Removal of Compression Artifacts Using Projections onto Convex Sets and Line Process Modeling

Yongyi Yang, *Member, IEEE,* and Nikolas P. Galatsanos, *Senior Member, IEEE*

Abstract— **In this paper, we present a new image recovery algorithm to remove, in addition to blocking, ringing artifacts from compressed images and video. This new algorithm is based on the theory of projections onto convex sets (POCS). A new family of directional smoothness constraint sets is defined based on line processes modeling of the image edge structure. The definition of these smoothness sets also takes into account the fact that the visibility of compression artifacts in an image is spatially varying. To overcome the numerical difficulty in computing the projections onto these sets, a divide-and-conquer (DAC) strategy is introduced. According to this strategy, new smoothness sets are derived such that their projections are easier to compute. The effectiveness of the proposed algorithm is demonstrated through numerical experiments using Motion Picture Expert Group based (MPEG-based) coders-decoders (codecs).**

Index Terms— **Compression artifacts, postprocessing, projections onto convex sets.**

I. Introduction

At the present time, transform-based compression is among the most popular, and is widely used for both still images and video. In order to achieve high compression ratio, part of the informtion about the transform coefficients of the original image is discarded in the encoder. Quantization, in various forms, is the most widely used approach for this purpose [5]. Thus, in the decoder the original image or video cannot be exactly reconstructed. Nevertheless, in a traditional decoder, including Joint Photographers Expert Group (JPEG) and Motion Pictures Expert Group (MPEG) [6], [12], the decoded images are obtained by simply taking the inverse transform of the received quantized transform coefficients. As a result, in low bit rate applications, the compressed images and video exhibit annoying compression artifacts.

"Blocking" artifacts—a direct result of independent block processing in codecs such as JPEG and MPEG—have perhaps received the most attention in the literature. Indeed, various attempts have been made in the past to develop postprocessing algorithms to improve the quality of block-transform compressed images. In [9], [14], [16], and [18], different filtering algorithms are used to remove blocking artifacts in the decoded image. In [11], [17], [19], [20], and [22], image recovery approaches are proposed for decoding high-quality images from compressed image data. In [23] a spatially adaptive recovery algorithm, which takes into account the local visibility of blocking artifacts, was introduced.

Apart from blocking, ringing is another annoying artifact that appears frequently in transform-based compressed images. However, it seems that this artifact has not received as much attention in the literature. Ringing artifacts typically appear as sharp oscillations or "ghost shadows" along the edges in compressed images (see top of Figs. 3 and 4). Ringing artifacts are especially objectionable in compressed video when they occur around an object that is moving relative to a flat background. In the video compression community, these artifacts are also known as *mosquito* or *corona* artifacts, depending on their appearance [7].

In this paper, we present a new image recovery algorithm to remove, in addition to blocking, ringing artifacts from compressed images and video. This new algorithm is based on the theory of projections onto convex sets (POCS). A new family of directional smoothness constraint sets is defined based on line processes modeling of the image edge structure. The definition of these smoothness sets takes into account the fact that the visibility of compression artifacts in an image is spatially varying. To overcome the numerical difficulty in computing the projections onto these sets, a divide-and-conquer (DAC) strategy is introduced. Based on this strategy new smoothness sets are derived such that their projections are easier to compute.

From an image recovery point of view, this work provides a new paradigm for the use of line processes. The concept of line processes was first introduced in the study of compound stochastic processes modeling of image fields [3], [4], [13], [15]. In this study, line processes along the horizontal, vertical, and two diagonal directions are introduced to model the edge structure in an image. Then, the line process along each direction is incorporated into a smoothness constraint set so that local image activity is captured. In addition to the guaranteed convergence property of the POCS algorithm, this approach also offers implementation flexibility and computational simplicity.

To our best knowledge, line processes were used for the first time in [15] for recovering compressed images. The proposed algorithm in [15] is based on the theory of mean field annealing [3]. Since there is no rigorous proof for the convergence of the mean field annealing algorithm, the convergence properties of this algorithm cannot be rigorously established. In [15], line processes in only two directions were used and the resulting image model did not take into account the local visibility of the coding artifacts.

Manuscript received February 2, 1996; revised January 24, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Amy R. Reibman.

The authors are with the Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: npg@ece.iit.edu).

Publisher Item Identifier S 1057-7149(97)07033-4.

The rest of this paper is organized as follows. In Section II a brief review of the POCS theory and previous related work is given. In Section III the new smoothness constraint sets are defined. In Section IV the DAC strategy is introduced, and the projections onto the resulting new smoothness sets are derived. In Section V the estimation of the line processes and the visibility weights is presented. In Section VI, the POCS algorithm for recovering MPEG compressed images is outlined. In Section VII, numerical experiments are presented to demonstrate the merit of the proposed algorithm. Finally, in Section VIII conclusions and suggestions for future research are given.

II. Review of Previous Work

Since the pioneering work of Youla [26], the POCS method has been applied successfully to solve a number of image processing problems. The main result from the POCS theory that we shall use for the rest of the paper is briefly summarized as follows: Given m closed convex sets[1] $C_i, i = 1, 2, \cdots, m,$ in a Hilbert space, and $C_0 \triangleq \cap_{i=1}^{m} C_i$ nonempty, let P_i denote the projection operator onto C_i, i.e.,

$$||\boldsymbol{f} - P_i\boldsymbol{f}|| = \min_{\boldsymbol{g}\in C_i} ||\boldsymbol{f} - \boldsymbol{g}|| \tag{1}$$

and $P_i\boldsymbol{f}$ is called the projection of $\boldsymbol{f}$ onto C_i. Then, the iteration

$$\boldsymbol{f}_{k+1} = P_m P_{m-1} \cdots P_1 \boldsymbol{f}_k, \qquad k = 0, 1, 2, \cdots \tag{2}$$

will converge to a point of C_0 for an arbitrary initial point $\boldsymbol{f}_0$ [26].

The key for applying the POCS method for image recovery problems is to express every piece of available knowledge about the unknown image, which is to be recovered, by a closed convex constraint set in the image space. Then, an image vector that lies in the intersection of all the constraint sets will satisfy all the available knowledge and henceforth can be taken as a solution to the recovery problem. Clearly, such a solution can be found by invoking the POCS iterative algorithm in (2). Thus, the definition of a POCS-based image recovery algorithm requires two steps in general: 1) the definition of the closed convex constraint sets that are used; and 2) the derivation of the projections onto these constraint sets. The main advantage of this approach is that it provides flexibility in incorporating *prior* knowledge about the unknown image into the recovery algorithm. Indeed, any type of prior knowledge can be included in a POCS-based recovery algorithm as long as it can be represented by a closed convex constraint set in the image space.

In earlier studies, Rosenholtz and Zakhor [17] and Yang *et al.* [22], [23] used the POCS method to decode high-quality images from compressed image data. According to [17], [22], and [23], image decoding was treated as an image recovery problem. Two types of constraint sets were used in the recovery algorithms: 1) the constraint sets that are based on the transmitted compressed image data; and 2) the constraint sets that are based on *a priori* knowledge about the original image that is available at the decoder without extra coding cost.

Following the work in [22] and [23], a digital $M \times N$ image is treated as a vector in the Euclidean space $R^{M\cdot N}$ by lexicographic ordering either by rows or columns, where as distance measure the l_2 norm is used. Then, the block discrete-cosine transform (BDCT) [6], [12] is viewed as a linear transform on $R^{M\cdot N}$ and is denoted by B using matrix notation. In a BDCT-based image codec, the quantized BDCT coefficients of an image are available at the decoder. The knowledge of these quantized BDCT coefficients [17], [22] leads to the following constraint set:

$$\begin{aligned} C_T \triangleq \{\boldsymbol{f}\colon F_n^{\min} \le (B\boldsymbol{f})_n \le F_n^{\max},& \\ \forall n = 1, 2, \cdots, M\cdot N\}& \end{aligned} \tag{3}$$

where $F_n^{\min}$ and $F_n^{\max}$ are the end-points of the quantization interval that is associated with the received quantized level of the nth BDCT coefficient $(B\boldsymbol{f})_n$.

The constraint set C_T is closed and convex, and the projection of an image vector $\boldsymbol{f}$ onto C_T is given by

$$P_T\boldsymbol{f} = B^{-1}\boldsymbol{F} \tag{4}$$

where the nth component of $\boldsymbol{F}$ is determined by

$$F_n = \begin{cases} F_n^{\min}, & \text{if } (B\boldsymbol{f})_n < F_n^{\min} \\ F_n^{\max}, & \text{if } (B\boldsymbol{f})_n > F_n^{\max} \\ (B\boldsymbol{f})_i, & \text{otherwise} \end{cases} \tag{5}$$

for $n = 1, 2, \cdots, M\cdot N$. Note that due to the unitary nature of the BDCT transform B, its inverse B^{-1} is simply its transpose.

In addition to the data constraint set C_T, smoothness constraint sets were also introduced in [22], [23] to enforce cross-block-boundary smoothness in the decoded images. Write an $M \times N$ image $\boldsymbol{f}$ in its column vector form as

$$\boldsymbol{f} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N\} \tag{6}$$

where $\boldsymbol{f}_i$ denotes the ith column of the image. Let Q be a linear operator such that $Q\boldsymbol{f}$ gives the difference between adjacent columns at the coding block boundaries of $\boldsymbol{f}$. For example, for the case of $N = 352$ and 8×8 blocks

$$Q\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_8 & - & \boldsymbol{f}_9 \\ \boldsymbol{f}_{16} & - & \boldsymbol{f}_{17} \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ \boldsymbol{f}_{344} & - & \boldsymbol{f}_{345} \end{bmatrix}. \tag{7}$$

Then the norm of $Q\boldsymbol{f}$

$$||Q\boldsymbol{f}|| = \left[\sum_{i=1}^{43} ||\boldsymbol{f}_{8\cdot i} - \boldsymbol{f}_{8\cdot i+1}||^2\right]^{1/2} \tag{8}$$

is a measure of the total intensity variations between the image pixels at the adjacent block boundary columns. Thus, by forcing this quantity to be small, we can suppress blocking

[1] A set C is *convex* if for $\boldsymbol{x}_1, \boldsymbol{x}_2 \in C, \boldsymbol{x}_3 \triangleq \alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2$ is also in C for all $0 < \alpha < 1$.

artifacts along the vertical boundaries. This leads to the following constraint set [22]:

$$C_s \triangleq \{\boldsymbol{f}: ||Q\boldsymbol{f}|| \leq E\} \tag{9}$$

where E is a scalar upper bound that defines the size of this set. In practice, the quantity E is estimated from the received compressed image data. The set C_s is closed and convex. Its projection operator is derived in [22].

In [23], the fact that the local visibility of the coding artifacts in compressed images is spatially varying was further taken into account. For example, blocking artifacts are more visible and more annoying in a smooth region than in a texture region of an image. To exploit this fact, a visibility matrix W was introduced into the between-block-boundary variation term $Q\boldsymbol{f}$ to characterize the local visibility of the coding artifacts. This gives rise to a new smoothness constraint set

$$C_w \triangleq \{\boldsymbol{f}: ||WQ\boldsymbol{f}|| \leq E\}. \tag{10}$$

The visibility matrix W is diagonal, and its elements are determined by the local statistical properties of the image. In practice W is estimated using the compressed image data at no extra coding cost [23]. The set C_w is closed and convex, and its projection operator is derived in [23]. In a similar fashion, we can extend this to define a constraint set for the block boundary rows to enforce cross-block-boundary smoothness.

The incorporation of these smoothness sets into the POCS recovery algorithm leads to a spatially adaptive processing algorithm where image pixels are treated differently according to the local visibility properties of the coding artifacts. The resulting decoded images from this approach are demonstrated to be better than that obtained from traditional decoders, both visually and objectively using a distance metric [23].

Unfortunately, none of the smoothness constraint sets described above address explicitly the ringing artifacts. In this study, we propose a new class of smoothness constraint sets to address this issue specifically. Due to its popularity, the BDCT-based codec is considered in rest of the development. However, the proposed approach can be applied to other transform based image codecs as well.

III. Smoothness Constraint Sets on Ringing Artifacts

Since ringing artifacts mainly occur around edges in a compressed image, it is natural for a processing algorithm to process the image pixels around the image edges and leave the pixels on the edges unprocessed. Such an approach is very desirable, since it will preserve the existing edge structure in a compressed image while removing ringing artifacts. The main difficulty, however, is that the edge locations are not immediately available at the decoder. In other words, the locations of the ringing artifacts in a compressed image are not known *a priori,* unlike the blocking artifacts which are known to occur at block boundaries. This gives rise to the need for a systematic approach to describe the edge locations in an image.

In the study of compound stochastic processes modeling of image fields, the concept of *line processes* was introduced to describe the transition in the statistical model of an image field [4], [13], [21]. In this paper, line processes are used to describe the edge structure in an image. The edges in an image typically appear in regions where sharp intensity transitions occur and they exhibit different orientations. As a result, line processes along different directions are used to describe the existence of edge elements between two neighboring pixels along different directions. Due to the use of a rectangular image sampling structure, the following four directions are considered: The horizontal (or 0°), the vertical (or 90°), the positive diagonal (or $+45°$), and the negative diagonal (or $-45°$). This concept is illustrated in Fig. 1(a), where the image pixel locations are denoted by the symbols "•" while the line sites denoted by the symbols "$\times$". The 0° process is used to characterize the existence of a horizontal edge element between two neighboring pixels along the vertical direction, while the 90° process is used for the existence of a vertical edge element between two neighboring pixels along the horizontal direction. In a similar fashion, the +45° process and the −45° process are defined for the edges along the +45° and the −45° directions, respectively. To clarify this idea, Fig. 1(b) furnishes a pictorial example in which the realizations of the four line processes are shown. In this example the symbols "−," "|," "/," and "\" are used, respectively, to indicate the existence of 0°, 90°, +45°, and −45° line processes between two adjacent image pixels.

Using these line processes, we can define smoothness constraint sets to capture the smoothness properties of an image in the four directions. Let $l_v(i,j)$ denote the 90° line process at the site (i,j), i.e.,

$$l_v(i,j) \triangleq \begin{cases} 1, & \text{if these exists an edge between} \\ & \quad \text{the pixels } (i,j) \text{ and } (i,j+1) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

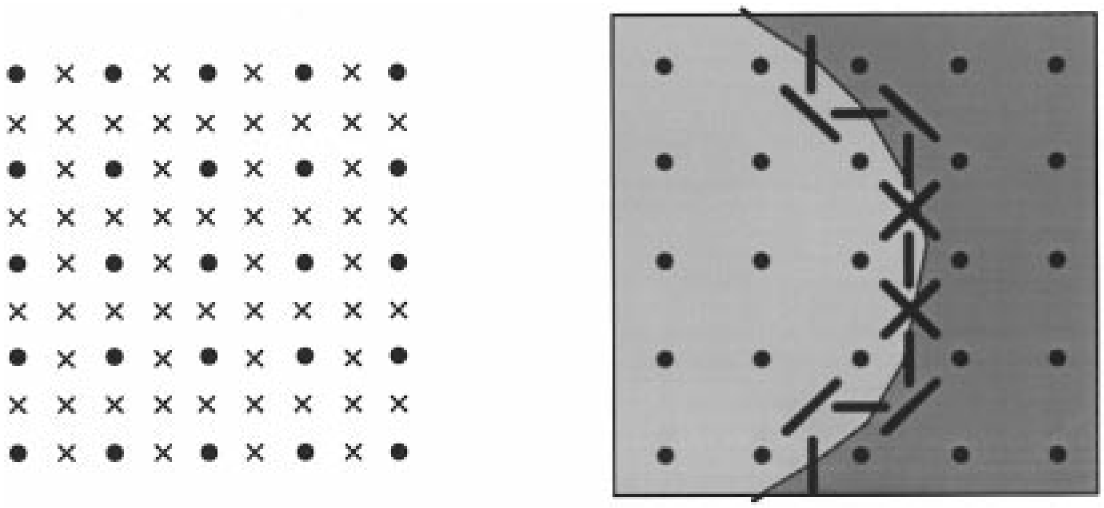
for $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, N-1$. Then, the quantity

$$V_h(\boldsymbol{f}) \triangleq \sum_{i=1}^{M} \sum_{j=1}^{N-1} (1 - l_v(i,j)) \cdot \{w_h(i,j)[f(i,j) - f(i,j+1)]\}^2 \tag{12}$$

gives the weighted variation of the entire image along the horizontal direction. For the purpose of artifacts removal, the factors $w_h(i,j)$ in (13) are used to quantify the local image activity along the horizontal direction. Thus, by forcing the quantity $V_h(\boldsymbol{f})$ small we can enforce smoothness along the horizontal direction in an image. This leads to the following constraint set:

$$C_h \triangleq \{\boldsymbol{f}: V_h(\boldsymbol{f}) \leq E_h^2\}. \tag{13}$$

Note from (12) that the variations at the 90° line sites are excluded from the term $V_h(\boldsymbol{f})$. As a result, the projection onto the set C_h will result in *spatially adaptive* smoothing of the image pixels along the horizontal direction, and in particular image pixels at the 90° line sites are excluded from being smoothed. This certainly helps to remove coding artifacts such as ringing in a compressed image, as will become clear from the discussion in the next section. In a similar fashion, $l_h(i,j), l_p(i,j)$, and $l_n(i,j)$ are defined as in (11) for the 0°, +45°, and −45° line processes, respectively. Also, the



(a)

(b)

Fig. 1. Illustration of line processes sites and their realization. (a) "•" indicates the pixel locations and "×" the possible line process sites. (b) "−," "|," "/," and "\" indicate the existence of line processes in the 0°, 90°, +45°, and −45° directions, respectively, for the edge shown.

weighted variations $V_v(\boldsymbol{f}), V_p(\boldsymbol{f})$, and $V_n(\boldsymbol{f})$ along the 90°, +45°, and −45° directions, respectively, are defined as in (12). Consequently, smoothness sets C_v, C_p, and C_n are defined, respectively, for the 90°, +45°, and −45° directions.

In order to apply these directional smoothness constraint sets in a recovery algorithm, we need to compute their associated projections. Unfortunately, this by no means is an easy matter. To illustrate this point, let's try to derive the projector P_h for the set C_h. Consider an image vector $\boldsymbol{f} \notin C_h$, and let $\boldsymbol{g}$ denote its projection onto C_h, i.e., $\boldsymbol{g} = P_h\boldsymbol{f}$. To obtain $\boldsymbol{g}$, we use the Lagrange multipliers approach. Consider the function

$$J = \sum_{i=1}^{M}\sum_{j=1}^{N}[(g(i,j) - f(i,j)]^2 + \lambda[V_h(\boldsymbol{g}) - E_h^2]. \qquad (14)$$

Taking the partial derivatives $\partial J/\partial g(i,j)$ and setting them to zero yields the following systems of equations:

$$D_i(\lambda)\cdot \boldsymbol{g}_{\boldsymbol{i}}' = \boldsymbol{f}_{\boldsymbol{i}}', \qquad i = 1, 2, \cdots, M \qquad (15)$$

where$\boldsymbol{g}_{\boldsymbol{i}}' = (g(i,1), g(i,2), \cdots, g(iN))^T, \boldsymbol{f}_{\boldsymbol{i}}' = (f(i,1), f(i,2), \cdots, f(i,N))^T$,and $D_i(\lambda)$is a matrix defined by the equation shown at the bottom of the page, where for convenience the following notation is used: $\overline{w}_h^2(i,j) \triangleq [1 - l_h(i,j)]w_h^2(i,j)$. Thus, the projection $\boldsymbol{g}$ can be found by solving for the row vectors $\boldsymbol{g}_{\boldsymbol{i}}'$ from (15). The parameter λ in the matrix $D_i(\lambda)$ is determined through the condition that the resulting $\boldsymbol{g}$ should satisfy

$$V_h(\boldsymbol{g}) = E_h^2. \qquad (16)$$

From (15) it is clear that the solution $\boldsymbol{g}_{\boldsymbol{i}}' = D_i^{-1}(\lambda)\boldsymbol{f}_{\boldsymbol{i}}'$ is a nontrivial nonlinear function of λ. As a result, computing the projection onto the set C_h for a large N becomes a difficult task, if not impossible. Similarly, one can verify that the same difficulty exists for computing the projections of other directional smoothness constraint sets. In the next section, we are going to apply a DAC strategy to overcome this difficulty.

IV. Smoothness Constraint Sets Using Divide-and-Conquer

As explained in Section III, it is difficult to include the directional smoothness constraint sets C_h, C_v, C_p, and C_n directly in a recovery algorithm because of the numerical difficulty in computing their projections. In the following we will apply a DAC strategy to these sets so that new smoothness constraint sets are derived and their projections are much easier to compute. To derive these new constraint sets, the directional variation term of the entire image in each set, e.g., $V_h(\boldsymbol{f})$ in C_h in (12), is decomposed into a summation of subterms in an interleaved fashion such that each subterm accounts for a part of the total directional variation. Then smoothness constraints sets are introduced on these subterms of directional variations. Due to the inherent blocking structure of BDCT compression, special attention is paid to the horizontal/vertical smoothness sets C_h and C_v so that the resulting new smoothness sets can also remove blocking artifacts in addition to ringing artifacts. This is possible because of the flexibility of the POCS approach.

$$\begin{bmatrix} 1+\lambda\overline{w}_h^2(i,1) & -\lambda\overline{w}_h^2(i,1) & 0 & 0 & \cdots & 0 \\ -\lambda\overline{w}_h^2(i,1) & 1+\lambda[\overline{w}_h^2(i,1)+\overline{w}_h^2(i,2)] & -\lambda\overline{w}_h^2(i,2) & 0 & \cdots & 0 \\ 0 & -\lambda\overline{w}_h^2(i,2) & 1+\lambda[\overline{w}_h^2(i,2)+\overline{w}_h^2(i,3)] & -\lambda\overline{w}_h^2(i,3) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1+\lambda\overline{w}_h^2(i,N-1) \end{bmatrix}$$

A. Horizontal/Vertical Smoothness Constraint Sets

Consider the horizontal variation term

$$V_h(\boldsymbol{f}) = \sum_{i=1}^{M}\sum_{j=1}^{N-1}(1-l_v(i,j)) \cdot \{w_h(i,j)[f(i,j)-f(i,j+1)]\}^2. \tag{17}$$

For notational simplicity, assume N a multiple of 8. Then for each $k = 1, 2, \cdots, 8$, define

$$V_{hk}(\boldsymbol{f}) \triangleq \sum_{i=1}^{M}\sum_{b=0}^{N_k}(1-l_v(i,8\cdot b+k))\{w_h(i,8\cdot b+k) \cdot [f(i,8\cdot b+k)-f(i,8\cdot b+k+1)]\}^2 \tag{18}$$

where $N_k = N/8-2$ for $k = 8$, and $N/8-1$ otherwise. Clearly

$$V_h(\boldsymbol{f}) = \sum_{k=1}^{8} V_{hk}(\boldsymbol{f}). \tag{19}$$

In the above the horizontal variation $V_h(\boldsymbol{f})$ of the entire image has been divided into a sum of smaller terms, $V_{hk}(\boldsymbol{f})$. Each term $V_{hk}(\boldsymbol{f})$ captures the weighted horizontal variation for every other eight columns of the image. Here, $V_{hk}(\boldsymbol{f})$ is so defined to reflect the fact that 8 × 8 block-size is used for the BDCT. Indeed, for $k = 1, 2, \cdots, 7$, the term $V_{hk}(\boldsymbol{f})$ is the sum of the weighted variations between the kth column and $(k+1)$th column within all coding blocks; and for $k = 8$, it is the sum of the weighted variations between all block boundary columns.

Since each of the smaller terms $V_{hk}(\boldsymbol{f})$ captures the horizontal variations of the image in part, they can be used to define constraint sets to enforce the horizontal smoothness. The following constraint sets follow:

$$C_{hk} \triangleq \{\boldsymbol{f}\colon V_{hk}(\boldsymbol{f}) \le E_{hk}^2\} \quad \text{for} \quad k = 1, 2, \cdots, 8. \tag{20}$$

Note that for $\Sigma_{k=1}^{8} E_{hk}^2 \le E_h^2, \boldsymbol{f} \in \cap_{k=1}^{8} C_{hk}$ implies that $\boldsymbol{f} \in C_h$. That is, the constraint imposed by the set C_h is automatically satisfied provided the constraints by the sets C_{hk} are satisfied altogether. Thus, the sets C_{hk} can be used instead of C_h in a recovery algorithm to enforce smoothness along the horizontal direction. There are two advantages in using these sets. One of them is that blocking artifacts are automatically taken into account by these sets. This is because the set C_{h8} essentially enforces the smoothness between block boundary columns, a role that the set C_s in (9) and the set C_w in (10) played. The other advantage is that their projections, unlike their counterpart C_h, can be computed efficiently, as will become clear in the following derivation.

It is straightforward to demonstrate that the sets C_{hk} are closed and convex. To derive the projector P_{hk} for C_{hk}, let $\boldsymbol{g} \triangleq P_{hk}\boldsymbol{f}$ for an image $\boldsymbol{f} \notin C_{vk}$. Consider the Lagrange auxiliary function

$$J = \sum_{i=1}^{M}\sum_{j=1}^{N}[g(i,j)-f(i,j)]^2 + \lambda[V_{hk}(\boldsymbol{g}) - E_{hk}^2]. \tag{21}$$

Taking the partial derivatives $\partial J/\partial g(i,j)$ and setting them to zero yields the following.

1) For each $i = 1, 2, \cdots, N$, and for each $j = 8\cdot b$ with $b = 0, 1, \cdots, N_k$,

$$\begin{aligned}&[g(i,j+k)-f(i,j+k)] \\ &\quad + \lambda(1-l_h(i,j+k))w_h^2(i,j+k) \\ &\quad \cdot [g(i,j+k)-g(i,j+k+1)] = 0 \\ &[g(i,j+k+1)-f(i,j+k+1)] \\ &\quad - \lambda(1-l_h(i,j+k))w_h^2(i,j+k) \\ &\quad \cdot [g(i,j+k)-g(i,j+k+1)] = 0.\end{aligned}$$

2) The rest of the pixels $g(i,j)$ do not appear in the term $V_{hk}(\boldsymbol{g})$, so we have

$$g(i,j) - f(i,j) = 0. \tag{22}$$

Solving these equations yields the following projection $\boldsymbol{g}$.

1) For each $i = 1, 2, \cdots, N$, and for each $j = 8\cdot b$ with $b = 0, 1, \cdots, N_k$,

$$\begin{aligned}&g(i,j+k) \\ &\quad = f(i,j+k) - \beta_{i,j,k}[f(i,j+k)-f(i,j+k+1)]\end{aligned} \tag{23}$$

$$\begin{aligned}&g(i,j+k+1) \\ &\quad = f(i,j+k+1) + \beta_{i,j,k}[f(i,j+k)-f(i,j+k+1)]\end{aligned} \tag{24}$$

where

$$\beta_{i,j,k} \triangleq \begin{cases} 0, & \text{if } l_v(i,j+k) = 1 \\ \dfrac{\lambda w_h^2(i,j+k)}{1+2\lambda w_h^2(i,j+k)}, & \text{otherwise.}\end{cases} \tag{25}$$

2) For the rest of the pixels

$$g(i,j) = f(i,j). \tag{26}$$

The parameter λ in (25) is found through the condition that the resulting image vector $\boldsymbol{g}$ satisfies $V_{hk}(\boldsymbol{g}) = E_{hk}^2$, or equivalently

$$\Psi(\lambda) \triangleq V_{hk}(\boldsymbol{g}) - E_{hk}^2 = 0. \tag{27}$$

Observing (18) and (23)–(25), we have

$$\Psi(\lambda) = \sum_{i=1}^{M}\sum_{b=0}^{N_k}\frac{(1-l_v(i,8\cdot b+k))w_h^2(i,8\cdot b+k)}{[1+2\lambda w_h^2(i,8\cdot b+k)]^2} \cdot [f(i,8\cdot b+k)-f(i,8\cdot b+k+1)]^2 - E_{hk}^2 = 0. \tag{28}$$

Clearly, this is a nonlinear equation in λ. In [23], it was shown that this type of equation has a unique positive root in λ and it is this root that corresponds to the projection $\boldsymbol{g}$; furthermore, the Newton's method given by

$$\lambda_{n+1} \triangleq \lambda_n - \Psi(\lambda_n)/\Psi'(\lambda_n) \tag{29}$$

with $\lambda_0 = 0$ converges *increasingly* to this positive root.

A few remarks are in order. First, from (23)–(25) we observe 1) when $l_v(i, j+k) = 1$, i.e, when there exists an edge between the pixels $(i, j+k)$ and $(i, j+k+1)$, we have $g(i, j+k) = f(i, j+k)$ and $g(i, j+k+1) = f(i, j+k+1)$, that is, no smoothing is done over the pixels at the line process sites; and 2) when $l_v(i, j+k) = 0$, i.e, when no edge exists between the pixels $(i, j+k)$ and $(i, j+k+1)$, we obtain

$$\begin{aligned} &|g(i, j+k) - g(i, j+k+1)| \\ &\quad = \frac{1}{1+2\lambda w_h^2(i, j+k)} |f(i, j+k) - f(i, j+k+1)|. \end{aligned} \tag{30}$$

Clearly, the difference between the image pixels is reduced in the resulting image and the degree of reduction is determined by the local weighting factor $w_h^2(i, j+k)$. This demonstrates that projecting an image vector onto the set C_{hk} indeed enforces adaptive smoothing along the horizontal direction and, furthermore, the line processes $l_v(i, j)$ help preserving the edges in the image. Finally, (26) reveals that each set C_{hk} enforces the smoothness only for every other eight columns of the image. Thus, by including all these sets in a recovery algorithm, the overall horizontal smoothness can be enforced. The above development has employed the DAC strategy to break up the overall smoothness constraint into pieces so that each piece is much easier to deal with.

A final remark is that when all the visibility factors $w_h(i, j)$ assume a uniform value, say one, (28) simply becomes a second-order equation in λ. In such a case, λ has a simple closed-form solution. Furthermore, the computations in (23) and (24) also become much simpler. Thus, the constraint sets C_{hk} can be simplified by segmenting the image according to its local activity factors $w_h(i, j)$ so that constraint sets can be defined for each segment using a uniform weight. Note that a similar approach was discussed in detail in [23]. The interested reader is referred to [23] for more details. Note that this approach also applies to the smoothness constraint sets that are to be defined in the following.

Similarly, by applying the same strategy we can introduce constraint sets $C_{vk}, k = 1, 2, \cdots, 8$, in place of the set C_v to enforce vertical smoothness.

B. Diagonal Smoothness Constraint Sets

Consider the smoothness constraint set C_n along the $-45°$ direction

$$C_n = \{\boldsymbol{f}: V_n(\boldsymbol{f}) \le E_n^2\}. \tag{31}$$

The term $V_n(\boldsymbol{f})$ captures the weighted variation of the entire image along the $-45°$ direction. For an $M \times N$ image, there are a total of $D = M + N - 1$ diagonals along the $+45°$ direction. Let $\mathcal{D}_d$ denote the set of the pixels on the dth diagonal, i.e.,

$$\begin{aligned} \mathcal{D}_d \triangleq \{(i, j)\colon\ & i + j - 1 = d, i = 1, 2, \cdots, M, \\ & j = 1, 2, \cdots, N\}. \end{aligned} \tag{32}$$

For example, the first diagonal $\mathcal{D}_1$ contains pixel (1, 1); the second diagonal $\mathcal{D}_2$ contains pixels (2, 1) and (1, 2); $\cdots$, etc..

Then, the term $V_n(\boldsymbol{f})$ can be rewritten as

$$\begin{aligned} V_n(\boldsymbol{f}) = \sum_{d=1}^{D-2} \sum_{(i,j)\in\mathcal{D}_d} &(1 - l_p(i, j)) w_n^2(i, j) \\ &\cdot [f(i, j) - f(i+1, j+1)]^2. \end{aligned} \tag{33}$$

If we define

$$\begin{aligned} V_{n1}(\boldsymbol{f}) \triangleq \sum_{d=1,2,5,6,9,10,\cdots} \sum_{(i,j)\in\mathcal{D}_d} &(1 - l_p(i, j)) w_n^2(i, j) \\ &\cdot [f(i, j) - f(i+1, j+1)]^2 \end{aligned} \tag{34}$$

and

$$\begin{aligned} V_{n2}(\boldsymbol{f}) \triangleq \sum_{d=3,4,7,8,11,12,\cdots} \sum_{(i,j)\in\mathcal{D}_d} &(1 - l_p(i, j)) w_n^2(i, j) \\ &\cdot [f(i, j) - f(i+1, j+1)]^2 \end{aligned} \tag{35}$$

then

$$V_n(\boldsymbol{f}) = V_{n1}(\boldsymbol{f}) + V_{n2}(\boldsymbol{f}). \tag{36}$$

That is, the total variation along the $-45°$ direction has been divided into two terms. As illustrated in Fig. 2(a), the term $V_{n1}(\boldsymbol{f})$ captures the variations between the pixels connected by the solid arrows, while the term $V_{n2}(\boldsymbol{f})$ captures that between pixels connected by the dashed arrows.

Based on $V_{n1}(\boldsymbol{f})$ and $V_{n2}(\boldsymbol{f})$,P we can define the following smoothness constraint sets

$$C_{n1} = \{\boldsymbol{f}: V_{n1}(\boldsymbol{f}) \le E_{n1}^2\} \tag{37}$$

and

$$C_{n2} = \{\boldsymbol{f}\colon\ V_{n2}(\boldsymbol{f}) \le E_{n2}^2\}. \tag{38}$$

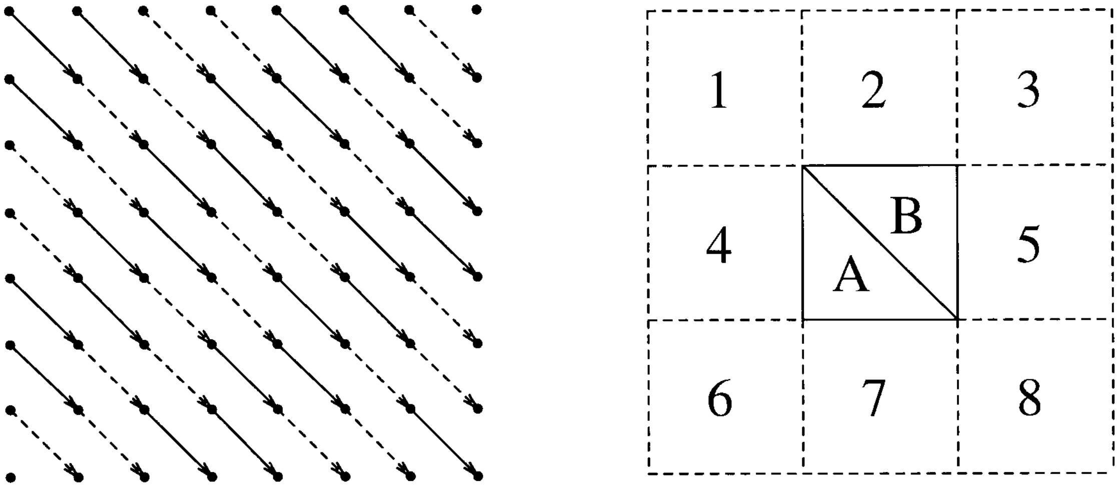
Note that for $E_{n1}^2 + E_{n2}^2 \le E_n^2, \boldsymbol{f} \in C_{n1} \cap C_{n2}$ implies that $\boldsymbol{f} \in C_n$. That is, the constraint imposed by the set C_n is automatically satisfied provided that the constraints by the sets C_{hk} are both satisfied. Thus, the sets C_{n1} and C_{n2} can be used instead of C_n in a recovery algorithm to enforce smoothness along the $-45°$ direction. As illustrated in Fig. 2(a), the set C_{n1} enforces smoothness between the pixels connected by the solid arrows, while C_{n2} enforces smoothness between the pixels connected by the dashed arrows. As a result, the smoothness for the entire image is enforced when these two sets are used together.

It is straightforward to demonstrate that the sets C_{n1} and C_{n2} are closed and convex. The projections for C_{n1} and C_{n2} can be derived in a fashion similar to that for the sets C_{hk}. The result for the set C_{n1} is furnished below.

For an image vector $\boldsymbol{f} \notin C_{n1}$, its projection $\boldsymbol{g} \triangleq P_{n1}\boldsymbol{f}$ onto the set C_{n1} is computed as follows.

1) Solve for λ from

$$\begin{aligned} \sum_{d=1,2,5,6,9,10,\cdots} \sum_{(i,j)\in\mathcal{D}_d} &\frac{(1 - l_p(i, j)) w_n^2(i, j)}{[1 + 2\lambda w_n^2(i, j)]^2} \\ &\cdot [f(i, j) - f(i+1, j+1)]^2 - E_{n1}^2 = 0. \end{aligned} \tag{39}$$



(a)

(b)

Fig. 2. (a) Illustration of the pixel grids used in the DAC strategy for the smoothness set in the $-45°$ direction. (b) Illustration of the neighboring blocks used to compute visibility in the $-45°$ direction.

2) For $(i,j) \in \mathcal{D}_d$ with $d = 1, 2, 5, 6, 9, 10, \cdots$, compute

$$g(i,j) = f(i,j) - \gamma_{i,j}[f(i,j) - f(i+1,j+1)] \tag{40}$$

$$g(i+1,j+1) = f(i+1,j+1) + \gamma_{i,j} \cdot [f(i,j) - f(i+1,j+1)] \tag{41}$$

where

$$\gamma_{i,j} \triangleq \begin{cases} 0, & \text{if } l_p(i,j) = 1 \\ \dfrac{\lambda w_n^2(i,j)}{1 + 2\lambda w_n^2(i,j)}, & \text{otherwise.} \end{cases} \tag{42}$$

Note that when $l_p(i,j) = 1$, i.e, when there exists an edge between the pixels (i,j) and $(i+1,j+1)$, we have $g(i,j) = f(i,j)$ and $g(i+1,j+1) = f(i+1,j+1)$. That is, no smoothing is done over the pixels (i,j) and $(i+1,j+1)$. This means that the projection onto the set C_{n1} indeed preserves the edges in the image.

3) The rest of the image pixels are unchanged.

In a similar fashion, we can define two sets C_{p1} and C_{p2} in place of the set C_p to enforce smoothness along the +45° direction. Again, the details are omitted for brevity.

V. Determining Parameters for the Smoothness Constraint Sets

In the definition of the directional smoothness constraint sets in Sections III and IV, a few parameters were introduced. For example, for the horizontal smoothness sets

$$C_{hk} = \{\boldsymbol{f}\colon V_{hk}(\boldsymbol{f}) \le E_{hk}^2\} \tag{43}$$

where $k = 1, 2, \cdots, 8$, the following parameters are used: the 90° line processes $l_v(i,j)$, the horizontal activity factors $w_h(i,j)$, and the smoothness bounds E_{hk}^2. These parameters have to be determined when the sets are used in a recovery algorithm. One way would be to determine them from the original image at the expense of extra coding cost. For practical purposes, a more reasonable approach is to determine them from the compressed image data. In the following, the latter approach is discussed.

A. The Line Processes

Edge detection is a well-known image processing problem; see [2], for example. In this study, directional derivatives are used to estimate the line processes. Consider the 90° line process for example. At site (i,j), we compute the first order horizontal difference from the compressed image $\boldsymbol{f}'$

$$|D_h \boldsymbol{f}'_{(i,j)}| \triangleq |f'(i,j) - f'(i,j+1)|. \tag{44}$$

Then the line process $l_v(i,j)$ is determined using a thresholding decision rule, i.e.,

$$l_v(i,j) = \begin{cases} 1, & \text{if } |D_h \boldsymbol{f}'_{(i,j)}| \ge T_h \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

where T_h is the decision threshold.

The choice of the threshold T_h needs some discussion. Due to the existence of blocking artifacts in the compressed image $\boldsymbol{f}'$ for a BDCT-based codec, the quantity $|D_h \boldsymbol{f}'_{(i,j)}|$ at a block boundary site tends to be larger than that at a site inside a block. As a result, if T_h is chosen to be too small, many sites at block boundaries would be falsely classified as line process sites. This obviously is undesirable, since it will affect the removal of blocking artifacts. Thus, the threshold T_h needs to be chosen to reflect the degree of blocking. Another consideration is that edges are typically small probability events in a natural image. Thus, only a small portion of sites at block boundaries should be classified as edges. With these in mind, the following estimation scheme is used:

$$T_h \triangleq \mu_h + \alpha\sigma_h \tag{46}$$

where μ_h, σ_h^2 denote, respectively, the mean and the variance of $|D_h \boldsymbol{f}'_{(i,j)}|$ over the vertical block boundary sites. In our experiments, it is demonstrated that α typically in the range of $0.5 \sim 2.0$ gives good results. It is also demonstrated that this approach adapts very well to the coding bit rate, which determines the severeness of the coding artifacts.

In a similar fashion, the other line processes can be estimated. For example, for the $+45°$ line processes the first order difference $|D_n \boldsymbol{f}'_{(i,j)} \triangleq |f'(i,j) - f'(i+1, j+1)|$ is used. In such a case, the decision threshold T_p is simply chosen to be the average the decision thresholds for the 0° and 90° line processes.

B. The Visibility Weights

The weighting factors in the directional smoothness sets are used to capture both the local statistical properties of an image and the characteristics of the human visual system. The introduction of these weights is based on the observation that the visibility of coding artifacts, like that of noise [1], decreases with the local image activity and in very bright or very dark areas of the image [23]. In the past, a number of different methods have been proposed to quantify the local image activity; see [1], [8], and [10], for example.

In this paper, the effect of coding artifacts is taken into account for the estimation of the local image activity from compressed image data. The following facts are considered: 1) the existence of blocking artifacts tends to cause the image variation to become larger at a block boundary site; and 2) the presence of ringing artifacts tends to cause the image variation to become larger at sites that are close to edges. As a result, in such cases the local image activity of the compressed image is no longer a good indication of the activity of the original image. Fortunately, both the ringing and the blocking artifacts have only limited extent in their occurrence. That is, the blocking artifacts are limited only to the sites at the block boundaries, and ringing artifacts are limited only within blocks inside which edges exist. Therefore, the local activity for these questionable sites in the above cases can be estimated from their neighboring sites.

With the above considerations in mind, the horizontal activity factors $w_h(i,j)$, for example, are determined in the following fashion: At first, sites are classified into *computable* and *noncomputable* ones. A site is said to be computable if it lies inside a block that does not contain any 90° line processes; otherwise, it is said to be noncomputable. Clearly, a site is noncomputable only if it either lies at a block boundary or lies inside a block which contains 90° line processes.

For a computable site (i,j), the factor $w_h(i,j)$ is determined using the average horizontal variation of the compressed image along the ith row inside the block. That is

$$w_h(i,j) \triangleq \left(1 + \tfrac{1}{7} \sum_{j=j_l}^{j_r - 1} |f'(i,j) - f'(i,j+1)|\right)^{-1} \tag{47}$$

where j_l, j_r denote, respectively, the position of the leftmost and the rightmost column of the block. Note that in (47), the variations between block boundary pixels are not included. Also, one is added to the denominator to avoid mathematical singularity.

Next, consider a noncomputable site (i,j) that lies inside a block. In such a case, there exists at least one 90° line site inside the block. The weight $w_h(i,j)$ is computed from its neighboring blocks based on the relative location of the site (i,j) with respect to the line process sites. With $(i, j_l - 1), (i, j_r + 1)$ denoting the closest neighbors to the site (i,j) from its left neighboring block and its right neighboring block, respectively, the following possibilities exist.

- If no 90° line process exists from the site $(i, j_l - 1)$ to the site $(i, j_r + 1)$ along the ith row and at the same time both the sites $(i, j_l - 1)$ and $(i, j_r + 1)$ are computable, then

$$w_h(i,j) \triangleq \frac{w_h(i, j_l - 1) + w_h(i, j_r + 1)}{2}. \tag{48}$$

  That is, the average of the two immediate neighboring computable sites is used.
- Otherwise, if no 90° line process exists from the site $(i, j_l - 1)$ to the current site (i,j) along the ith row, then we set $w_h(i,j) \triangleq w_h(i, j_l - 1)$, provided that the site $(i, j_l - 1)$ is computable; Similarly, if no 90° line process site exists from the current site (i,j) to the site $(i, j_r + 1)$ along the ith row, then we set $w_h(i,j) \triangleq w_h(i, j_r + 1)$, provided that the site $(i, j_r + 1)$ is computable.
- Otherwise, for the other cases, set

$$w_h(i,j) \triangleq \frac{1}{1 + T_h}. \tag{49}$$

  Note that the above definition gives the minimum for the possible value of the weight $w_h(i,j)$ at a computable site. This is not difficult to see from (47) because of the edge detection scheme used for the 90° line process.

Finally, for a site (i,j) that is noncomputable because it lies at a block boundary, the average weight of its two horizontal neighboring sites is used provided that none of them are line process sites; in case either of them is a line process site but not the both, the weight of the other is used; otherwise, the minimum weight is used.

In a similar fashion, the weights along the other directions can be computed, though a few exceptions exist for the two diagonal directions. Take the weights $w_n(i,j)$ for example. First, if the site (i,j) is computable, i.e., when no $+45°$ line sites exist inside the block, the local average variation is found over a part of the block, unlike in (47) the average is over a single row. To be specific, consider a site within the center block of Fig. 2(b). If the site is in region A, i.e., the lower-left half of the block, then the average is over the region A; if it is in region B, then the average is over the region B; and finally, if it is on the diagonal that separates A and B, then the average is over the whole block. This new added complexity is due to the fact that the number of sites along a $-45°$ diagonal within a block is different among different diagonals. On the other hand, if the site (i,j) is noncomputable, then $w_n(i,j)$ is determined from its neighboring blocks. Note, however, the neighboring blocks in such a case are no longer fixed. Again, consider a site within the center block of Fig. 2(b). If the site is in region A, then the neighboring blocks are 4 and 7; if it is in region B, then the neighboring blocks are 2 and 5; and finally, if it is on the diagonal that separates A and B, then they are 1 and 8.

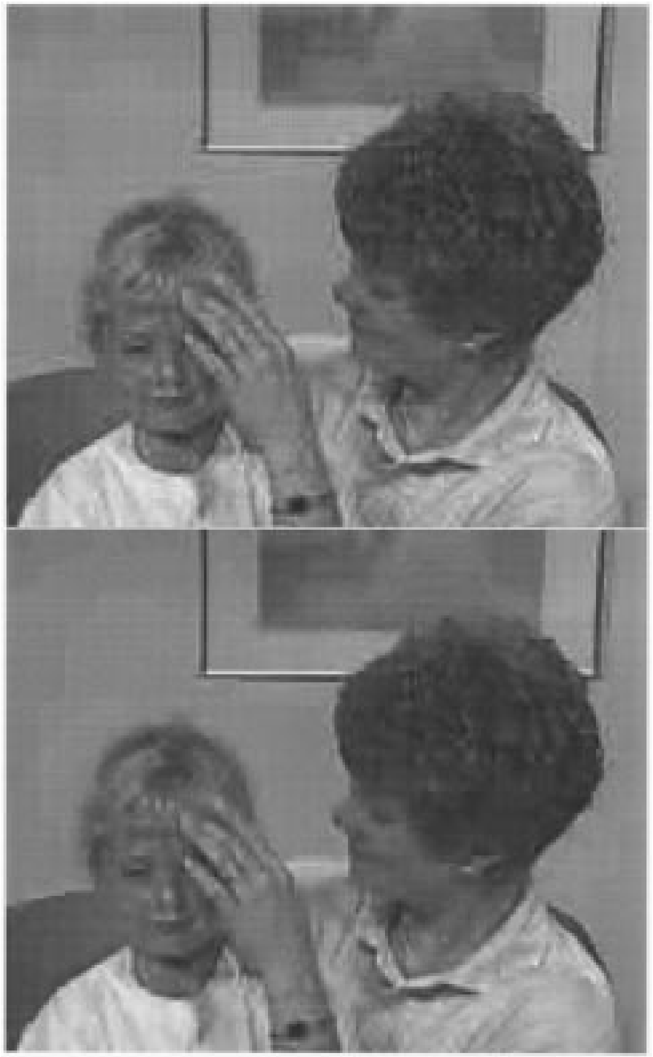


Fig. 3. "Mother and daughter" frame 21: MPEG compressed image (top); POCS reconstructed image (bottom).

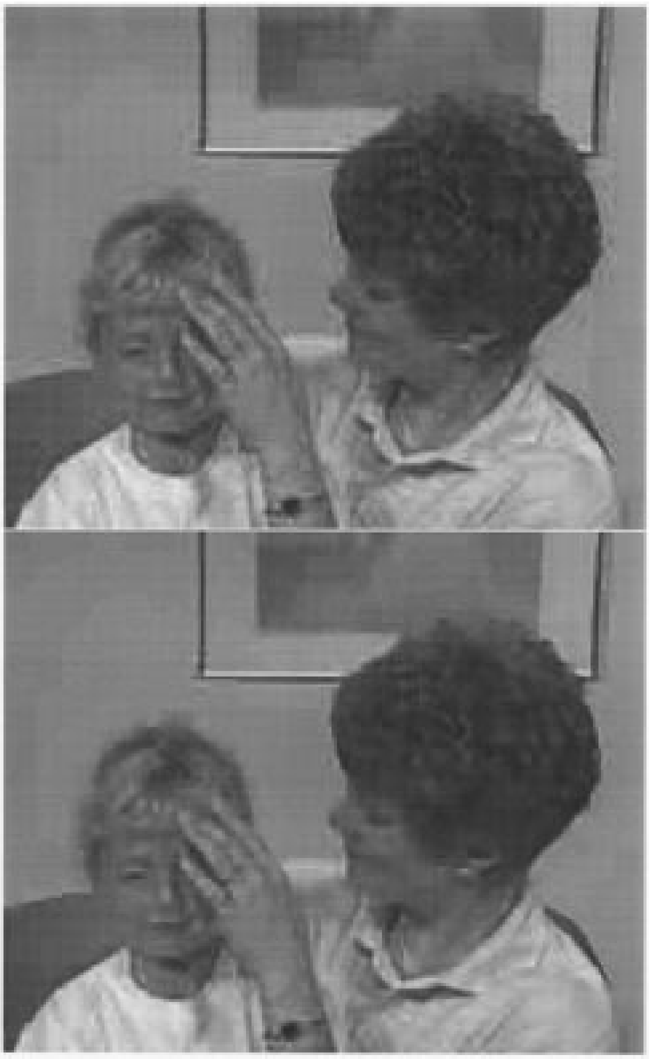


Fig. 4. "Mother and daughter" frame 22: MPEG compressed image (top); POCS reconstructed image (bottom).

C. The Smoothness Bounds

The bounds of the smoothness sets are determined from the received data, i.e., the compressed image $\boldsymbol{f}'$. Take the sets C_{hk} for example, the bounds E_{hk}^2 are determined by

$$E_{hk}^2 \triangleq \kappa \cdot \tfrac{1}{7} \sum_{l=1}^{7} V_{hl}(\boldsymbol{f}') \tag{50}$$

for $k = 1, 2, \cdots, 8$. In (50), the constant κ is used to control the degree of smoothness. It is demonstrated in the experiments that κ typically in the range of $\frac{1}{3} \sim 1$ gives good results. Note that the variation $V_{h8}(\boldsymbol{f}')$ is not included in (50) because it tends to be significantly larger than the others due to the existence of blocking artifacts.

Similarly, the bounds for the other smoothness sets can be determined. For example, for the sets C_{n1} and C_{n2} we simply set

$$E_{n1}^2 = E_{n2}^2 \triangleq \kappa \cdot \tfrac{1}{2} V_n(\boldsymbol{f}'). \tag{51}$$

VI. POCS Decoding for MPEG Video

In this section, we illustrate how to incorporate the directional smoothness constraint sets discussed in earlier sections into a recovery algorithm to decode images from compressed image data. Due to its recent popularity, the MPEG video compression standard is used here for the illustration.

In MPEG video coding, a video sequence is divided into groups of pictures (GOP's). In each GOP, pictures of different types are used to exploit both the spatial and the temporal redundancy [12]. More specifically, intracoded pictures (I-pictures), predictive coded pictures (P-pictures), and bidirectionally predictive coded pictures (B-pictures) are used. Since pictures of different types are coded differently, we need to define constraint sets to capture the transmitted data information for each picture type.

For I-pictures, the transmitted data information is captured by the constraint set in (3). For P-pictures, the quantized BDCT coefficients of the motion compensation error are transmitted in addition to the motion vector information. Based on the

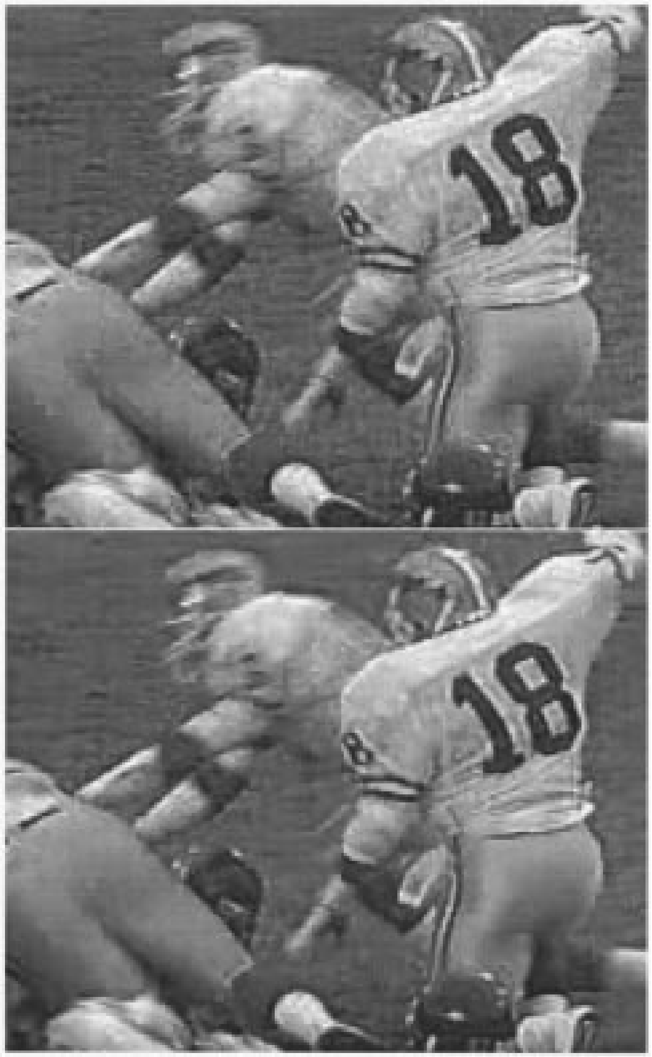


Fig. 5. "Football" frame 26: MPEG compressed image (top); POCS reconstructed image (bottom). Note that only a 176 × 144 segment of the whole image is shown here.

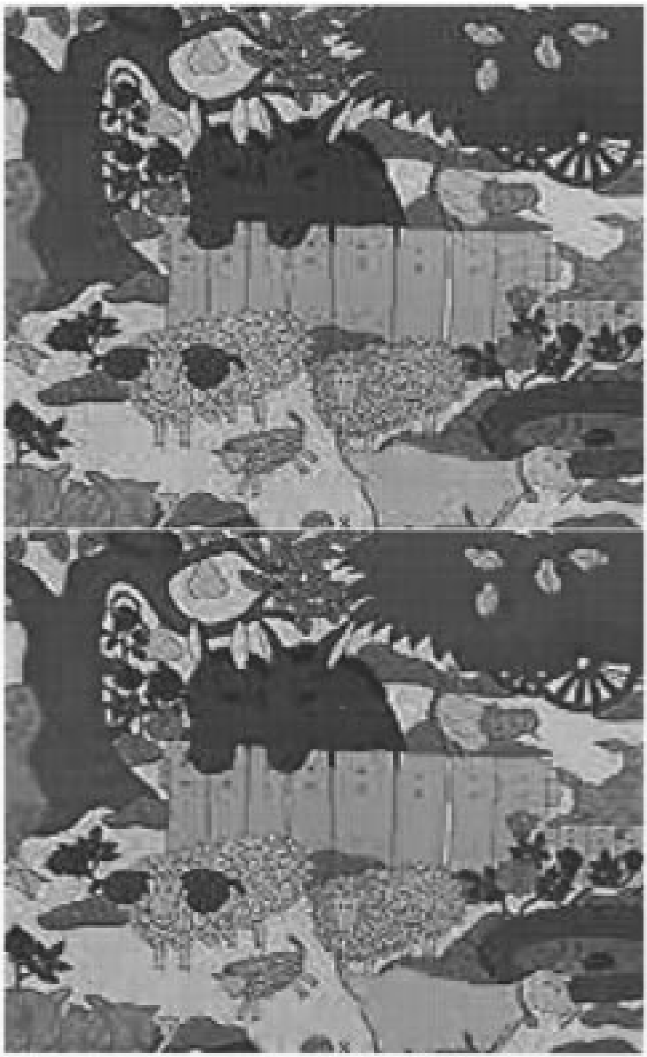


Fig. 6. "Mobile" frame 31: MPEG compressed image (top); POCS reconstructed image (bottom). Note that only a 176 × 144 segment of the whole image is shown here.

received data, the following constraint set [24] can be defined:

$$\begin{aligned} C_P \triangleq \{&\boldsymbol{f}\colon F_n^{\min} \le [B(\boldsymbol{f}-\hat{\boldsymbol{f}}_p)]_n \le F_n^{\max}\\ &n=1,2,\cdots,M\cdot N\} \end{aligned} \tag{52}$$

where $F_n^{\min}$ and $F_n^{\max}$ are the end-points of the quantization interval that is associated with the received quantized level of the nth BDCT coefficient $[B(\boldsymbol{f}-\hat{\boldsymbol{f}}_p)]_n$. In (52), $\hat{\boldsymbol{f}}_p$ denotes the predicted P-picture using the received motion vector information.

It is easy to show that the set C_P is closed and convex. Furthermore, for an image vector $\boldsymbol{f} \notin C_P$, its projection $\boldsymbol{g} = P_P\boldsymbol{f}$ onto C_P is given by

$$\boldsymbol{g} = \hat{\boldsymbol{f}}_p + B^{-1}\boldsymbol{F} \tag{53}$$

where the nth component of $\boldsymbol{F}$ is given by

$$F_n = \begin{cases} F_n^{\min}, & \text{if } [B(\boldsymbol{f}-\hat{\boldsymbol{f}}_p)]_n < F_n^{\min}\\ F_n^{\max}, & \text{if } [B(\boldsymbol{f}-\hat{\boldsymbol{f}}_p)]_n > F_n^{\max}\\ (B\boldsymbol{f})_n, & \text{if } F_n^{\min} \le [B(\boldsymbol{f}-\hat{\boldsymbol{f}}_p)]_n \le F_n^{\max} \end{cases} \tag{54}$$

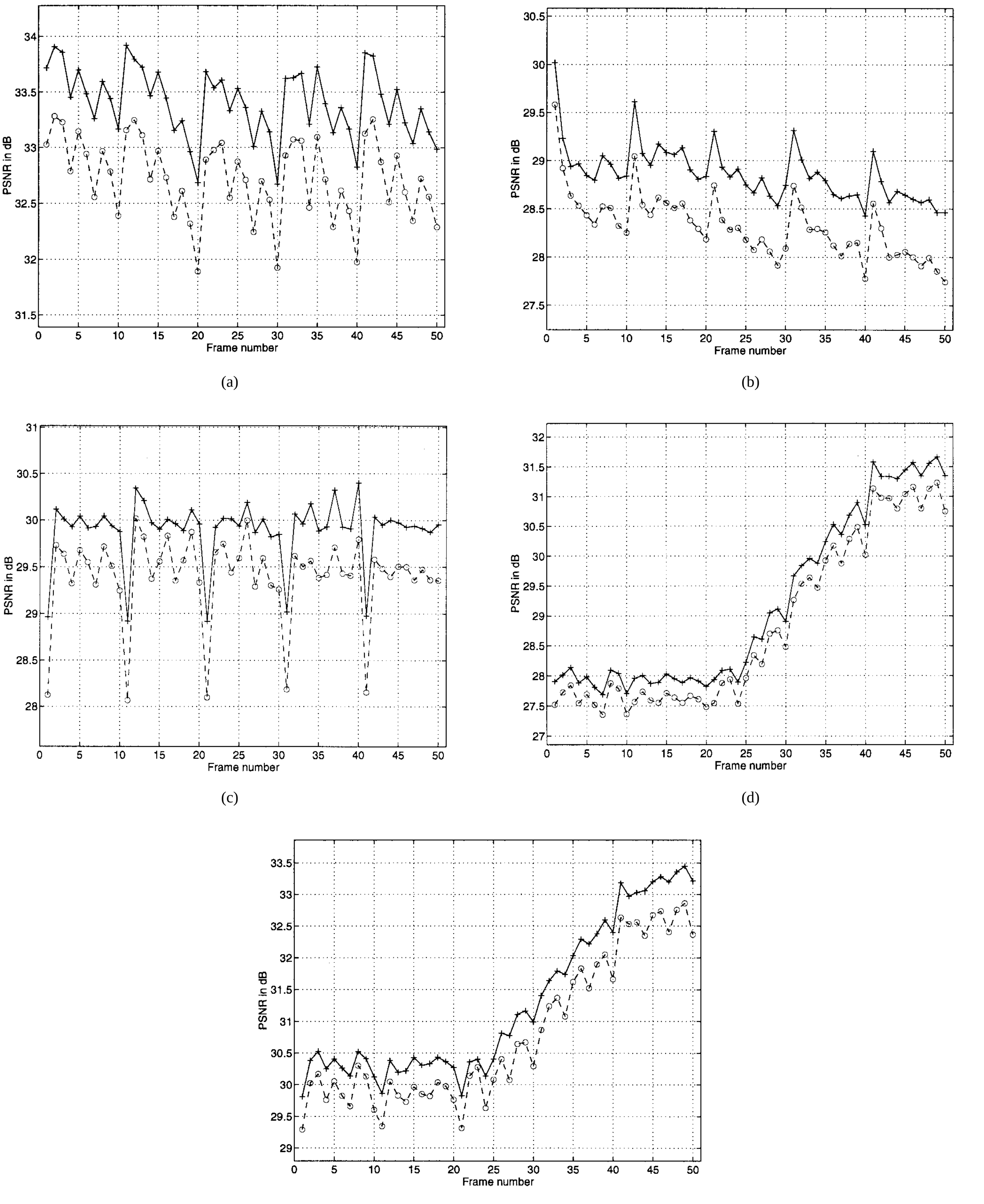
for $n = 1, 2, \cdots, M \cdot N$.

A constraint set of the same form as in (52) can be used to capture the data information for a B-picture. In such a case, the vector $\hat{\boldsymbol{f}}_p$ is replaced with the bidirectionally predicted B-picture using the received motion vector information.

With constraint sets clearly defined and their projections derived, the POCS algorithm in (2) can then be used to decode images from MPEG compressed data. For convenience, let $\boldsymbol{P}_h$ denote the concatenation of the projectors for the horizontal smoothness sets C_{hk}, i.e., $\boldsymbol{P}_h \triangleq P_{h8}\cdots P_{h2}P_{h1}$. Similarly, define $\boldsymbol{P}_v \triangleq P_{v8}\cdots P_{v2}P_{v1}, \boldsymbol{P}_n \triangleq P_{n2}P_{n1}$, and $\boldsymbol{P}_p \triangleq P_{p2}P_{p1}$. Then, the iterative POCS algorithm for an I-picture can be written as

$$\boldsymbol{f}_{k+1} = P_T\boldsymbol{P}_p\boldsymbol{P}_n\boldsymbol{P}_v\boldsymbol{P}_h\boldsymbol{f}_k; \qquad k = 0, 1, 2, \cdots \tag{55}$$

where P_T is the projector for the I-picture data set in (3). In practice, the MPEG compressed image can be used for the starting point $\boldsymbol{f}_0$. Similarly, the POCS algorithms for P- or B-pictures follow by replacing the projector P_T in (55) with the corresponding projector for the P- or B-picture data set.



(a)

(b)

(c)

(d)

(e)

Fig. 7. PSNR plots for MPEG compressed images (dashed) and POCS reconstructed images (solid). (a) "Mother and daughter" compressed at about 100 : 1. (b) "Football" compressed at about 25 : 1. (c) "Mobile" compressed at about 12 : 1. (d) "Table tennis" compressed at about 65 : 1. (e) "Table-tennis" compressed at about 35 : 1.

Note that additional constraint sets can be introduced into the POCS algorithm in (55) to take into account other available *prior* knowledge. One such example is the nonnegativity constraint on the pixel intensity. For more details see [22] and [23].

A final note is that the algorithm in (55) can also be used without modification to decode JPEG compressed images.

VII. Numerical Experiments

The proposed algorithms have been tested using both still images and image sequences. Due to space limitation, only a sample of the test results will be presented in the following to demonstrate the effectiveness of the proposed algorithms. One or two particular frames from each test sequence will be presented to demonstrate the quality improvement of the resulting images from the proposed algorithms over those of compressed ones. Also, to demonstrate the overall performance an objective metric the peak-signal-to-noise-ratio (PSNR) is used. The PSNR of a recovered $M \times N$ image $\boldsymbol{g}$ relative to its original image $\boldsymbol{f}$ is defined in decibels by

$$\text{PSNR} \triangleq 10 \log_{10} \frac{255^2 \cdot M \cdot N}{||\boldsymbol{f} - \boldsymbol{g}||^2}. \tag{56}$$

For comparison purposes, PSNR values are computed and plotted for both the compressed and the reconstructed images for the first 50 frames of each sequence.

The following test sequences are used in the following: "mother and daughter" (176 × 144), "football" (352 × 240), "mobile" (352 × 240), and "table-tennis" (352 × 240). The test sequences were first coded using an MPEG-based compression algorithm and then POCS reconstruction was applied. The following pattern was used for each GOP: "I, B, B, P, B, B, P, B, B, P." Purely for demonstration purposes, the default intra- and nonintraquantization matrices in MPEG-1 were used [12].

Shown in Fig. 3 are the MPEG compressed image (top) and the POCS reconstructed image (bottom) of frame 21 (I-frame) of the "mother and daughter" sequence. Also, Fig. 4 shows the results of frame 22 (B-frame) of this sequence. The PSNR result of this sequence is given in Fig. 7(a). The achieved MPEG compression ratio for this sequence is about 100 : 1.

Fig. 5 shows the MPEG compressed image (top) and the POCS reconstructed image (bottom) of frame 26 (B-frame) of the "football" sequence. The PSNR result for this sequence is shown in Fig. 7(b). The achieved MPEG compression ratio for this sequence is about 25 : 1.

Fig. 6 shows the MPEG compressed image (top) and the POCS reconstructed image (bottom) of frame 31 (I-frame) of the "mobile" sequence. The PSNR result for this sequence is shown in Fig. 7(c). The achieved MPEG compression ratio for this sequence is about 12 : 1. Note that for this sequence, only the line process was used in the POCS algorithm (i.e., a uniform visibility weights were used).

Finally, the PSNR results for the "table-tennis" sequence are presented in Fig. 7(d) and (e) when this sequence is compressed at ratio 65 : 1 and 35 : 1, respectively.

From these experiments, it is clear that the proposed algorithms can correct effectively MPEG compression artifacts such as blocking and ringing. The POCS reconstructed images have higher PSNR's than their MPEG counterparts. A final note is that the proposed POCS-based algorithms are observed to converge very rapidly. All the results given in this section are obtained after $3 \sim 5$ iterations.

VIII. Conclusions

In this paper we presented a new POCS-based recovery algorithm to reconstruct transform-based compressed images and video. The new feature of this algorithm comparing with our previous work in [22] and [23] is its capability to correct ringing artifacts. Since ringing is an artifact that appears in most transform-based codecs, this algorithm can be used for codecs that use other transforms as well. Also, this work provides a new approach for the use of line processes in image recovery problems. The resulting algorithm from this approach is easy to implement and converges very rapidly.

A remaining challenge in the video recovery problem is to further utilize the between-frame smoothness property in a video sequence. This is accomplished in [25] and [27] by using the transmitted motion vector information.

References

[1] G. Anderson and A. Netravali, "Image restoration based on a subjective criterion," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 845–853, Dec. 1976.

[2] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–697, Nov. 1986.

[3] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRF's: Surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 401–412, May 1991.

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[5] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1991.

[6] ISO/IEC CD 10918-1, *Digital Compression and Coding of Continuous-tone Still Images, Part 1: Requirements and Guidelines*, Mar. 15, 1991.

[7] M. Kaneko, Y. Hatori, and A Koike, "Improvement of transform coding algorithm for motion compensated interframe prediction errors," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 1068–1078, Aug. 1987.

[8] A. K. Katsaggelos, J. Biemond, R. M. Mersereau, and R. W. Schafer, "Nonstationary iterative image restoration," in *Proc. ICASSP-85*, pp. 696–699.

[9] C. Kuo and R. Hsieh, "Adaptive postprocessor for block encoded images," *IEEE Trans Circuits Syst. Video Technol.*, vol. 5, pp. 298–304, Aug. 1995.

[10] R. L. Lagendijk, J. Biemond, and D. E. Boekee, "Regularized iterative restoration with ringing reduction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1874–1888, Dec. 1988.

[11] J. Luo, C. Chen, K. Parker, and T. S. Huang, "Artifact reduction in low bit rate DCT-based image compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1363–1368, Sept. 1996.

[12] ISO/IEC DIS 11172, *Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbits/s*, 1992.

[13] F. C. Jeng and J. W. Woods, "Compound Gauss-Markov random fields for image restoration," *IEEE Trans. Signal Processing*, vol. 39, pp. 683–697, Mar. 1991.

[14] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 74–82, Apr. 1995.

[15] T. Ozcelik, J. Brailean, and A. K. Katsaggelos, "Image and video compression algorithms based on recovery techniques using mean field annealing," *Proc. IEEE,* vol. 83, pp. 304–316, Feb. 1995.

[16] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1258–1267, Oct. 1986.

[17] R. Rosenholtz and A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 91–94, Mar. 1992.

[18] K. Sauer, "Enhancement of low bit-rate coded images using edge detection and estimation," *Comput. Vis. Graph. Image Process.: Graph. Models Image Process.*, vol. 53, pp. 52–62, Jan. 1991.

[19] R. L. Stevenson, "Reduction of coding artifacts in transform image coding," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing,* Minneapolis, MN, 1993, pp. V-405–408.

[20] T. P. O'Rourke and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 490–409, Dec. 1995.

[21] J. W. Woods, S. Dravida, and R. Mediavilla, "Image estimation using doubly stochastic Gaussian random field models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 245–253, Mar. 1987.

[22] Y. Yang, N. Galatsanos, and A. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans Circuits Syst. Video Technol.*, vol. 3, pp. 421–432, Dec. 1993.

[23] ——, "Projection-based spatially adaptive image reconstruction of block-transform compressed images," *IEEE Trans. Image Processing,* vol. 4, pp. 896–908, July 1995.

[24] Y. Yang and N. Galatsanos, "Projection-based decoding of low bit-rate MPEG data," in *Proc. SPIE, Visual Communications and Image Processing*, Chicago, IL, Sept. 1994, pp. 331–342.

[25] ——, "Projection-based image recovery from low bit rate MPEG data," in *Proc. IASTED Int. Conf. on Signal and Image Processing*, Las Vegas, NV, Nov. 20–23, 1995, pp. 202–205.

[26] D. C. Youla, "Generalized image restoration by the method of alternating orthogonal projections," *IEEE Trans. Circuits Syst..*, vol. 25, pp. 694–702, Sept. 1978.

[27] M. Choi, Y. Yang, and N. Galatsanos, "Multichannel regularized recovery of compressed video," in *Proc. ICIP-97,* to be published.

Yongyi Yang (M'94) was born in Shandong, China in 1964. He received the B.S.E.E. and M.S.E.E. degrees from Northern Jiaotong University, Beijing, China, in 1985 and 1988, respectively. He received the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from Illinois Institute of Technology, Chicago, in 1992 and 1994, respectively.

He was previously with the faculty of the Institute of Information Science, Northern Jiaotong University. Currently, he is an Assistant Professor at the Department of Electrical and Computer Engineering, Illinois Institute of Technology. His research interests are in image recovery, image and video compression, general areas of signal processing, and applied mathematical and statistical methods.

Dr. Yang is a member of Sigma Xi and the IEEE Signal Processing Society.

Nikolas P. Galatsanos (S'84–M'89–SM'95) was born in Athens, Greece, in 1958. He received the Dipl. degree in electrical engineering from the National Technical University of Athens in 1982. He received the M.S. and the Ph.D. degrees, both in electrical and computer engineering, from the University of Wisconsin, Madison, in 1984 and 1989, respectively.

Since August of 1989, he has been on the faculty of the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, where he is currently an Associate Professor. His current research interests include image processing and, more specifically, recovery, compression of single and multichannel/frame images, and computer vision.

Dr. Galatsanos served as an Associate Editor for the IEEE Transactions on Image Processing and presently serves as an Associate Editor for Signal Processing Magazine.