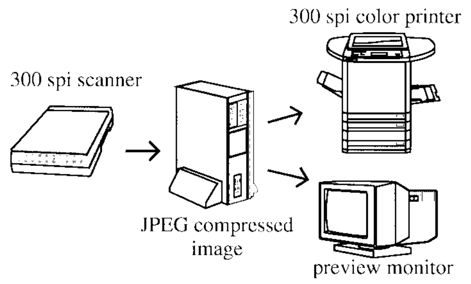


Fig. 1. Images may be acquired and compressed for display or printing at several resolution on distinct devices. A good example is a system for scanning documents in a personal computer. The images are scanned and printed at high resolutions while they are previewed at a lower resolution.

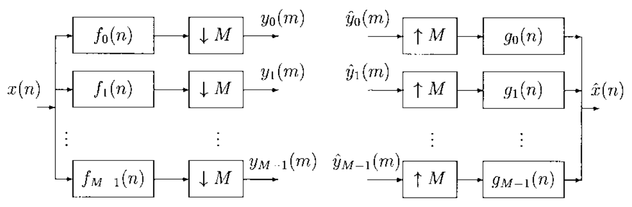


Fig. 2. M-channel uniform filterbank.

Transform coding [1], [2] is very popular for image compression. Depending on the transform used, the resizing task can be greatly simplified. For example, downsampling by factors that are powers of two is trivial if we use the discrete wavelet transform (DWT) or other similar subband approaches [3]. However, we focus our attention on other attractive transforms called *parallel* or *M-channel* [4]. In these, the input image is directly decomposed into several subbands at once. Examples of these transforms are block transforms such as the discrete cosine transform (DCT) [5], discrete sine transform (DST) [5], etc. Along with block transforms we can also cite the lapped transforms [6] such as lapped orthogonal transform (LOT) [6], [7], the families of generalized LOT (GenLOT) [8], [9] and extended lapped transforms (ELT) [6], [10], as well as other M-channel filterbanks [4]. The most popular image compression scheme is the JPEG baseline system, which is a *de facto* standard and is based on the DCT [11].

We will study fast inverse systems to reconstruct a downsampled approximation of a signal. It is assumed that such a signal is available in the transform domain. Therefore, the method is only applicable when the inverse transform and decoding process is not hardwired to a particular method, since we redesign the synthesis section in order to obtain a

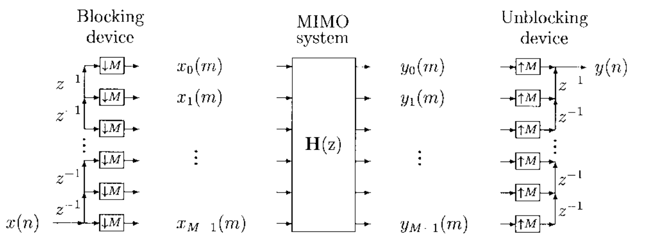


Fig. 3. Polyphase transfer matrix to implement a linear periodically time-varying system.

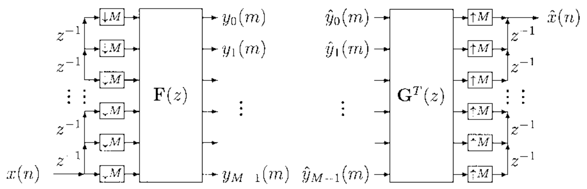


Fig. 4. Analysis and synthesis systems as polyphase transfer matrices.

fast reconstruction.

In Section II, filterbanks and polyphase transfer matrices are briefly presented. Common *post-processing-type* resampling operations are discussed in Section III. The concept of block resampling as a practical simplification of the problem and solutions for downscaled inverses are presented in Section IV. Section V discusses fast inverses and presents design examples of fast inverses for several block and lapped transforms. Some reconstructed image approximations are shown as examples. Particularly, the JPEG baseline system is discussed in detail because it is a common application. Finally, Section VI contains the conclusions of this work.

In terms of notation, our conventions are as follows. $\mathbf{I}_n$ is the $n \times n$ identity matrix. $\mathbf{0}_{n\times m}$ stands for the $n \times m$ null matrix. $[\ ]^T$ signifies transposition. In general, capital boldface letters are reserved for matrices, so that $\mathbf{a}$ represents a vector while $\mathbf{A}$ represents a matrix. Unless otherwise stated, only column vectors are used. A sequence $x(n)$ can be expressed in its polyphase components $x_i(m)(0 \le i \le M-1)$ as $x_i(m) = x(Mm+i)$. The z-transforms of $x(n)$ and of $x_i(n)$ are given by $X(z)$ and $X_i(z)$, respectively. The M polynomials $X_i(z)$ are grouped in $M \times 1$ vectors denoted by $\mathbf{x}(z)$. In a similar fashion, the polyphase components of a sequence can be further decomposed into M polyphase components yielding the notation $x_{ij}(n)(0 \le j \le M-1)$ which has z-transform $X_{ij}(z)$. The $M \times M$ polynomials $X_{ij}(z)$ are grouped in an $M \times M$ matrix denoted by $\mathbf{X}(z)$. The rows of $\mathbf{X}(z)$ are the polyphases of $X_i(z)$ and may also be grouped into vector $\mathbf{x}_k^T(z)$.

II. Filterbanks and Polyphase Transfer Matrices

We assume the use of a finite impulse response (FIR) uniform filterbank as shown in Fig. 2. Block and lapped transforms are special cases of this class of filterbank. There are M analysis filters $f_k(n)$ and M synthesis filters $g_k(n)(0 \le k \le M-1)$. The signal $x(n)$ is decomposed into M subband signals $y_k(m)(0 \le k \le M-1)$. After processing or quantization, the subband signals $\hat{y}_k(m)$ are used to construct the signal $\hat{x}(n)$. In accordance with our notation one can regard $f_k(n), g_k(n)$, and $y_k(n)$ as polyphase components of virtual sequences $f(n), g(n)$, and $y(n)$, respectively. For simplicity, let $f_k(n)$ and $g_k(n)$ have L taps each, padding zeros if necessary, and let $L = NM$, for integer N.

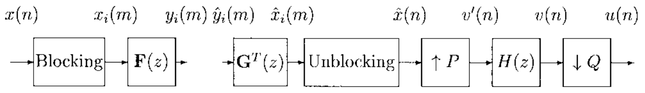


Fig. 5. Analysis and synthesis systems with resampling as postprocessing.

It is sometimes more convenient to work with polyphase structured systems. A multi-input multi-output (MIMO) system is regarded here as a linear system of FIR filters relating M polyphase components of the input signal to M polyphase components of the output signal [4]. Such a system is described by a polyphase transfer matrix (PTM) [4]. Let $\mathbf{H}(z)$ denote a PTM relating the polyphase components of $x(n)$ and $y(n)$ as in Fig. 3. The conversion from sequence $x(n)$ to its polyphase components is also called a blocking operation[1] (see Fig. 3). The signal is actually partitioned into blocks of M samples each. It is also a conversion from a serial to a parallel stream of numbers. The inverse operation (from polyphases to the original signal) is also called the unblocking operation.

Then

$$\mathbf{H}(z) = \begin{bmatrix} H_{00}(z) & H_{01}(z) & \cdots & H_{0,M-1}(z) \\ H_{10}(z) & H_{11}(z) & \cdots & H_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1,0}(z) & H_{M-1,1}(z) & \cdots & H_{M-1,M-1}(z) \end{bmatrix}. \tag{1}$$

$H_{ij}(z)$ represents a linear time-invariant (LTI) FIR system relating the ith input polyphase to the jth output polyphase. According to our notation, $H_{ij}(z)$ are the z-transforms of the polyphase components of $h_i(n)$, which itself is one polyphase component of the sequence $h(n)$. Such a system is said to be linear periodically time-varying (LPTV) [4]. To see this, let $h(n)$ be the impulse response of an ℓ-tap LTI FIR filter, $\ell = kM$, and let $Y(z) = H(z)X(z)$. Then

$$y(n) = h(n) * x(n) = \sum_{n=0}^{\ell-1} h(n)\ x(m-n)$$
$$= \sum_{i=0}^{k-1}\sum_{j=0}^{M-1} h(iM+j)\ x(m-iM-j).$$

Thus

$$y_q(p) = \sum_{i=0}^{k-1}\sum_{j=0}^{M-1} h_j(i)x_{q-j}(p-i) = \sum_{j=0}^{M-1} h_j(p) * x_{q-j}(p)$$

and

$$Y_q(z) = \sum_{j=0}^{M-1} H_j(z)X_{q-j}(z). \tag{2}$$

[1]The delay chain to create polyphase components is the opposite of the one used in [4].

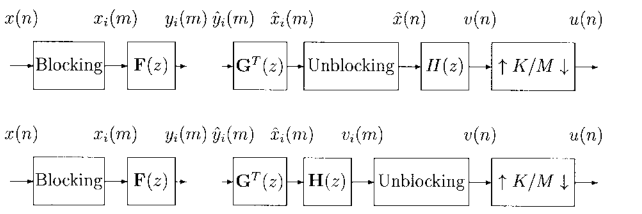


Fig. 6. The analysis and synthesis systems with nonuniform downsampling. The filter can be implemented in its polyphase components and moved past the unblocking device.

Noting that $X_{-n}(z) = z^{-1}X_{M-n}(z)$ for $0 < n < M$, we arrive at [4]

$$\mathbf{H}(z) = \begin{bmatrix} H_0(z) & z^{-1}H_{M-1}(z) & \cdots & z^{-1}H_1(z) \\ H_1(z) & H_0(z) & \cdots & z^{-1}H_2(z) \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1}(z) & H_{M-2}(z) & \cdots & H_0(z) \end{bmatrix}. \tag{3}$$

While (1) represents a general LPTV system, an LTI system has to obey (3). It is also known that a paraunitary system obeying (3) yields an allpass LTI filter, and vice versa [4]. Constraining the system to be LTI, FIR, and paraunitary, we can show that $H(z)$ is a delay and only one $H_i(z)$ is nonzero. Hence, a nontrivial paraunitary FIR MIMO system always represents an LPTV filter.

Similarly, we can describe the analysis and synthesis systems using PTM's as in Fig. 4. The signal is blocked and passed through the analysis PTM $\mathbf{F}(z)$. It is reconstructed from the subbands using the PTM $\mathbf{G}^T(z)$ followed by an unblocking device. $\mathbf{G}(z)$ is transposed to maintain our notation, since the columns of $\mathbf{G}^T(z)$ (rows of $\mathbf{G}(z)$) will actually correspond to the filters $g_k(n)$. As $L = NM$, the PTM's have entries of order $N-1$. In practice, neither PTM represents an LTI system. However, if we let $\hat{\mathbf{y}}(z) = \mathbf{y}(z)$, and define

$$\mathbf{T}(z) = \mathbf{G}^T(z)\mathbf{F}(z) \tag{4}$$

such that $\hat{\mathbf{x}}(z) = \mathbf{G}^T(z)\mathbf{F}(z)\mathbf{x}(z) = \mathbf{T}(z)\mathbf{x}(z)$, the perfect reconstruction (PR) property of filterbanks requires that [4], [6]

$$\mathbf{T}(z) = z^{-N+1}\mathbf{I}_M \tag{5}$$

and the overall system is an LTI filter (a pure delay).[2] Hence, $T(z) = z^{-L+1}$, $t(n) = \delta(n-L+1)$ and $\hat{x}(n) = x(n-L+1)$.

III. Common Resampling Methods

The image is only available at the reference resolution. As we are *creating* a new image, there is no *target* image to compare results of alternative resampling methods. There are several ways to accomplish this operation, but we discuss two main approaches.

A. Resampling as Postprocessing

The most straightforward (and the most used) method to rescale the compressed image is to decompress the image performing an inverse transform and, then, rescale the resulting image. The complete analysis, synthesis, and rescaling flow diagram is shown in Fig. 5.

We simplify the problem to allow only resampling of K/M times the original sampling rate. In this case, the general uniform scaling method of Fig. 5 can be immediately used by setting $P = K$ and $Q = M$. The drawback of such an approach resides in its high implementation complexity. First, the image has to be completely recovered by performing an inverse transform in its entirety. Second, the processing is applied to the full image. In theory, the filtering should be done in an even higher sampling rate (K times higher). Although the filter can be efficiently implemented using its polyphase components, it still requires a relatively high computational load.

A simplification can be achieved if we allow nonuniform resampling and assume $K < M$. For this, we employ the symbol $\uparrow K/M \downarrow$ in Fig. 6, which means retain K out of M samples. More generically, one can retain nK out of nM samples. In Fig. 6 the filter $h(n)$ is a lowpass with cutoff on $K\pi/M$ to avoid excessive aliasing. Then, the filtered signal $v(n)$ is resampled to the final sampling rate yielding $u(n)$. Implementing $h(n)$ in the form of (3), it is easy to see that the system $\mathbf{H}(z)$ can be moved past the unblocking device as shown in Fig. 6. Hence

$$\mathbf{v}(z) = \mathbf{H}(z)\mathbf{G}^T(z)\mathbf{F}(z)\mathbf{x}(z) = \mathbf{H}(z)\mathbf{T}(z)\mathbf{x}(z) \tag{6}$$

and for PR filter banks

$$V(z) = z^{-L+1}H(z)X(z). \tag{7}$$

The overall system is, as expected, an LTI filter followed by a resampler. If M/K is an integer, the system becomes a trivial uniform downsampler, where $P = 1$ and $Q = M/K$, such that the filter may have its cutoff at π/Q.

The synthesis system is effectively

$$\mathbf{S}(z) = \mathbf{H}(z)\mathbf{G}^T(z). \tag{8}$$

The effective synthesis filters are

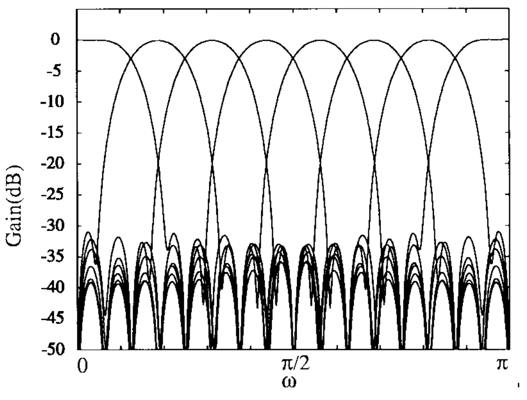
$$s_k(n) = h(n) * g_k(n). \tag{9}$$

The filter may not be explicitly implemented after regular synthesis. This possibility apparently reduces the complexity of the reconstruction process, since we may only design and implement $\mathbf{S}(z)$. However, the main disadvantage is that generally $\mathbf{G}(z)$ is designed to possess a fast implementation algorithm. Such fast implementation may be lost by implementing $\mathbf{S}(z)$, instead. Therefore, in certain cases, it still may be advantageous to implement the filter after the regular synthesis.

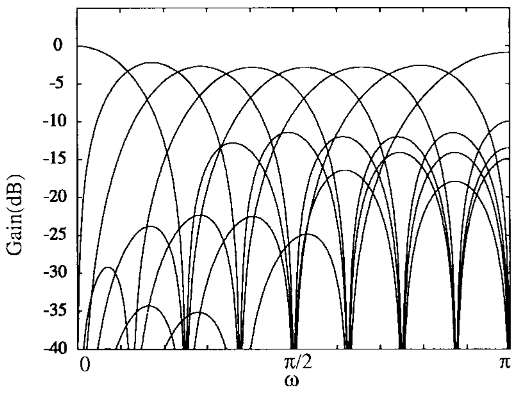
B. Discarding Subbands

A computationally efficient way to implement the filter may borrow the filtering properties (stopband attenuation, etc.) of the filters $f_k(n)$ and $g_k(n)$. If the filters have good stopband attenuation, we can simply discard the highest frequency $M-K$ subbands and keep the lower K subbands. Then, we

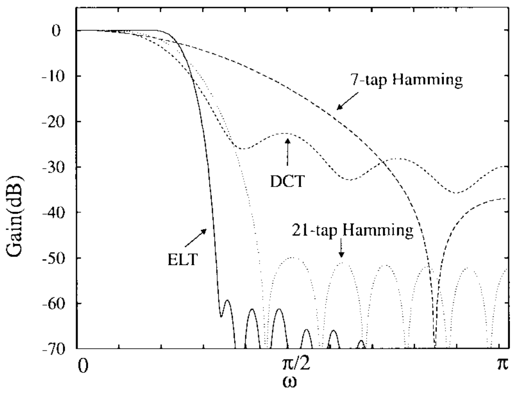
[2]The results in this paper may also work for low-delay filterbanks [12].



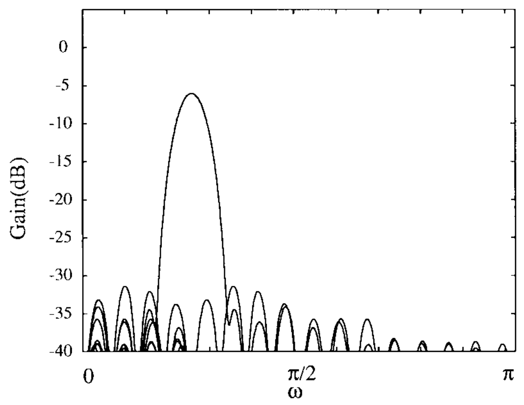
(a)



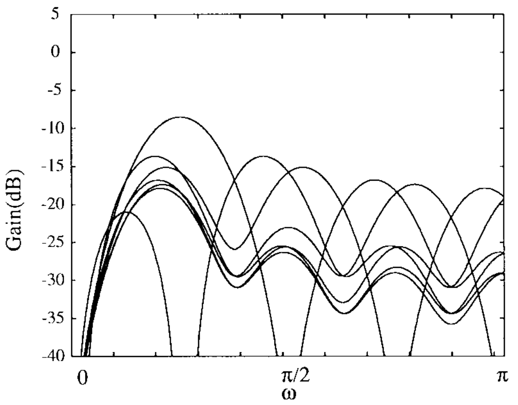
(b)



(c)



(d)



(e)

Fig. 7. Frequency response plots for several filters useful for downsampling a signal at 4 : 1 ratio ($M = 8$ and $K = 2$). (a) The eight filters for the ELT (32 taps each). (b) The eight filters for the DCT (eight taps each). (c) The principal filters $A_0(e^{j\omega})$ for the DCT and ELT are compared to Hamming-window filters with seven and 21 taps designed for cutoff at $\pi/4$. (d) The remaining filters $A_l(e^{j\omega})$ for the ELT. (e) Same for the DCT.

can either set $y_i(n) = 0$ for $K \leq i \leq M-1$, or set $g_i(n) = 0$ for $K \leq i \leq M-1$ and $0 \leq n \leq L-1$. Let

$$\mathbf{I}'_{K,M} = \text{diag}\{\underbrace{1, 1, \cdots, 1}_{K\ 1's}, \underbrace{0, 0, \cdots, 0}_{M-K\ 0's}\}. \tag{10}$$

The equivalent synthesis system is given by

$$\mathbf{S}(z) = \mathbf{G}^T(z)\mathbf{I}'_{K,M}. \tag{11}$$

In this case, $\mathbf{T}(z) = \mathbf{G}^T(z)\mathbf{I}'_{K,M}\mathbf{F}(z)$ and

$$T_{ij}(z) = \sum_{l=0}^{K-1} G_{li}(z)F_{lj}(z). \tag{12}$$

Unless severe restrictions are imposed to the filterbank design, $\mathbf{T}(z)$ will not represent an LTI system obeying (3). Let $W = e^{-j2\pi/M}$. The overall transfer in a filterbank is governed by [4]

$$\hat{X}(z) = \frac{1}{M}\sum_{l=0}^{M-1}\left(\sum_{k=0}^{M-1} G_k(zW^l)F_k(z)\right)X(zW^l)$$
$$= \frac{1}{M}\sum_{l=0}^{M-1} A_l(z)X(zW^l).$$

Aliasing canceling systems are designed so that $A_l(z) = 0$ for $l > 0$. Hence, $T(z) = (1/M)A_0(z)$. If we set $F_k(z) = 0$ for $k \geq K$ then $A_l(z) = \sum_{k=0}^{K-1} G_k(zW^l)F_k(z)$ and even alias canceling systems will lead to $A_l(z) \neq 0$ for $l > 0$ after discarding subbands. However, if $A_l(z)$ is small $(\forall z, l \neq 0)$ we have approximately

$$T(z) \approx \frac{1}{M}A_0(z) = \frac{1}{M}\sum_{k=0}^{K-1} G_k(z)F_k(z). \tag{13}$$

Concluding, if the aliasing terms are sufficiently small, the equivalent filter is approximately LTI and given by

$$H_{\text{LPTV}} \approx H_{\text{LTI}}(e^{j\omega}) = \frac{1}{M}A_0(e^{j\omega}). \tag{14}$$

Therefore, "good" filters may approximate a reasonably selective equivalent filter after discarding subbands. Note that we cannot compare the approach just described with the method described in the previous section. The reason for this is because by discarding subbands we arrive at an LPTV filter, which is different in nature than traditional LTI filters. However, we can consider LTI filters as a subset of LPTV filters. The approximation in (14) is useful to give a reference point for comparisons because $A_0(z)$ is an LTI filter. Fig. 7 shows a comparison of the frequency response of the LPTV filter and of two LTI filters. Note that secondary filters [Fig. 7(d) and (e)] are less attenuated in the transition region of the main filter. Also note that lowpass regions are well attenuated in both cases. It is well known that for image filtering, the best filter is not correlated to the best approximation to the ideal filter (i.e., the infinite-length sinc filter), not only because the image is itself a finite segment but also due to nonlinearities of the human visual system. Subjective evaluation will be carried later on and, for the time being, it suffices to point out that the operation of discarding subbands can yield a reasonable filtering operation.

IV. Block Resampling

A. Downsampling

The goal in this paper is to directly design a synthesis system that would output K samples of $u(n)$ for every block of M subband samples ($y_i(m)$ for $0 \leq i \leq M-1$). In this case, we can design a synthesis PTM such that

$$\mathbf{S}(z) = \begin{bmatrix} S_{00}(z) & S_{01}(z) & \cdots & S_{0,M-1}(z) \\ S_{10}(z) & S_{11}(z) & \cdots & S_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ S_{K-1,0}(z) & S_{K-1,1}(z) & \cdots & S_{K-1,M-1}(z) \end{bmatrix}. \tag{15}$$

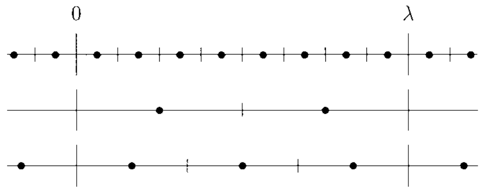


Fig. 8. Interval of a block originally with eight samples, is resampled at two and three samples per block. The interval of the block is defined as ranging from zero to λ, and K uniform cells fill the intervals. The sampling occurs at the center of the cell.

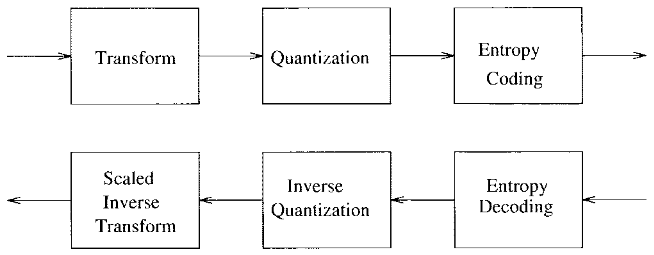


Fig. 9. General diagram for a JPEG-like image compression system with downsampled inverse.

Thus, $\mathbf{T}(z)$, the overall transfer, is now a $K \times M$ matrix given by

$$\mathbf{T}(z) = \mathbf{S}(z)\mathbf{F}(z). \tag{16}$$

$\mathbf{S}(z)$ implies filtering and resampling. We can design $\mathbf{S}(z)$ by block resampling of the output signal of a regular analysis–synthesis system and implement filtering by processing the subbands, i.e.,

$$\mathbf{S}(z) = \mathbf{\Phi}(z)\mathbf{G}^T(z)\mathbf{C}(z) \tag{17}$$

where $\mathbf{\Phi}(z)$ is a $K \times M$ resampling matrix and $\mathbf{C}(z)$ is the filtering operator. For example, if $\mathbf{C}(z)$ is a diagonal matrix with zero order entries, it will perform filtering by solely weighting the subband coefficients. Let

$$\mathbf{v}(z) = \mathbf{G}^T(z)\mathbf{C}(z)\mathbf{F}(z)\mathbf{x}(z) \tag{18}$$

so that

$$\mathbf{u}(z) = \mathbf{\Phi}(z)\mathbf{v}(z). \tag{19}$$

The signal $v(n)$ has M samples per block, while the final signal $u(n)$ has only K samples per block. The design of the resampling matrix $\mathbf{\Phi}(z)$ can be accomplished in several ways [4], [13], [14]. We have chosen to approximate the signal $v(n)$ by constructing a continuous curve from which $v(n)$ can be found by uniform sampling. Then, we resample the continuous functions at the desired rate to obtain $u(n)$. This approach will only work if the samples of $v(n)$ are smooth enough to generate smooth curves. The assumption is that the LPTV filter assumed by processing subbands will do a good job of removing high-frequency components. Also, we may have to find a suitable interpolation function. Linear interpolation (fitting a straight line in between every two samples of $u(n)$) is generally visually pleasing. Splines and higher order polynomials may be used as well [13], [14].

The sampling grid we have chosen is shown in Fig. 8. In Fig. 8, the interval of a block originally with eight samples, is resampled at two and three samples per block. The interval of the block is defined as ranging from zero to λ, and K uniform cells fill the intervals. The sampling occurs at the center of the cell. This particular sampling grid seems to make the resampling task unnecessarily more complicated, because, for example, if $K = 2$ and $M = 8$, we cannot simply subsample the output by a factor of 4. However, we will explain later how this sampling grid can lead to fast implementation algorithms.

It is straightforward to design $\mathbf{\Phi}(z)$ to allow the proposed resampling. For example, with a piecewise linear interpolation between samples, $M = 8$ and $K = 2$, we have

$$\mathbf{\Phi}(z) = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}.$$

In another example, assume $M = 8, K = 3$, and a higher order interpolation. In this case, for a particular block, the original samples $v_m(n)$ are located in the interval $[0, \lambda)$ at positions $(m + 0.5)\lambda/8$. The final samples $u_m(n)$ are located at $(m + 0.5)\lambda/3$. Let the interpolation formulas be

$$\begin{aligned} u_0(n) &= av_7(n-1) + bv_0(n) + cv_1(n) + dv_2(n) \\ u_1(n) &= ev_2(n) + fv_3(n) + fv_4(n) + ev_5(n) \\ u_2(n) &= dv_5(n) + cv_6(n) + bv_7(n) + av_1(n+1) \end{aligned}$$

for a through f as real constants. The resampling matrix is no longer memoryless as the interpolation extends beyond the block boundaries. Hence

$$\mathbf{\Phi}(z) = \begin{bmatrix} bz^{-1} & cz^{-1} & dz^{-1} & 0 & 0 & 0 & 0 & az^{-2} \\ 0 & 0 & ez^{-1} & fz^{-1} & fz^{-1} & ez^{-1} & 0 & 0 \\ a & 0 & 0 & 0 & 0 & dz^{-1} & cz^{-1} & bz^{-1} \end{bmatrix}.$$

For the filtering operator, we can further simplify the problem by using the approach in Section III-B. Then, filtering is accomplished by discarding the highest frequency subbands and retaining only the K lower frequency ones. Let the polyphase components of $u(n)$ be $u_i(n) = u(nK + i)$ for $(0 \le i \le K - 1)$. Let $\bar{\mathbf{y}}(z)$ be the vector with the K lowest frequency subbands. We want to design a PTM, which we denote by $\mathbf{R}(z)$, directly relating these K subbands to the K output polyphase components as

$$\underbrace{\begin{bmatrix} U_0(z) \\ U_1(z) \\ \vdots \\ U_{K-1}(z) \end{bmatrix}}_{\mathbf{u}(z)} = \underbrace{\begin{bmatrix} S_{00}(z) & S_{01}(z) & \cdots & S_{0,K-1}(z) \\ S_{10}(z) & S_{11}(z) & \cdots & S_{1,K-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ S_{K-1,0}(z) & S_{K-1,1}(z) & \cdots & S_{K-1,K-1}(z) \end{bmatrix}}_{\mathbf{R}^T(z)} \times \underbrace{\begin{bmatrix} \hat{Y}_0(z) \\ \hat{Y}_1(z) \\ \vdots \\ \hat{Y}_{K-1}(z) \end{bmatrix}}_{\bar{\mathbf{y}}(z)}. \tag{20}$$

For this case, $\mathbf{C}(z) = \mathbf{I}'_{K,M}$ following (10). Let $\mathbf{\Lambda} = [\mathbf{I}_K, \mathbf{0}_{K \times M-K}]$ in such a way that $\bar{\mathbf{y}}(z) = \mathbf{\Lambda}\hat{\mathbf{y}}(z)$ and let $\bar{\mathbf{G}}(z) = \mathbf{\Lambda}\mathbf{G}(z)$. Hence

$$\mathbf{u}(z) = \mathbf{R}^T(z)\mathbf{\Lambda}\hat{\mathbf{y}}(z) = \mathbf{R}^T(z)\mathbf{\Lambda}\mathbf{F}(z)\mathbf{x}(z)$$

so that

$$\mathbf{S}(z) = \mathbf{R}^T(z)\mathbf{\Lambda}. \tag{21}$$

As $\mathbf{C}(z) = \mathbf{I}'_{K,M} = \mathbf{\Lambda}^T\mathbf{\Lambda}$, we have that

$$\mathbf{S}(z) = \mathbf{\Phi}(z)\mathbf{G}^T(z)\mathbf{\Lambda}^T\mathbf{\Lambda} \tag{22}$$

and

$$\mathbf{R}^T(z) = \mathbf{\Phi}(z)\mathbf{G}^T(z)\mathbf{\Lambda}^T = \mathbf{\Phi}(z)\bar{\mathbf{G}}^T(z). \tag{23}$$

This result implies that the K synthesis filters should be resampled versions of the K filters corresponding to the K lowest frequency subbands of the original synthesis filterbank $\mathbf{G}(z)$. Thus, the actual synthesis filters $r_k(n)$ have length NK and are found as[3]

$$\mathbf{r}_k(z) = \mathbf{\Phi}(z)\mathbf{g}_k(z). \tag{24}$$

By interpolating the impulse response of the filters, we just need to interpolate (and resample) one function for each block of the impulse response of the filter. We can also directly define a continuous function for the whole support of the filters generating K different continuous functions, which we denote by $\mu_k(t)$, for $0 \le k \le K - 1$. Thus,

$$r_k(n) = \mu_k\left(\frac{2n+1}{2K}\lambda\right) \tag{25}$$

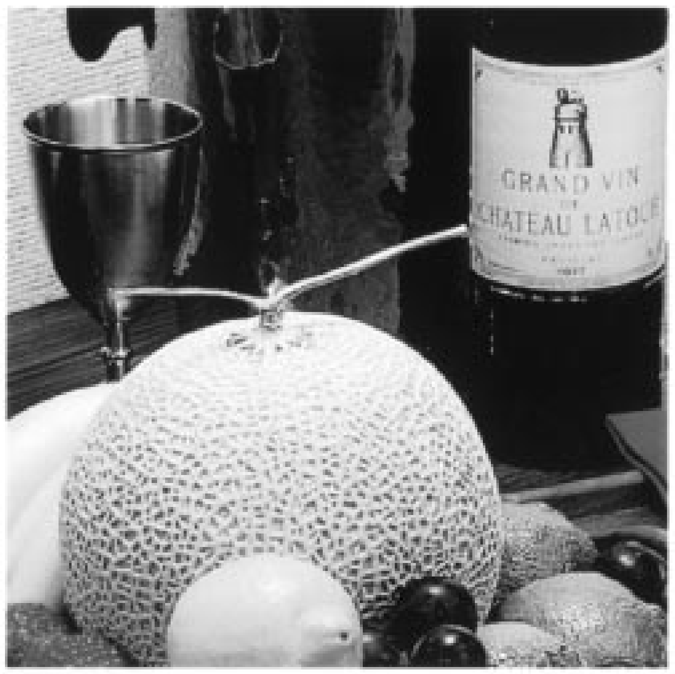
for λ representing the normalized support of M samples of the impulse response of $g_k(n)$.

B. Upsampling

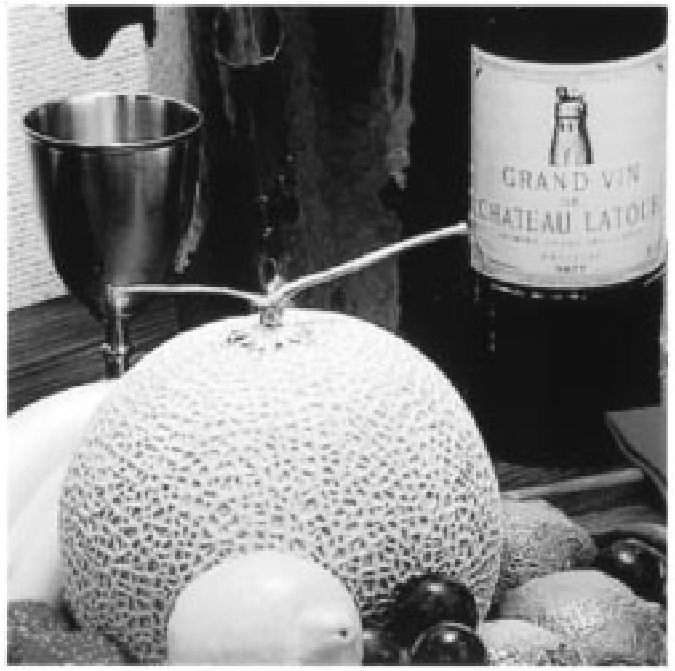
The reader may have noted that we have not devoted time to explain upsampling methods so far. The reason for this is that sometimes the use of a larger inverse transform may not be competitive (in terms of computational complexity) to a straightforward inverse followed by a simple interpolation method, such as linear interpolation. For completeness and for the cases where it may be advantageous to redesign the inverse transform for a larger resolution output, portions of the previous discussion are still applicable. Obviously, $K > M$, and both $\mathbf{T}(z)$ and $\mathbf{S}(z)$ are $K \times M$ matrices. Equations (15) through (19) are still valid, and so are the equations and comments pertaining to the resampling matrix $\mathbf{\Phi}(z)$. The resampling method is the same and Fig. 8 can still be useful as long as $M = 2$ and $K = 3, 8$ or $M = 3$ and $K = 8$.

[3]One could use a different resampling matrix for each filter, i.e., $\mathbf{r}_k(z) = \mathbf{\Phi}_k(z)\mathbf{g}_k(z)$.

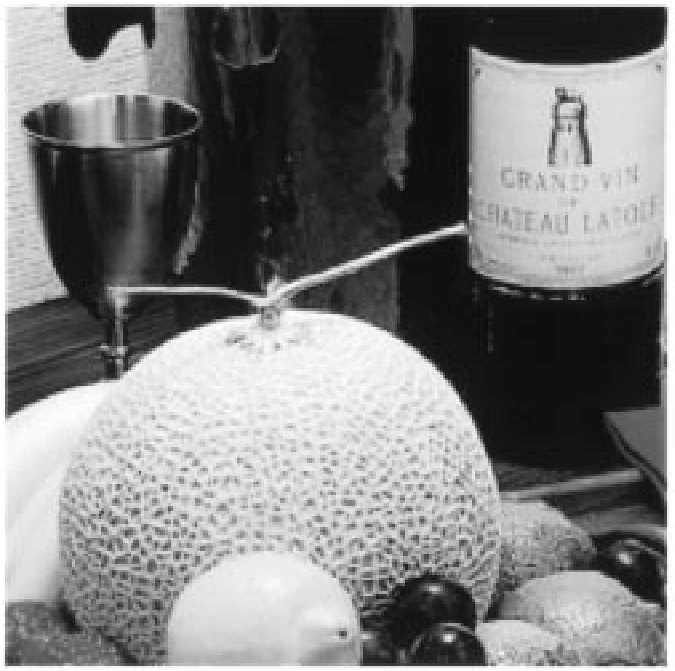
(a)



(b)



(c)



(d)

Fig. 10. JPEG-decompressed image with 800 × 800 pixels and its reconstruction at different resolutions using the fast downscaled inverse DCT. (a) 800 × 8 (b) 400 × 400. (c) 300 × 300. (d) 200 × 200.

We may also want to set $\mathbf{C}(z) = \mathbf{I}_M$ for simplification. The idea is that the M lower frequency filters of the hypothetical K-channel filter bank are upsampled versions of the regular M-channel synthesis $\mathbf{G}^T(z)$. The higher frequency filters do not exist, and this is equivalent to filling in higher frequency subbands with zero coefficients.

C. General Resampling Factors

In all cases the resampling factor is always $M : K$, for an integer K. In cases where it is necessary to have a noninteger K, the proposed method does not apply. The only comfort is to use it as a first approximation in the downsampling operation. Then, we set K as the closest integer smaller than the factor required and perform an interpolation (perhaps linear) in the space domain. The savings arise from the fact that we avoid using the full inverse transform. It is basically the complexity difference between performing a full $M \times M$ inverse and performing a reduced $K \times K$ inverse transform.

V. Fast Transforms

For full decompression, we assume that the synthesis system that will lead to PR is $\mathbf{G}(z)$, which is fixed. For each $\mathbf{G}(z)$ and for each value of K we may design $\mathbf{R}(z)$. Here, we follow one of two approaches:

- design $\mathbf{\Phi}(z)$ for given $\mathbf{G}(z)$;
- design $\mathbf{R}(z)$ possessing a fast algorithm to approximate given $\mathbf{G}(z)$ and $\mathbf{\Phi}(z)$.

In the last section we used the first approach. In this section, we use the second one because we are only looking for transforms with fast implementation algorithms.

In all image coding tests, a JPEG baseline coder was used [11] and, whenever necessary, the eight-channel DCT was replaced by the respective transform. The test image was always originally compressed at 1 b/pixel before being decompressed to the desired resolution.

A. DCT and JPEG

An example of a general diagram for a downscaled decompression of JPEG-compressed images is shown in Fig. 9. The basic steps in JPEG baseline compression [11] are as follows.

1) DCT is applied to 8 × 8 blocks ($M = 8$).
2) Each block is converted into a vector by scanning in zigzag order.
3) Coefficients are quantized.
4) One-dimensional (1-D) lossless differential pulse code modulation (DPCM) is applied to the DC coefficients.
5) Resulting vector is input to a lossless Huffman coder. The decompression is performed by inverting the order and function of those steps.

The $M \times M$ DCT is defined by a matrix denoted $\mathbf{D}_M$ whose elements are

$$d_{ij} = \sqrt{\frac{2}{M}}\alpha_i \cos\left(\frac{(2j+1)i\pi}{2M}\right) \tag{26}$$

for $0 \leq (i,j) \leq M-1$ and $\alpha_0 = 1/\sqrt(2)$ and $\alpha_i = 1$ for $i > 0$. If we resample each basis function of the DCT by directly defining the interpolating function as

$$\mu_k(t) = \alpha_k\sqrt{\frac{2}{M}}\cos(tk\pi). \tag{27}$$

It is easy to see that the DCT of all sizes will be composed by samples of this function. Hence, we simply use

$$\mathbf{R}^T(z) = \sqrt{\frac{K}{M}}\mathbf{D}_K^T. \tag{28}$$

For two-dimensional (2-D) resampling as is the case with JPEG, let the transformed block with 8 × 8 coefficients be denoted by $\mathbf{Y}$ and the block of $K \times K$ reconstructed pixels (for this block) be represented by $\mathbf{X}$. Also, let the block with the $K \times K$ lowest frequency coefficients of $\mathbf{Y}$ be denoted by $\bar{\mathbf{Y}}$. The reconstruction of the pixels is given by

$$\mathbf{X} = \frac{K}{M}\mathbf{D}_K^T\bar{\mathbf{Y}}\mathbf{D}_K. \tag{29}$$

Resizing in the DCT domain has been studied before [15], [16]. Similar resampling methods for the DCT have also been recently reported [17], [18].

An 800 × 800-pixel test image was compressed with JPEG (DCT $M = 8$). Fig. 10 shows its reconstruction using the present method for different resolutions.

The problem described in Fig. 1 is a factor of 4 resolution reduction in each direction. Assume the image is compressed using JPEG. In this special case and using $K = 2$ the synthesis is simplified to

$$\begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{30}$$

where x_{ij} are the 2 × 2 reconstructed pixels and y_{ij} are the 2 × 2 lowest frequency DCT coefficients in a block (originally with 8 × 8 coefficients). Note that no floating point numbers and no multiplications are involved. If the dequantized coefficients (integer numbers for JPEG) are still organized in a zigzag-scanned vector, the pixels can be reconstructed with the following C code:

```
/* zz[] is the vector in
       zigzag ordering */
t1 = zz[0] + zz[2];
t2 = zz[1] + zz[4];
t3 = zz[0] - zz[2];
t4 = zz[1] - zz[4];
y[0][0] = (t1 + t2) >> 3;
y[0][1] = (t1 - t2) >> 3;
y[1][0] = (t3 + t4) >> 3;
y[1][1] = (t3 - t4) >> 3;
```

An alternative to this approach is to discard the high-frequency coefficients and to decimate the output by a factor of 4. This can be efficiently implemented using a pruned inverse DCT as

$$\mathbf{\Phi}(z) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{31}$$

so that

$$\begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} = \begin{bmatrix} 0.3536 & 0.4904 \\ 0.3536 & -0.0975 \end{bmatrix}\begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix} \times \begin{bmatrix} 0.3536 & 0.3536 \\ 0.4904 & -0.0975 \end{bmatrix}. \tag{32}$$

This method involves multiplications and floating point numbers, while the proposed method does not.

For values of K other than 2, floating point numbers and multiplications are used. However, the inverse transform can always be implemented through a fast implementation of the $K \times K$ inverse DCT. There are very fast implementation algorithms available for the DCT [5] (for virtually all sizes and for the multidimensional case).

Fig. 11 shows a comparison of methods for reconstructing a 200 × 200-pels approximation of a compressed image

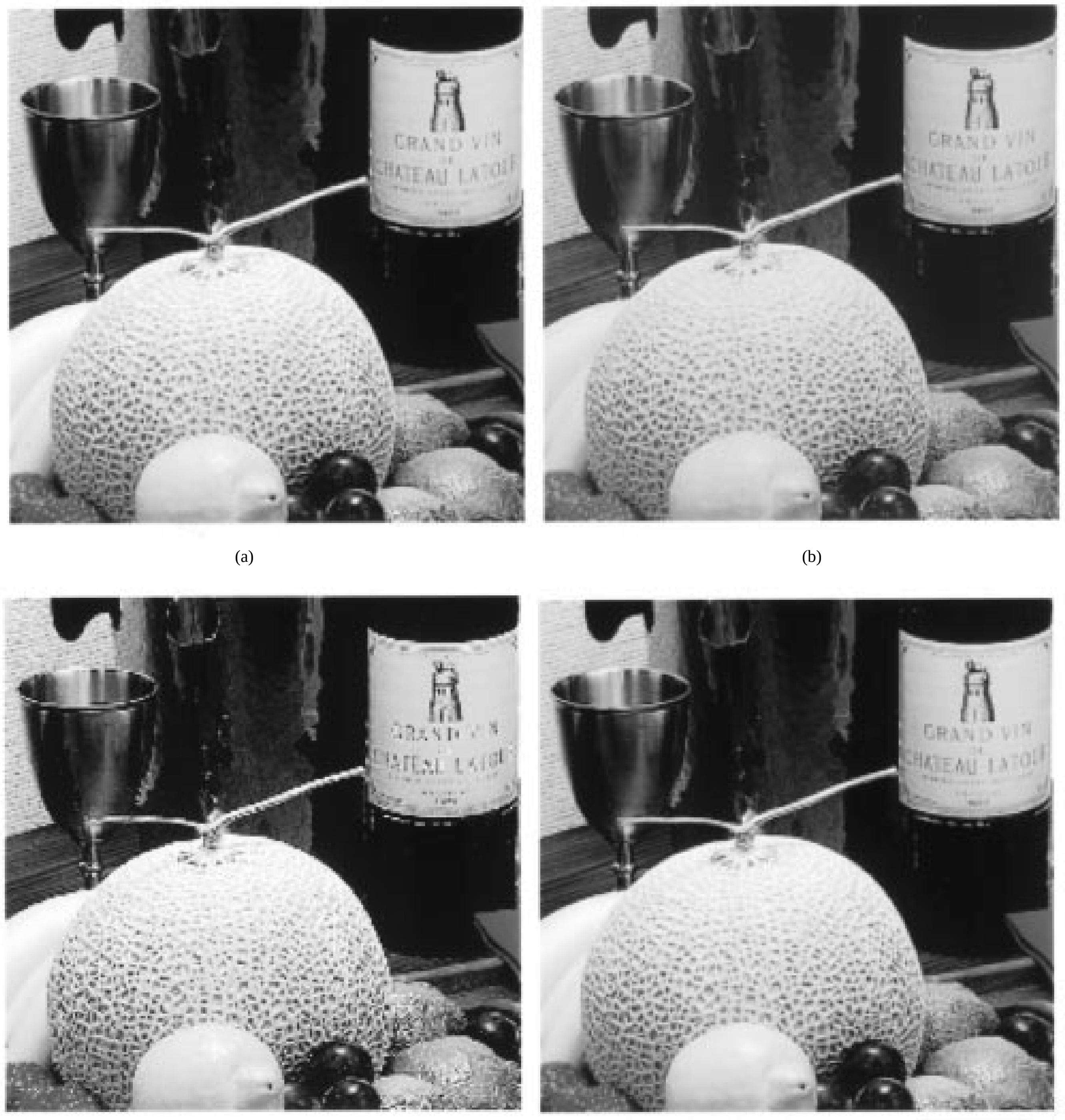
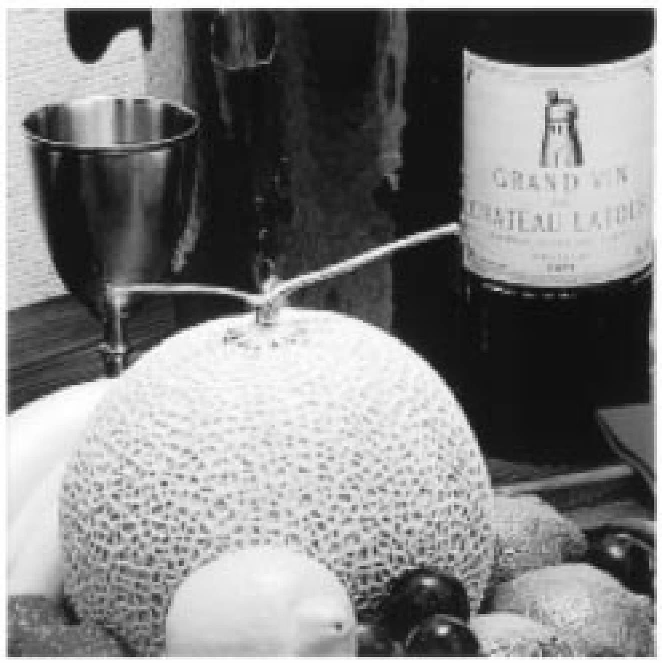
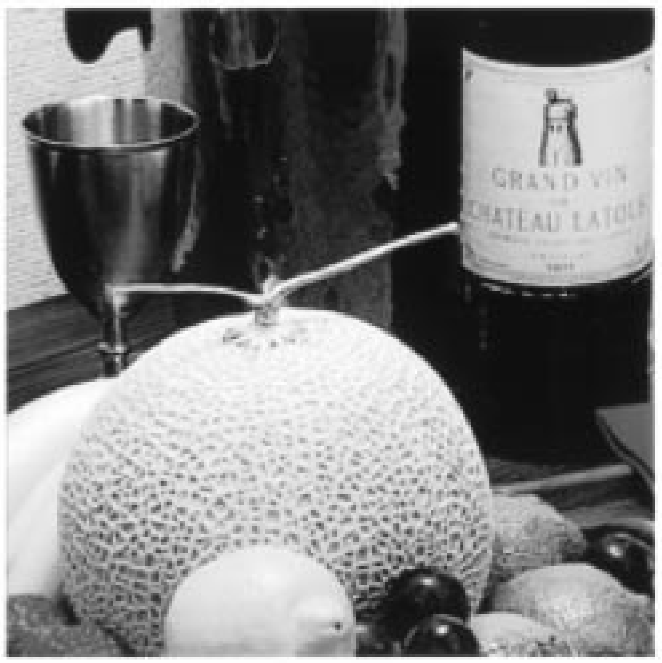


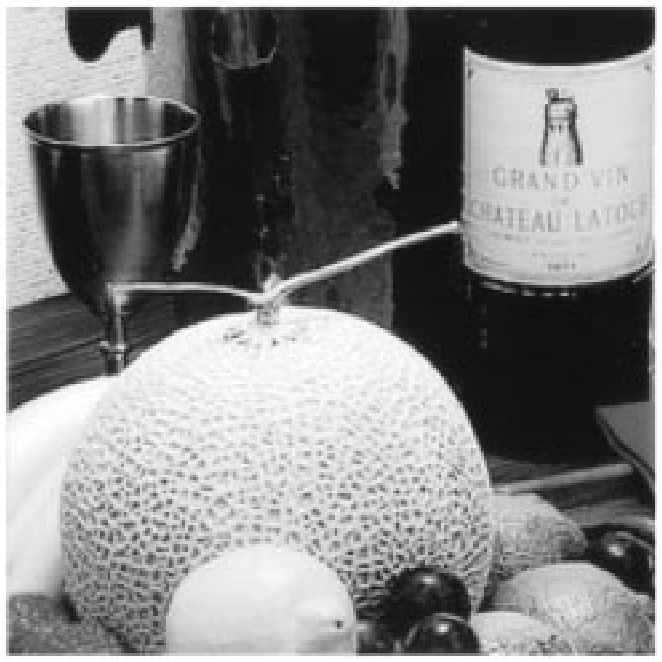
Fig. 11. Comparison of fast inverse methods for DCT (JPEG) with $M = 8$ and $K = 2$. (a) Proposed method without floating point numbers or multiplications. (b) Pruned inverse DCT. (c) Decimation without prefiltering. (d) Averaging over blocks of 4 × 4 pixels after decompression.

originally at 800 × 800-pels, i.e., $M = 8$ and $K = 2$. The compression method is JPEG (hence, using DCT). Fig. 11(a) shows the resulting image for the proposed method following (30), with no floating point number and no multiplication. A similar result is shown in Fig. 11(b) for a pruned DCT following (32). Fig. 11(c) is the resulting image after simply decimating the fully decompressed image. Fig. 11(d) is the resulting image after subsampling by averaging blocks of 4 × 4 pixels of the fully decompressed image. We also compared the method to better prefiltering methods followed by decimation. Figs. 11(e) and (f) show the result for a separable lowpass filter with cutoff at $\pi/4$ designed with the Hamming window. The filter sizes are 7 × 7 and 21× 21. Fig. 11(g) is the result using a 21 × 21 rectangular window (truncated sinc) while Fig. 11(h) shows the result using a filter implemented by zeroing out all but the lowest frequency coefficients of the fast Fourier transform (FFT) of the whole image.

From these results it is easy to see that the proposed method is not only simpler than its competitors, but it also ranks among


(e)


(f)


(g)

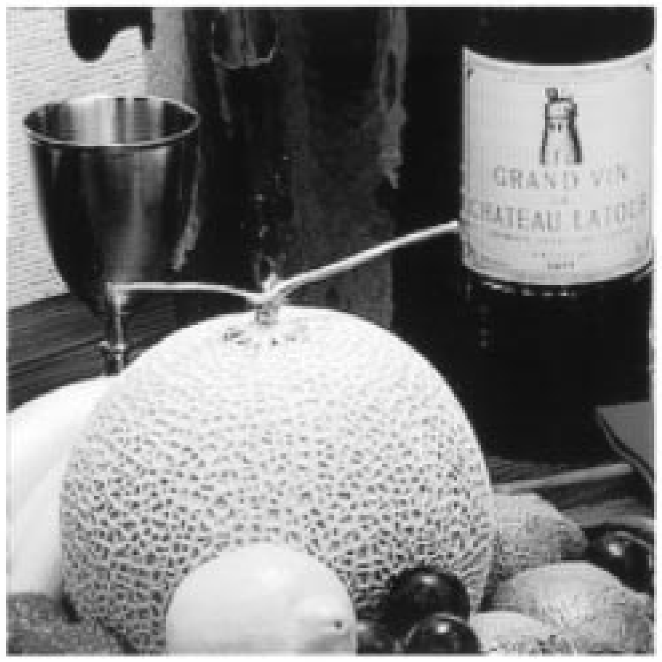

(h)

Fig. 11 *(Continued).* Comparison of fast inverse methods for DCT (JPEG) with $M = 8$ and $K = 2$. (e) Using a 7×7 Hamming window-based prefilter. (f) Using a $21\times$ 21 Hamming window-based prefilter. (g) Using a 21×21 rectangular window (truncated sinc) prefilter. (h) Filtering by applying rectangular window to DFT coefficients.

the top performers. Its performance is occasionally slightly inferior to the expensive methods shown in Fig. 11(e)–(g). Overall, the results of and Fig. 11(a) and Fig. 11(e)–(g) are comparable. Objective measures such as mean square error (MSE) do not reflect image quality in this case, mainly because we do not have a reference image to compare with. For example, if we let Fig. 11(e) be the reference, the images in Fig. 11(a) and (c) have similar MSE despite looking radically different.

In terms of complexity, let both additions and multiplications be counted as one operation regardless of the precision. For example, let us use the results given in Fig. 11. Then, the number of operations necessary to reconstruct one block of 2×2 pixels of the output image for each method is

$$\begin{array}{cccccccc} (a) & (b) & (c) & (d) & (e) & (f) & (g) & (h) \\ 12 & 20 & 672 & 736 & 1060 & 4196 & 4196 & >5000. \end{array}$$

Note that we assumed a separable implementation for the DCT based on a fast (but not necessarily state-of-the-art) 1-D

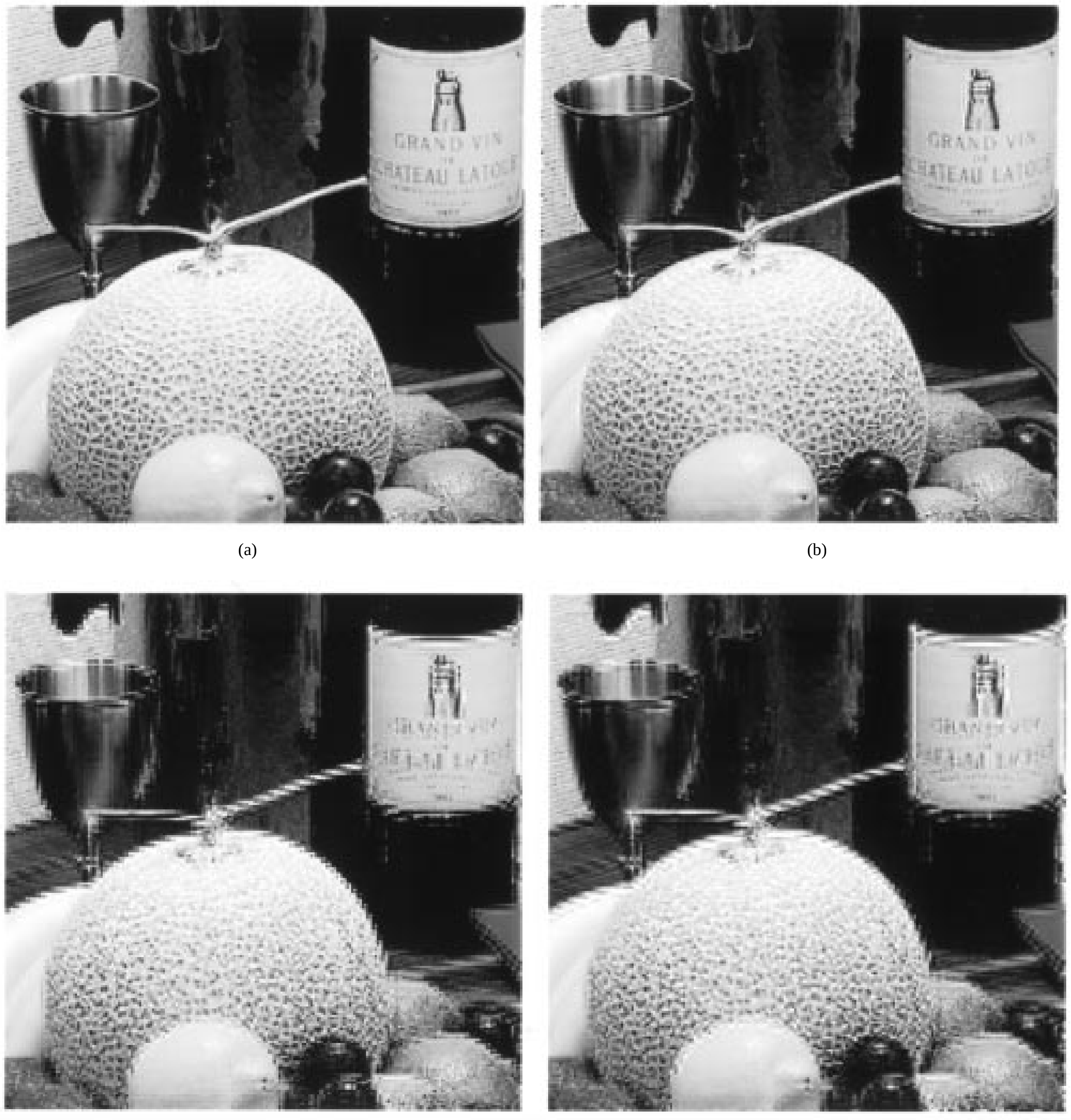


Fig. 12. Comparison of inverse mismatch for the fast inverse ELT with $M = 8$ and $K = 2$. (a) Proposed method using a similar continuous window for analysis and synthesis. (b) Same as (a) with window mismatch. (c) Reconstructing the image with the inverse DCT. (d) Reconstructing the image with same window but different cosine phase.

algorithm [6]. We also assumed a polyphase implementation of filters [4] without optimization for symmetry. The complexity of method c) is the complexity of the inverse 8×8 DCT and as K increases the complexity of a) will gradually approach this number. It is possible to save about 100 operations in methods c)–h) by using faster DCT implementations. However, it is also possible to remove four operations in methods a) and b). We can scale the matrices in (32) and we can include any multiplication factor into the JPEG quantizer table elements. Although methods c)–h) can be more efficiently implemented they would still be largely more complex than methods a)–b).

B. Other Sinusoidal Transforms and Cosine Modulated Filterbanks

Using the same reasoning applied to the DCT, we can use inverse transforms of smaller sizes to downsize other sinusoidal transforms such as other members of the family of the DCT and DST (including DCT and DST of type IV) [5], [6]. Hence, we have the formulations shown on the bottom of the page. In all cases, if the forward transform is given by matrix $\mathbf{D}_M^{\text{type}}$ we have

$$\mathbf{R}^T(z) = \sqrt{\frac{K}{M}}\mathbf{D}_K^{\text{type }T}. \tag{33}$$

Cosine modulated filterbanks can be applied as well by using exactly the same principles. Let us use an example which is the family of ELT's [6], [10]. The M filters of an ELT are

$$g_k(n) = h(n)\sqrt{\frac{2}{M}}\cos\left[\left(k+\frac{1}{2}\right)\frac{\pi}{M}\left(n+\frac{M+1}{2}\right)\right] \tag{34}$$

for $k = 0, 1 \cdots, M-1$ and $n = 0, 1, \cdots, L-1$. $h(n)$ is a window modulating the cosine terms. It represents a lowpass prototype filter that is then translated by modulation to the several bandpass filters found in the filterbank. A good choice for interpolation function is

$$\mu_k(t) = h'(t)\sqrt{\frac{2}{M}}\cos\left[\left(k+\frac{1}{2}\right)\left(t+\frac{\pi}{2}\right)\right] \tag{35}$$

where $h'(t)$ is a suitable curve interpolating $h(n)$ and represents the impulse response of a lowpass continuous filter. If the synthesis filter bank for the M-channel ELT is represented by $\mathbf{E}_M(z)$, the downsized inverse ELT can be applied to resample the signal.

$$\mathbf{R}^T(z) = \sqrt{\frac{K}{M}}\mathbf{E}_K^T(z). \tag{36}$$

The above result can only be used successfully if we can find a suitable function $h'(t)$ from which the modulating windows for the K-channel $(h^{(K)}(n))$ and the M-channel $(h^{(M)}(n))$ ELT could be derived. i.e.,

$$h^{(M)}(n) = h(M, n) \text{ and } h^{(K)}(n) = h(K, n) \tag{37}$$

where

$$h(k, n) = h'\left(\frac{2n+1}{2k}\lambda\right). \tag{38}$$

The conditions in (37) and (38) do not need to be fully met, but a good approximation will lead to a high-quality reconstruction. A comparison in this sense is carried in Fig. 12 for $M = 8$ and $K = 2$, using JPEG and replacing the DCT by the eight-channel ELT. Fig. 12(a) shows the reconstructed image when (37) and (38) are closely matched. Fig. 12(b) is the same as 12(a) but with a mismatch of the windows used in analysis and synthesis. To verify the susceptibility of the method to even larger mismatches, Fig. 12(c) shows the reconstructed image using the two-channel DCT for reconstruction, while Fig. 12(d) shows the results using another cosine modulated filterbank (different cosine phase but same window) for the reconstruction. These results show that the reconstruction method dictated by (36)–(38) are not very loose conditions, and a somewhat tight match to these conditions has to be followed for good results.

C. LOT and GenLOT

The LOT and GenLOT's are filterbanks (lapped transforms) whose filters have linear-phase (symmetric bases) [6]–[9]. The LOT can be viewed as a special case of a GenLOT [8], [9]. Let

$$\mathbf{W} = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{I}_{M/2} & \mathbf{I}_{M/2}\\ \mathbf{I}_{M/2} & -\mathbf{I}_{M/2}\end{bmatrix} \quad\text{and}\quad \mathbf{\Psi}_i = \begin{bmatrix}\mathbf{U}_i & \mathbf{0}_{M/2}\\ \mathbf{0}_{M/2} & \mathbf{V}_i\end{bmatrix}, \tag{39}$$

where $\mathbf{U}_i$ and $\mathbf{V}_i$ can be any $M/2 \times M/2$ orthogonal matrices and let

$$\mathbf{\Delta}(z) = \text{diag}\{\underbrace{1, 1, \cdots, 1}_{M/2\ 1's}, \underbrace{z^{-1}, z^{-1}, \cdots, z^{-1}}_{M/2 \text{ delays}}\}. \tag{40}$$

$$\mathbf{\Delta}^R(z) = \text{diag}\{\underbrace{z^{-1}, z^{-1}, \cdots, z^{-1}}_{M/2 \text{ delays}}, \underbrace{1, 1, \cdots, 1}_{M/2\ 1's}\}. \tag{41}$$

The paraunitary PTM of a GenLOT with N stages is given by

$$\mathbf{F}(z) = \left(\prod_{i=N-1}^{1}\mathbf{\Psi}_i\mathbf{W}\mathbf{\Delta}(z)\mathbf{W}\right)\mathbf{D}_M \tag{42}$$

where $\mathbf{D}_M$ is the M-channels DCT. Since it is a paraunitary transform

$$\mathbf{G}(z) = \left(\prod_{i=N-1}^{1}\mathbf{\Psi}_i\mathbf{W}\mathbf{\Delta}^R(z)\mathbf{W}\right)\mathbf{D}_M. \tag{43}$$

GenLOT's were meant to possess fast implementation algorithms, and for this reason $\mathbf{G}(z)$ was expressed in a product form. Furthermore, the design of GenLOT's is based on the design of the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$. This task is heavily based on optimization algorithms. Therefore, they do not have explicit formulas and analytical resampling may not be adequate. It is possible to show that the downscaled inverse GenLOT can have the same form of a GenLOT with fewer channels. This can be seen by inverting the DCT with another DCT of smaller size and discarding subbands by reducing the size of matrices $\mathbf{W}$ and $\mathbf{\Delta}(z)$. However, the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$ have to be reoptimized. Note that GenLOT's are only defined for M even

Transform	Matrix elements	Continuous function
DST	$\sqrt{\frac{2}{M}}\alpha_i \sin((2j+1)i\pi/2M)$	$\alpha_i\sqrt{\frac{2}{M}}\sin(ti\pi)$
DST-IV	$\sqrt{\frac{2}{M}}\sin((2j+1)\ (2i+1)\pi/4M)$	$\sqrt{\frac{2}{M}}\sin(t(2i+1)\pi/2M)$
DCT-IV	$\sqrt{\frac{2}{M}}\cos((2j+1)\ (2i+1)\pi/4M)$	$\sqrt{\frac{2}{M}}\cos(t(2i+1)\pi/2M)$

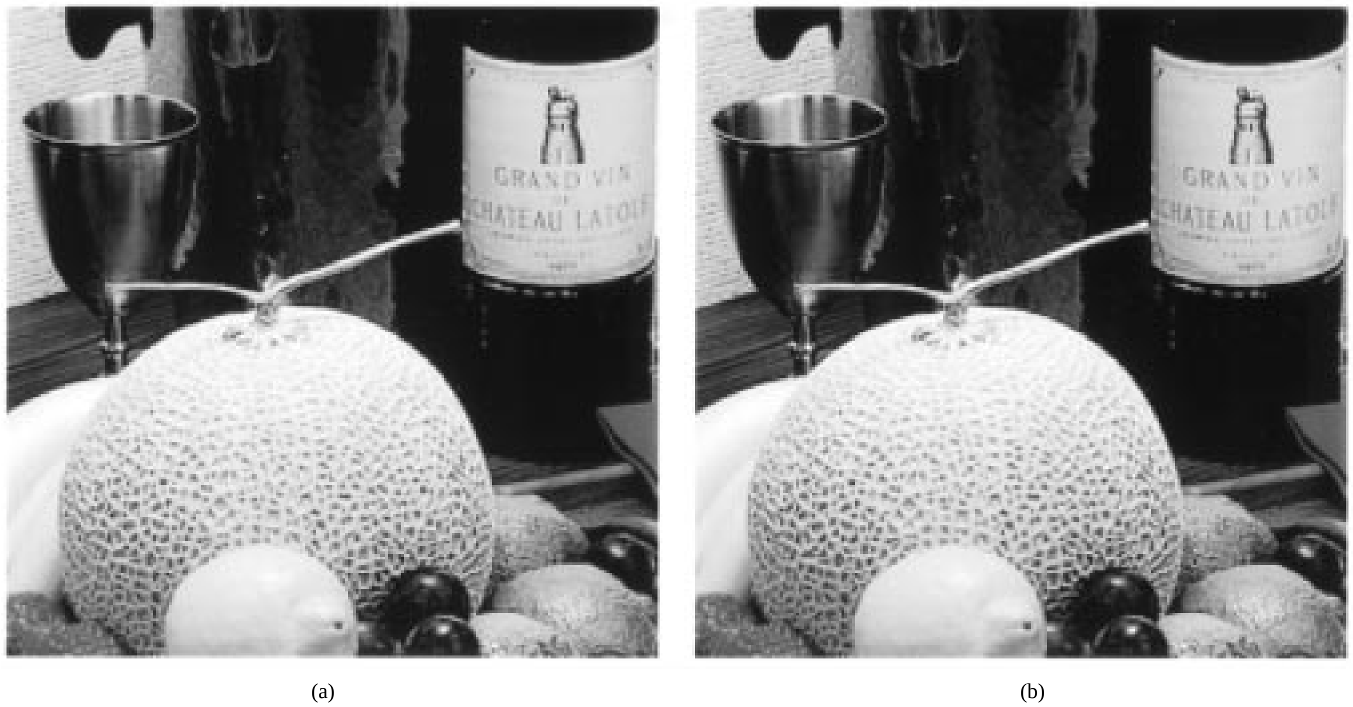


Fig. 13. Results for the fast inverse of the LOT with $M = 8$ and $K = 2$. (a) linear approximation synthesis. (b) DCT approximation.

and for $M > 2$. Thus, if $\mathbf{L}(z)$ is the PTM representing the K-channel synthesis GenLOT (with proper reoptimization of its parameters for given M-channel analysis GenLOT) we have

$$\mathbf{R}^T(z) = \sqrt{\frac{K}{M}}\mathbf{L}^T(z). \tag{44}$$

The LOT is a special case of a GenLOT with $N = 1$. For $M = 8$ the LOT has eight filters of 16 taps each, and each filter has even or odd symmetry. Malvar designed a LOT for $M = 8$ (which is particularly good for image coding) whose filter coefficients are shown at the bottom of the page. Each row of this array represents a filter and the sequence was truncated without any information loss because the filters are symmetric. Although there is no LOT for $M = 2$ we can design an inverse LOT-like two-channel filterbank that will approximate a linear resampling. The said approximation is given by

$$\begin{array}{lrrrr} 4g_0'(n) \rightarrow & 0.0068 & 0.7003 & 0.7003 & 0.0068 \\ 4g_1'(n) \rightarrow & -0.0068 & 0.7003 & -0.7003 & 0.0068. \end{array}$$

Note that such structure is similar to a four-tap quadrature mirror filter (QMF). Hence, it can be very efficiently implemented. Furthermore, the coefficients are close to those of the DCT. This is due to the concentration of large magnitude samples around the center of the LOT bases and because of the LOT, two-channels is a degenerated case. Such close similarity does not happen for $K > 2$. Fig. 13 shows reconstructed images at one quarter of the original resolution for JPEG and LOT (i.e., $M = 8$ and $K = 2$). In Fig. 13 reconstruction using the above filters and the DCT are tested. From these images, we conclude that both approximations yield reasonably good results.

VI. Conclusion

We studied methods to decompress a compressed image to a lower resolution. One can say that we actually studied downsampling in the transform domain in a framework of a downscaled inverse transform for a given forward transform. The idea is to save time and computation by not performing a full inverse transform and to avoid downsampling of a larger sized image. In order to not shadow the fact that a computationally complex downscaled transform can be more complex than a fast full transform, much of the attention was focused on fast inverse transforms.

−0.068	−0.031	0.038	0.128	0.226	0.316	0.385	0.422	0.422	⋯
−0.072	−0.086	−0.025	0.154	0.354	0.425	0.334	0.213	−0.213	⋯
0.023	0.144	0.150	−0.092	−0.370	−0.341	0.047	0.439	0.439	⋯
−0.023	0.072	0.242	−0.015	−0.404	−0.103	0.375	0.348	−0.348	⋯
0.038	0.068	−0.226	−0.031	0.385	−0.128	−0.422	0.316	0.316	⋯
0.009	0.145	−0.123	−0.099	0.334	−0.360	−0.106	0.449	−0.449	⋯
0.047	−0.092	0.023	0.150	−0.341	0.439	−0.370	0.144	0.144	⋯
0.055	−0.038	−0.028	0.120	−0.227	0.322	−0.449	0.352	−0.352	⋯

A general framework was presented allowing the use of an arbitrary lapped transform (FIR uniform filterbank) along with specific algorithms aimed at popular block and lapped transforms. As results have shown, the solutions we propose yield higher quality of the decompressed image compared to other simple reconstruction methods. Furthermore, it also leads to faster implementation. The basic idea is to *resample the synthesis filters instead of resampling the image*.

The real image exists at a higher resolution, while its downscaled version does not. Therefore, we cannot compare our results to any other result objectively. Even the operation of explicitly filtering and downsampling the image in space domain may be largely dependent on the choice of filter. The most reasonable criteria to ascertain the reconstruction quality is subjective, based on visual inspection. For this reason, instead of elaborating complex algorithms for the design of the resampling matrix $\mathbf{\Phi}$, which would optimize the reconstruction in any ad hoc sense, we concentrated on simple and fast approaches. As a final note, it is worth mention that all tests were also carried out for different compression ratios. The relative results, comparing the several methods, do not change significantly.

REFERENCES

[1] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[2] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*. Bellingham, WA: SPIE, 1991.

[3] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[5] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic, 1990.

[6] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech, 1992.

[7] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.

[8] R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, "The generalized lapped orthogonal transforms," *Electron. Lett.*, vol. 30, pp. 107–108, Jan. 1994.

[9] ——, "GenLOT: Generalized linear-phase lapped orthogonal transforms," *IEEE Trans. Signal Processing*, vol. 44, pp. 497–507, Apr. 1996.

[10] H. S. Malvar, "Extended lapped transforms: Properties, applications and fast algorithms," *IEEE Trans. Signal Processing*, vol. 40, pp. 2703–2714, Nov. 1992.

[11] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[12] K. Nayebi, T. P. Barnwell, and M. J. T. Smith, "Low delay QMF banks: Design and evaluation," *IEEE Trans. Signal Processing*, vol. 42, pp. 24–31, Jan. 1994.

[13] W. H. Press *et al*., *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[14] R. L. Burden, J. D. Faires, and A. C. Reynolds, *Numer. Anal.*, 2nd ed. Boston, MA: Prindle, Weber and Schmidt, 1981.

[15] J. M. Adant *et al*., "Block operations in digital signal processing with applications to TV coding," *Signal Processing*, vol. 13, pp. 285–397, Dec. 1987.

[16] K. N. Ngan, "Experiments on 2D decimation in time and orthogonal transform domains," *Signal Processing*, vol. 11, pp. 249–263, Oct. 1986.

[17] S. Martucci, "Image resizing in the DCT domain," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, 1995, vol. II, pp. 244–247.

[18] B. Natarajan and B. Vasudev, "A fast approximate algorithm for scaling down digital images in the DCT domain," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, 1995, vol. II, pp. 241–243.

Ricardo L. de Queiroz (S'92–M'95) received the B.S. degree from Universidade de Brasilia, Brazil, in 1987, the M.S. degree from Universidade Estadual de Campinas, Brazil, in 1990, and the Ph.D. degree from the University of Texas, Arlington, in 1994, all in electrical engineering.

From 1990 to 1991, he was with the DSP research group at Universidade de Brasilia as a Research Associate. In 1993, he received the Academic Excellence Award from the Electrical Engineering Department of the University of Texas, and in 1994 he was a Teaching Assistant at the same university. He joined Xerox Corporation, Webster, NY, in August 1994, where he is currently a Member of the Research Staff at the Color and Digital Imaging Systems Laboratory Group. His research interests are multirate signal processing, filterbanks, image and signal compression, color imaging, and processing of compressed images.

Reiner Eschbach received the M.S. and Ph.D. degrees in physics from the University of Essen, Germany, in 1983 and 1986, respectively.

He joined Xerox Corporation in 1988, where he became a Principal Scientist at the Xerox Digital Imaging Technology Center, Webster, NY, in 1994. His research interests include color image processing, digital halftoning, and compression. He is currently serving as Secretary on the Board of Directors of the Society for Imaging Science and Technology.