Transform Image Coding

What is exploited: Most of the image energy is concentrated in a small number of coefficients for some transforms

- the more energy compaction, the better

Some considerations:

- energy compaction in a small number of coefficients
- computational aspect: important (subimage by sub-image coding — 8×8 - 16×16)
- transform should be invertible
Examples of Transforms

1. Karhunen-Loeve Transform

\[ F_k (k_1, k_2) = \sum_{n=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) \cdot A(n_1, n_2; k_1, k_2) \]

\[ \lambda(k_1, k_2) \cdot A(n_1, n_2; k_1, k_2) = \]

\[ \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} K_f (n_1, n_2; l_1, l_2) \cdot A(l_1, l_2; k_1, k_2) \]

Covariance \[ K_f (n_1, n_2; l_1, l_2) = \]

\[ \mathbb{E} \left[ (x(n_1, n_2) - \bar{x}(n_1, n_2)) \cdot (x(l_1, l_2) - \bar{x}(l_1, l_2)) \right] \]

Comments:

- optimal in the sense that the coefficients are uncorrelated
- finding \( K_f (n_1, n_2; l_1, l_2) \) is hard
- no simple computational algorithm
- seldom used in practice
- On average, first \( M \) coefficients have more energy than any other transform

KL is best among all linear Transforms from: (a) Compaition (b) de-correlation.
Fig. 5.3.7 Images used for coding and statistics. (a) "Karen" has much more stationary statistics than (b) "Stripes."

Chapter 5, Basic Compression Techniques

KL T basis vectors for the image \(f\) with \(N=16\) and \(\mu = 0\). For each \(m\) the orthonormalized eigenvectors of \(R\), i.e.,

\[R_{m} = \lambda_{m}t \]

\[t_{m}t_{n} = \delta_{mn}\]

where the eigenvalues \(\lambda_{m}\) are nonnegative.

For example, Fig. 5.3.8 shows KL "Karen" in Fig. 5.3.7a using one-dimensional \(\mu = 0\). The eigenvectors are arranged according to increasing frequency, i.e., basis vector. Fig. 5.3.9 shows similar correlation model defined in Chapter 5.
size, the higher the energy compaction achieved by the transform. Also two-dimensional blocks achieve more compaction than one-dimensional blocks. Experience has shown that over a wide range of pictures there is not much improvement in average energy compaction for two dimensional block sizes above 8x8 pels. However, individual pictures with higher nonstationary statistics can always be found for which this rule of thumb is violated (for example, compare the KLT curves of Fig. 5.3.17 and Fig.5.3.19). Also, considerable correlation may remain between blocks, even though the correlation between pels within a block is largely removed.\[5.3.21\] We shall return to this point in a later section.

### 5.3.1f Miscellaneous Transforms

Several other transforms have been studied. For example, the Haar Transform\[5.3.8\] can be computed from an orthogonal (but not orthonormal) matrix $T$ that contains only $+1$'s, $-1$'s and zeros as shown in Fig. 5.3.23. This enables rather simple and speedy calculation, but at the expense of energy compaction performance.
constructed from a two-dimensional transform: the L-pel rows end-to-end, not only high adjacent pel
els with separation L. Thus the transform coded in this fasion the largest MSV are
larger not only at low L, etc. Fig. 5.3.19 shows the transform coefficients. For the separable DCT
results of separable picture “Karen” when only LT was derived from the mor
of pels. We see that the separable KLT. This is not adapt very well to all block size of 1×L pels.
5.3.19 by the separable KLT. This IS not adapt very well to all block size of 1×L pels.
transform coefficients. For
the image “Karen”. The coefficients having the largest MSV are transmitted.
NR results for p = 60% proves as the block size
small for block sizes
(a) 4×4 blocks, N=16. (b) 16×16 blocks, N=256.

Fig. 5.3.21 Comparison of truncation errors using separable, two-dimensional blocks with the image “Karen”. The coefficients having the largest MSV are transmitted.
Discrete Cosine Transform

\[ f(n, m) \rightarrow \hat{f}(n, m) = \frac{f(n, m) + f(n, -m) + f(-n, m) + f(-n, -m)}{4} \]

Comments:

- good energy compaction (better than DFT)
- fast algorithms
- all real coefficients
- most often used in practice (good quality image at bit rate less than 1 bit/pixel)
- other transforms: Hadamard, Haar, Slant, Sine, ...
The sequence \( Y(k) \) is related to \( y(n) \) through the 2N-point inverse DFT relation given by

\[
y(n) = \frac{1}{2N} \sum_{k=0}^{2N-1} Y(k) W_{2N}^{k n}, \quad 0 \leq n \leq 2N - 1.
\]  

(3.26)

From (3.20), \( x(n) \) can be recovered from \( y(n) \) by

\[
x(n) = \begin{cases} 
  y(n), & 0 \leq n \leq N - 1 \\
  0, & \text{otherwise}
\end{cases}
\]  

(3.25)

From (3.27), (3.28), and (3.29), and after some algebra,

\[
x(n) = \left\{ \begin{array}{ll} 
  \frac{1}{N} \left[ \frac{C_s(0)}{2} + \sum_{k=1}^{N-1} C_s(k) \cos \frac{\pi}{2N} k(2n + 1) \right], & 0 \leq n \leq N - 1 \\
  0, & \text{otherwise}
\end{array} \right.
\]  

(3.30)

Equation (3.30) can also be expressed as

\[
x(n) = \left\{ \begin{array}{ll} 
  \frac{1}{N} \sum_{k=0}^{N-1} w(k) C_s(k) \cos \frac{\pi}{2N} k(2n + 1), & 0 \leq n \leq N - 1 \\
  0, & \text{otherwise}
\end{array} \right.
\]  

(3.31a)

where

\[
w(k) = \begin{cases} 
  \frac{1}{2}, & k = 0 \\
  1, & 1 \leq k \leq N - 1
\end{cases}
\]  

(3.31b)

Equation (3.31) is the inverse DCT relation. From (3.25) and (3.31),

\[
C_s(k) = \left\{ \begin{array}{ll} 
  \sum_{n=0}^{N-1} 2x(n) \cos \frac{\pi}{2N} k(2n + 1), & 0 \leq k \leq N - 1 \\
  0, & \text{otherwise}
\end{array} \right.
\]  

(3.32a)

\[
x(n) = \left\{ \begin{array}{ll} 
  \frac{1}{N} \sum_{k=0}^{N-1} w(k) C_s(k) \cos \frac{\pi}{2N} k(2n + 1), & 0 \leq n \leq N - 1 \\
  0, & \text{otherwise}
\end{array} \right.
\]  

(3.32b)

From the derivation of the DCT pair, the DCT and inverse DCT can be computed by

**Computation of Discrete Cosine Transform**

**Step 1.** \( y(n) = x(n) + x(2N - 1 - n) \)

**Step 2.** \( Y(k) = \text{DFT}[y(n)] \) (2N-point DFT computation)

**Step 3.** \( C_s(k) = \left\{ \begin{array}{ll} 
  W_{2N}^{k 0} Y(k), & 0 \leq k \leq N - 1 \\
  0, & \text{otherwise}
\end{array} \right. \)
Computation of Inverse Discrete Cosine Transform

**Step 1.** \( Y(k) = \begin{cases} W_{2N}^{-k^2}C_x(k), & 0 \leq k \leq N - 1 \\ 0, & k = N \\ -W_{2N}^{-k^2}C_x(2N - k), & N + 1 \leq k \leq 2N - 1 \end{cases} \)

**Step 2.** \( y(n) = \text{IDFT} \{ Y(k) \} \) (2N-point inverse DFT computation)

**Step 3.** \( x(n) = \begin{cases} y(n), & 0 \leq n \leq N - 1 \\ 0, & \text{otherwise} \end{cases} \)

In computing the DCT and inverse DCT, Steps 1 and 3 are computationally quite simple. Most of the computations are in Step 2, where a 2N-point DFT is computed for the DCT and a 2N-point inverse DFT is computed for the inverse DCT. The DFT and inverse DFT can be computed by using fast Fourier transform (FFT) algorithms. In addition, because \( y(n) \) has symmetry, the 2N-point DFT and inverse DFT can be computed (see Problem 3.20) by computing the N-point DFT and the N-point inverse DFT of an N-point sequence. Therefore, the computation involved in using the DCT is essentially the same as that involved in using the DFT.

In the derivation of the DCT pair, we have used an intermediate sequence \( y(n) \) that has symmetry and whose length is even. The DCT we derived is thus called an even symmetrical DCT. It is also possible to derive the odd symmetrical DCT pair in the same manner. In the odd symmetrical DCT, the intermediate sequence \( y(n) \) used has symmetry, but its length is odd. For the sequence \( x(n) \) shown in Figure 3.9(a), the sequence \( y(n) \) used is shown in Figure 3.9(b). The length of \( y(n) \) is \( 2N - 1 \), and \( \hat{y}(n) \), obtained by repeating \( y(n) \) every \( 2N - 1 \) points, has no artificial discontinuities. The detailed derivation of the odd symmetrical DCT is considered in Problem 3.22. The even symmetrical DCT is more commonly used, since the odd symmetrical DCT involves computing an odd-length DFT, which is not very convenient when one is using FFT algorithms.

![Figure 3.9](image-url) Example of (a) \( x(n) \) and (b) \( y(n) = x(n) + x(2N - 2 - n) - x(N - 1)\delta(n - (N - 1)) \). The sequence \( y(n) \) is used in the intermediate step in defining the odd symmetrical discrete cosine transform of \( x(n) \).
D C T

- Signal independent

- $\rho \rightarrow 1 : \text{KLT} \rightarrow \text{DCT}$
  for first order
  Markov Image model.

- Type II DCT:

$$S(K_1, K_2) = \sqrt{\frac{4}{N^2}} C(K_1)C(K_2)$$

$$\sum_{n_1 = 0}^{N-1} \sum_{n_2 = 0}^{N-1} s(n_1, n_2) \cos\left(\frac{\pi 2(n_1 + 1)K_1}{2N}\right) \cos\left(\frac{\pi 2(n_2 + 1)K_2}{2N}\right)$$

$$C(K) = \begin{cases} 
\frac{1}{\sqrt{2}} & K = 0 \\
1 & \text{otherwise}
\end{cases}$$
The outer products of the rows are shown in Figure 12.4. Notice that the basis matrices show increased variation as we go from the top left matrix, corresponding to the \( b_{00} \) coefficient, to the bottom right matrix, corresponding to the \( b_{(N-1)(N-1)} \) coefficient.

The DCT is closely related to the discrete Fourier transform (DFT) mentioned in Chapter 11, and in fact can be obtained from the DFT. However, in terms of compression, the DCT performs better than the DFT.

Recall that when we find the Fourier coefficients for a sequence of length \( N \), we assume that the sequence is periodic with period \( N \). If the original sequence is as shown in Figure 12.5a, the DFT assumes that the sequence outside the interval of interest behaves in the manner shown in Figure 12.5b. This introduces sharp discontinuities, at the beginning and end of the sequence. In order to represent these sharp discontinuities the DFT needs nonzero coefficients for the high-frequency components. As these components are needed only at the two endpoints of the sequence, their effect needs to be cancelled out at other points in the sequence. Thus, the DFT adjusts other coefficients accordingly. When we discard the high-frequency coefficients (which should not have been there anyway) during the compression process, the coefficients that were cancelling out the high-frequency effect in other parts of the sequence result in the introduction of additional distortion.
Discarding Transform Coefficients (cont.)

Threshold coding: Coefficients with values above a given threshold are coded

- location as well as amplitude has to be coded
- run-length coding is useful (many zeroes)

![Bit allocation map example](image)

Figure 10.44 Example of bit allocation map at 1 bit/pixel for zonal discrete cosine transform image coding. Block size = 16 x 16 pixels.
Zonal coding: Eliminate coefficients in a fixed zone

(a)

(b)

# Bits for coefficient i with variance $6_i^2$

$$b_i = \frac{B}{M} + \frac{1}{2} \log_2 6_i^2 - \frac{1}{2} M \sum_{i=1}^{M} \log_2 6_i^2$$

$M = \# \text{ of coefficients to be coded}$

$B = \text{Total \# of bits}$
Scalar Quantization of a Vector Source

- Assume $N$ scalars: $f_i \quad 1 \leq i \leq N$
- Each $f_i$ is quantized to $L_i$ reconstruction levels.
- Total of $B$ bits to code $N$ scalars.
- Optimum bit allocation strategy depends on (a) error criterion and (b) pdf of each random variable.
- Assume we minimize MSE: $\Sigma_{i=1}^{N} E[(f_i - f_i)^2]$ with respect to $B_i$ the number of bits for the $ith$ scalar for $1 \leq i \leq N$.
- Assume pdf of all $f_i$ is the same except they have different variances.
- Use Lloyd Max quantizer.
- Under these conditions we have:
  \[ B_i = \frac{B}{N} + \frac{1}{2} \log \frac{\sigma_i^2}{\left[ \prod_{j=1}^{N} \sigma_j^2 \right]^{1/N}} \]
  - $\sigma_i^2$ is the variance of $f_i$
  - $L_i = \frac{\sigma_i}{\left[ \prod_{j=1}^{N} \sigma_j \right]^{1/N} 2^{B/N}}$
  - $L_i$ is the number of reconstruction levels for source $i$
and (b) show the results of DCT image coding at 1 bit/pixel and ½ bit/pixel, respectively. The original image is the 512 × 512-pixel image shown in Figure 10.45(a). In both examples, the subimage size used is 16 × 16 pixels, and adaptive zonal coding with the zone shape shown in Figure 10.43(b) and the zone size adapted to the local image characteristics has been used.

Figure 10.47  DCT-coded image with visible blocking effect.

Figure 10.48  Example of DCT image coding. (a) DCT-coded image at 1 bit/pixel. NMSE = 0.87, SNR = 20.7 dB. (b) DCT-coded image at ½ bit/pixel. NMSE = 0.97, SNR = 20.2 dB.
and are selected from a zone of triangular shape shown in Figure 10.43(a). From Figure 10.45, it is clear that the reconstructed image appears more blurry as we retain a smaller number of coefficients. It is also clear that an image reconstructed from only a small fraction of the transform coefficients looks quite good, illustrating the energy compaction property.

Another type of degradation results from quantization of the retained transform coefficients. The degradation in this case typically appears as graininess in the image. Figure 10.46 shows the result of coarse quantization of transform coefficients. This example is obtained by using a 2-bit uniform quantizer for each retained coefficient to reconstruct the image in Figure 10.45(b).

A third type of degradation arises from subimage-by-subimage coding. Since each subimage is coded independently, the pixels at the subimage boundaries may have artificial intensity discontinuities. This is known as the blocking effect, and is more pronounced as the bit rate decreases. An image with a visible blocking effect is shown in Figure 10.47. A DCT with zonal coding, a subimage of 16 × 16 pixels, and a bit rate of 0.15 bit/pixel were used to generate the image in Figure 10.47.

Examples. To design a transform coder at a given bit rate, different types of image degradation due to quantization have to be carefully balanced by a proper choice of various design parameters. As was discussed, these parameters include the transform used, subimage size, selection of which coefficients will be retained, bit allocation, and selection of quantization levels. If one type of degradation dominates other types of degradation, the performance of the coder can usually be improved by decreasing the dominant degradation at the expense of some increase in other types of degradation.

Figure 10.48 shows examples of transform image coding. Figure 10.48(a)
the blocking effect at
ansforms called lapped
used are overlapped.
nains the same as the
representing a subimage
subimage size. Even
f coefficients nly from the transform
uces the blocking
ed, the total number of
reconstruction remains
filter the image at the
roach, the coding pro-
ually exclusive. The
ised by segmentation
ency components. In
bimage boundaries to
procedure used at the
image discontinuities
an does not increasing
meth was shown
DCT in reducing the
w the filtering method
apped orthogonal
ample of reduction
.50(a) shows an image
processed by applying
ly the pixels at the
rate applications, while
. A hybrid transform
aveform and transform-
coding methods at
true 2-D transform
by a 1-D transform, such
 is then coded
(or row). This is illus-
each row of data well.
Due to the transform
ave been by wavefor-
ten issues such as

in the blocking effect at
transforms called lapped
used are overlapped.

Figure 10.50 Example of blocking effect reduction using a filtering method. (a) Image
of 512 × 512 pixels with visible blocking effect. The image is coded by a zonal DCT coder
at 0.2 bit/pixel. (b) Image in (a) filtered to reduce the blocking effect. The filter used is a
3 × 3-point \( h(n_1, n_2) \) with \( h(0, 0) = \frac{1}{2} \) and \( h(n_1, n_2) = \frac{1}{8} \) at the remaining eight points.

Figure 10.51 Hybrid transform/waveform coder.

10.4.5 Adaptive Coding and Vector Quantization

Transform coding techniques can be made adaptive to the local characteristics
within each subimage. In zonal coding, for example, the shape and size of the

Selection of the zone shape and size in zonal coding are simpler than those with a
2-D transform coder. Hybrid coding of a single image frame has not been used
extensively in practice, perhaps because the method does not reduce the correlation
in the data as much as a 2-D transform coder and the complexity in a 2-D transform
coder implementation is not much higher than a hybrid coder. As will be discussed
in Section 10.6, however, hybrid coding is useful in interframe image coding.
Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding

Ruth Rosenholtz and Avideh Zakhor

Abstract—We propose a new iterative block reduction technique based on the theory of projection onto convex sets. The basic idea behind this technique is to impose a number of constraints on the coded image in such a way as to restore it to its original artifact-free form. One such constraint can be derived by exploiting the fact that the transform-coded image suffering from blocking effects contains high-frequency vertical and horizontal artifacts corresponding to vertical and horizontal discontinuities across boundaries of neighboring blocks. Since these components are missing in the original uncoded image, or at least can be guaranteed to be missing from the original image prior to coding, one step of our iterative procedure consists of projecting the coded image onto the set of signals that are bandlimited in the horizontal or vertical directions. Another constraint we have chosen in the restoration process has to do with the quantization intervals of the transform coefficients. Specifically, the decision levels associated with transform coefficient quantizers can be used as lower and upper bounds on transform coefficients, which in turn define boundaries of the convex set for projection. Thus, in projecting the “out-of-bound” transform coefficient onto this convex set, we will choose the upper (lower) bound of the quantization interval if its value is greater (less) than the upper (lower) bound. We present a few examples of our proposed approach.

I. INTRODUCTION

Transform coding is one of the most widely used image compression techniques. It is based on dividing an image into small blocks, taking the transform of each block and discarding high-frequency coefficients and quantizing low-frequency coefficients. Among various transforms, the discrete cosine transform (DCT) is one of the most popular because its performance for certain class of images is close to that of the Karhunen–Loève transform (KLT), which is known to be optimal in the mean squared error sense.

Although DCT is used in most of today’s standards such as JPEG and MPEG, its main drawback is what is usually referred to as the “blocking effect.” Dividing the image into blocks prior to coding causes blocking effects—discontinuities between adjacent blocks—particularly at low bit rates. In this paper, we present an iterative technique for the reduction of blocking effects in coded images.

II. ITERATIVE RESTORATION METHOD

The block diagram of our proposed iterative approach is shown in Fig. 1. The basic idea behind our technique is to impose a number
of constraints on the coded image in such a way as to restore it to its original artifact-free-form. We derive one such constraint from the fact that the coded image with $N \times N$ blocks has high-frequency horizontal and vertical artifacts corresponding to the discontinuities at the edges of the $N \times N$ blocks. Therefore, one step of our procedure consists of bandlimiting the image in the horizontal and vertical directions. We refer to this constraint as the filtering constraint.

We derive the second constraint from the quantizer and thus refer to it as the quantization constraint. Because the quantization intervals for each DCT coefficient is assumed to be known in decoding a DCT encoded image, the quantization constraint ensures that in restoring images with blocking effects, DCT coefficients of $N \times N$ blocks remain in their original quantization interval.

If $S_1$ denotes the set of bandlimited images, and $S_2$ denotes the set of images whose $N \times N$ DCT coefficients lie in specific quantization intervals, our goal can be stated as that of finding an image in the intersection of $S_1$ and $S_2$. One way to achieve this is to start with an arbitrary element in either of the two sets and iteratively map it back and forth to the other set, until the process converges to an element in the intersection of the two sets. Under these conditions convergence can be guaranteed by the theory of projection onto convex sets (POCS) if sets $S_1$ and $S_2$ are convex, and if the mapping from each set to the other is a projection [6]. By definition, the projection of an element $x$ in set $A$ onto set $B$ is equivalent to finding the closest element, according to some metric, in $B$ to $x$.

To apply the above idea to our problem, we first notice that two sets $S_1$ and $S_2$ are both convex. We also choose the mean squared error as our metric of closeness. This implies that a projection from $S_2$ to $S_1$ can be accomplished by any bandlimitation algorithm such as ideal low-pass filtering. It also implies that projection from $S_1$ to $S_2$ can be accomplished by moving $N \times N$ DCT coefficients that are outside their designated quantization interval to the closest boundary of their respective quantization intervals. For instance, if a particular $N \times N$ DCT coefficient, which is supposed to be in the range $[a, b]$, takes on a value greater than $b$, it is projected to $b$. Alternatively, if it takes on a value smaller than $a$ it is projected onto $a$.

Having explained the constraints, convex sets, and projections, we now summarize our proposed iterative procedure shown in Fig. 1. In the first part of each iteration, we low pass filter, or bandlimit, the image that has high-frequency horizontal and vertical components corresponding to the discontinuities between $N \times N$ blocks. In the second part of each iteration we apply the quantization constraint as follows. First we divide the image into $N \times N$ blocks and take the DCT of each. Then we project any coefficient outside its quantization range onto its appropriate value. Under these conditions, the POCS theory guarantees that iterative projection between the sets $S_1$ and $S_2$ results in convergence to an element in the intersection of the two sets.

III. EXPERIMENTAL RESULTS

Fig. 2(a) shows the original, unquantized $512 \times 512$ Lena, and (b), (c), and (d) show its JPEG encoded version to 0.43, and 0.24,
Fig. 2(a) Original 512 x 512 image, Lena. 2(b) Lena quantized to 0.43 bpp. 2(c) Lena quantized to 0.24 bpp. 2(d) Lena quantized to 0.15 bpp. The quantization tables for Figs. 2(b), (c), and (d) are included in the Appendix.

Strictly speaking, the band-limitation portion of our algorithm corresponds to a true projection if the image under consideration is convolved with an ideal low-pass filter. Since an ideal low-pass filter cannot be implemented in practice, we have chosen to approximate it with a 3 x 3 finite impulse response (FIR) filter of the form

\[
\begin{align*}
 h(0, 0) &= 0.2042, \\
 h(0, 1) &= h(0, -1) = h(1, 0) = h(-1, 0) = 0.1239, \\
 h(0, 2) &= h(0, -2) = h(2, 0) = h(2, 0) = 0.0751.
\end{align*}
\]

We now show examples of our iterative algorithm. Fig. 3(a) shows five iterations of our algorithm applied to the 0.43-bpp quantized image of Fig. 2(b). The FIR filter of (1) was used for the band-limitation step. As Fig. 2(b) shows, blocking artifact has been removed without introducing excessive blurring. For comparison purposes, the result of applying the low-pass filter in (1) to Fig. 2(b) for five times, without applying the quantization constraint, is also shown in Fig. 3(b). Although consecutive low-pass filtering removes most of the blocking effect, it blurs the image in a noticeable way. We have found that applying the low-pass filter of (1) once rather than five times, results in a less blurry image than in Fig. 3(b), but at the same time does not remove all the blocking effect.

Figs. 4(a) and (b) show application of our algorithm to the 0.24-bpp quantized image of Fig. 2(c) for 5 and 20 iterations, respectively. The FIR filter of (1) was used for the band-limitation step. As seen, the blocking artifact is better removed in Fig. 4(b) than in 4(a), while they are as sharp as each other. For comparison purposes, Fig. 4(c) and (d) show the result of applying the low-pass filter of (1) to Fig. 2(c), 5 and 20 times, respectively. Comparing Fig. 4(c) and 4(d) to Fig. 4(a) and (b), respectively, we find that the latter pair are more blurry than the former. Thus, applying the quantization constraint prevents the images from becoming excessively blurry.

Fig. 5(a) shows application of our algorithm to the 0.15-bpp quantized image of Fig. 2(d) for 20 iterations. The FIR filter of (1) was used for the band-limitation step. For comparison purposes, Fig. 5(b) shows the result of applying the low-pass filter of (1) to Fig. 2(d), 20 times. Comparing Fig. 5(b) to 5(a), we find that the latter is considerably more blurry than the former.

**IV. CONCLUSIONS**

The major conclusions to be drawn from this paper are as follows: 1) the proposed iterative algorithm using a 3 x 3 low-pass filtering of (1) results in images that are free of blocking artifacts and excessive blurring; 2) low-pass filtering by itself could remove blockiness but at the expense of increased blurriness.

It is conceivable to generate images similar to Figs. 5(a) and 4(b) without having to apply our algorithm for as many as 20 iterations. Our conjecture is that this could be achieved by increasing the region of support of the impulse response of the filter of (1). In practical hardware implementations however, 3 x 3 convolvers are more readily available than, say, 30 x 30 ones.

We have checked the convergence of our algorithm and found that it converges after 20 iterations or so. This is encouraging since there is no guarantee that the intersections of our particular convex sets is nonempty, and the theory of POCS only guarantees convergence in situations where the intersection is nonempty.

One way to increase the likelihood of convergence is to vary the confidence with which the ideal solution is in the chosen constraint set, by varying its size. For example, if we choose prototype constraint sets as in [10], using the statistics of the
Fig. 3(a) Result of applying the iterative algorithm to Fig. 2(b) for five iterations with the low-pass filter of (1) used for bandlimitation. (b) Result of low-pass filtering Fig. 2(b) five times using the filter in (1).

quantization noise, we can change the boundaries and the size of the constraint set in a controlled fashion and therefore increase the likelihood of a solution in the intersection of the constraint sets. Examples of such prototype constraint sets include bounded variation from the Weiner solution and pointwise adaptive smoothness. The latter constraint has the obvious advantage of being locally adaptive to changes in the characteristics of the image. Projection onto fuzzy sets is another way of increasing the size of our convex sets [9].

APPENDIX

The quantization table for Fig. 2(b) is

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>24</td>
<td>28</td>
<td>32</td>
<td>36</td>
<td>80</td>
<td>98</td>
<td>144</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>28</td>
<td>34</td>
<td>52</td>
<td>70</td>
<td>128</td>
<td>184</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>32</td>
<td>48</td>
<td>74</td>
<td>114</td>
<td>156</td>
<td>190</td>
</tr>
<tr>
<td>32</td>
<td>34</td>
<td>48</td>
<td>58</td>
<td>112</td>
<td>128</td>
<td>174</td>
<td>196</td>
</tr>
<tr>
<td>36</td>
<td>52</td>
<td>74</td>
<td>112</td>
<td>136</td>
<td>162</td>
<td>206</td>
<td>224</td>
</tr>
<tr>
<td>80</td>
<td>70</td>
<td>114</td>
<td>128</td>
<td>162</td>
<td>208</td>
<td>242</td>
<td>200</td>
</tr>
<tr>
<td>98</td>
<td>128</td>
<td>156</td>
<td>174</td>
<td>206</td>
<td>242</td>
<td>240</td>
<td>206</td>
</tr>
<tr>
<td>144</td>
<td>184</td>
<td>190</td>
<td>196</td>
<td>224</td>
<td>200</td>
<td>206</td>
<td>208</td>
</tr>
</tbody>
</table>

For Fig. 2(c) it is

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>60</td>
<td>70</td>
<td>70</td>
<td>90</td>
<td>120</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>70</td>
<td>96</td>
<td>130</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>80</td>
<td>120</td>
<td>200</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>70</td>
<td>96</td>
<td>120</td>
<td>145</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
</tbody>
</table>

Fig. 4(a) Result of applying the iterative algorithm to Fig. 2(c) for 5 iterations with the low-pass filter of (1) used for bandlimitation. (b) Result of applying the iterative algorithm to Fig. 2(c) for 20 iterations with the low-pass filter of (1) used for bandlimitation. (c) Result of low-pass filtering Fig. 2(c) five times using the filter in (1). (d) Result of low-pass filtering Fig. 2(c) 20 times using the filter in (1).
Fig. 5(a) Result of applying the iterative algorithm to Fig. 2(d) for 20 iterations with the low-pass filter of (1) used for bandlimitation. (b) Result of low pass filtering Fig. 2(d) 20 times using the filter in (1).

The 255 entry in the above tables indicates that the coefficient was discarded.

REFERENCES