# 5.8

# Recovery Methods for Postprocessing of Compressed Images

Y. Yang
*Illinois Institute of Technology*

N. P. Galatsanos
*University of Ioannina*

A. K. Katsaggelos
*Northwestern University*

1 Introduction ..... 761
2 Basic Theory of Projections onto Convex Sets ..... 762
3 Projections onto Convex Sets–Based Image Recovery from Compressed Data ..... 763
3.1 Overview of the Methodology • 3.2 The JPEG Discrete Cosine Transform Algorithm • 3.3 Vector Notation • 3.4 Constraint Set Based on Transform Coefficients • 3.5 Constraints Based on Prior Knowledge • 3.6 Recovery Algorithm • 3.7 Adaptive Processing • 3.8 Choice of Weight Factors in $\boldsymbol{W}$ for Adaptive Processing • 3.9 Determination of Constant $\lambda$ in the Projector $P_{wc}$ • 3.10 The Recovery Algorithm
4 Maximum A Posteriori Estimation for Postprocessing of Compressed Images ..... 770
4.1 Defintion of Likelihood • 4.2 Definition of Image Prior • 4.3 Image Reconstruction Algorithm
5 Conclusions ..... 772
References ..... 773

## 1 Introduction

The objective of postprocessing is to improve the quality of the images produced in the decoder of a lossy image compression system. Such systems produce high compression ratios; however, to do so they also discard information, which is deemed not important, and thus introduce distortion to the original image.

The distortions produced by lossy compression algorithms can be categorized into two classes. First, all lossy compression algorithms produce what is called "ringing" artifacts. This is nothing more than a Gibbs type of oscillations around sharp-intensity transitions in the image. These artifacts appear at high compression ratios in all transform-based codecs due to the low-pass nature of such systems, see for example Section 3.2 and [22] for the "classic" and the JPEG 2000 systems, respectively. These systems are by far the most popular compression systems. Second, the classic JPEG compression algorithm at high compression ratios produces what is called "blocking" artifact. This artifact originates from the independent quantization of the block discrete cosine transform (DCT) coefficients, which is used in the classic JPEG algorithm.

Since image compression is an important technologic problem, there has been a significant effort to improve via postprocessing the image produced by lossy image codecs. Consequently, there have been many techniques developed to ameliorate "ringing" and "blocking" artifacts in compressed images. A survey of these techniques is beyond the scope of this tutorial chapter. However, generally speaking, such efforts can be classified broadly into two categories. First, *filtering-based techniques*, where the immediate objective is to ameliorate these artifacts based on certain assumptions and prior knowledge about the original image that is not supposed to have such artifacts. The second category consists of *recovery-based techniques*, of which the objective is to

recover the original image rather than ameliorate the compression artifact. In recovery-based methods, the amelioration of the compression artifact is a "side benefit" that stems naturally from recovering the original image which is artifact free.

Traditionally, the fields of signal recovery and compression have developed separately because the problems that they examine seem very different and the techniques used seem unrelated. However, there is a growing consent that the two fields are quite related. Indeed, in signal recovery, one attempts to reconstruct the best possible replica of a signal from limited information. Similarly, the decoder of a lossy compressed image bit stream is faced with exactly the same problem. In addition to compression artifacts, errors can also occur due to the transmission channel. Therefore, signal recovery techniques need to be used in the receiver to reconstruct the best possible bit stream and thus "conceal" the channel errors. Although the channel error "concealment" problem is very similar, in this chapter we will focus only on recovery algorithms that ameliorate compression artifacts.

This chapter, presents in a tutorial manner, the second category of techniques for compressed image postprocessing. This choice reflects our expertise rather than our opinion about the relative importance of the two categaries of techniques; for more details on filtering based techniques and the error concealment problem see for example [23]. In what follows, we will present two general methodologies that have been applied successfully by a number of researchers to recovery-based postprocessing of compressed images. The choice of these techniques reflects their generality and the fact that they have been used widely to many other image recovery problems. More specifically, we will present the projections onto convex sets (POCS) and the maximum a posteriori (MAP) methodologies. In Section 2, we present the basic theory of the POCS methodology, and in Section 3 the details of the application of POCS to the compressed image postprocessing problem. In Section 4, the application of the MAP methodology is presented, and finally in Section 5, we present our conclusions. Concluding the introduction of this chapter, we want to point out that even in the context of the POCS and MAP methodologies for postprocessing of compressed images, this chapter is not intended to be a survey. Thus, the choice of the material presented here was mainly based on our view of its value as representative of this area.

## 2 Basic Theory of Projections onto Convex Sets

We begin with a few key concepts related to a convex set. In a vector space, a set is said to be convex if every point on the line segment connecting *any* two points in the set is also in the set. Thus, for a convex set $C$ the following holds true: Let $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $0 \leq \mu \leq 1$, then $\mathbf{x} = \mu\mathbf{x}_1 + (1-\mu)\mathbf{x}_2 \in C$. Or, equivalently, a set $C$ is convex if

$$\mathbf{x} = \mu\mathbf{x}_1 + (1-\mu)\mathbf{x}_2 \in C \tag{1}$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $0 \leq \mu \leq 1$.

Consider a set $C$ that is closed and convex in a Hilbert space $\mathbf{H}$ (e.g., the Euclidean space $R^n$). The following holds true: For each point $\mathbf{x} \in \mathbf{H}$, there exits a *unique* point $\mathbf{x}^*$ in $C$ that is closest to $\mathbf{x}$. In short,

$$\|\mathbf{x} - \mathbf{x}^*\| = \min_{\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|, \tag{2}$$

and $\mathbf{x}^* \in C$. Thus, Equation (2) defines a rule that assigns to every $\mathbf{x} \in \mathbf{H}$ its unique nearest neighbor in the set $C$, which we shall call the *projection* of $\mathbf{x}$ onto $C$. This rule is completely determined by the set $C$ and is called the *projector* or *projection operator* onto $C$, which we shall denote by $P_C$. That is, for every $\mathbf{x} \in \mathbf{H}$ its projection $P_C\mathbf{x}$ onto $C$ is defined by $\|\mathbf{x} - P_C\mathbf{x}\| = \min_{\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|$.

Following the definition of the projector $P_C$ onto a closed convex set $C$, we now introduce an extension of $P_C$ called a *relaxed projector*. For each constant $\lambda$ in the range of $(0, 2)$, we define an operator $T_C$ as

$$T_C = I + \lambda(P_C - I), \tag{3}$$

where $I$ is the identify operator on $\mathbf{H}$. That is, for each $\mathbf{x} \in \mathbf{H}$, the operator $T_C$ acts in the following fashion

$$T_C\mathbf{x} = \mathbf{x} + \lambda(P_C\mathbf{x} - \mathbf{x}) = (1-\lambda)\mathbf{x} + \lambda P_C\mathbf{x}. \tag{4}$$

The constant $\lambda$ is called the relaxation parameter of $T_C$. Clearly, when the constant $\lambda = 1$, $T_C$ is simply the projector $P_C$. As illustrated in Fig. 1, when $\lambda \in (0, 1)$ the point $T_C\mathbf{x}$ lies in-between the point $\mathbf{x}$ and its projection $P_C\mathbf{x}$. As a result, the point $T_C\mathbf{x}$ lies outside the set. On the other

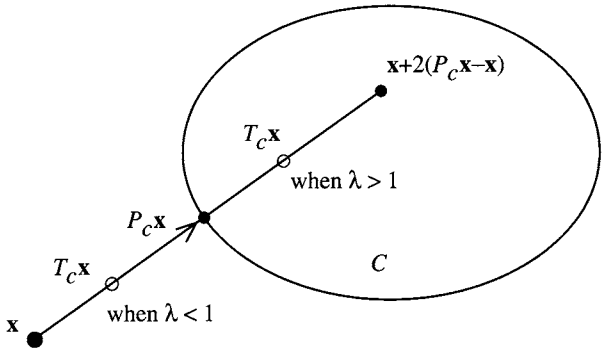


**FIGURE 1** Geometric interpretation of the operator $P_C$: when $\lambda$ varies from 0 to 2, $T_C\mathbf{x}$ traces out the line segment from the point $\mathbf{x}$ to the point $\mathbf{x} + 2(P_C\mathbf{x} - \mathbf{x})$.

hand, when $\lambda \in (1,2)$, the point $T_C\mathbf{x}$ lies farther away from the point $\mathbf{x}$ than the projection $P_C\mathbf{x}$ does. These properties of a relaxed projector are useful for speeding up the convergence of projection-type algorithms.

To the best of our knowledge the fundamental theory of POCS was developed in the 1960s by Gubin et al. [1], Bregman [2], Halperin [3], and Opial [4]. In turn, these researchers built on von Neumann's 1950s work on alternating orthogonal projections [5]. This theory was later introduced to the image processing community by Youla's influential work in [6] and [7]. Since then, it has found many applications in various signal and image problems [8,9].

Assume that $C_1, C_2, \cdots, C_m$ denote $m$ closed convex sets in a Hilbert space $\mathbf{H}$, and $C_0$ denotes their intersection set $C_0 = \bigcap_{i=1}^{m} C_i$. For each $i = 1, 2, \cdots, m$, let $P_i$ denote the projection operator onto the set $C_i$, and $T_i$ denote the corresponding relaxed projector $T_i = I + \lambda_i(P_i - I)$ for $\lambda_i \in (0,2)$. Then we have the following:

***Theorem 1 (Fundamental Theorem of POCS)***: Assume that $C_0$ is nonempty. Then for every $\lambda_i \in (0,2)$, $i = 1, 2, \cdots, m$, the sequence $\{\mathbf{x}_n\}$ generated by

$$\mathbf{x}_{n+1} = T_m T_{m-1} \cdots T_1 \mathbf{x}_n \tag{5}$$

with an arbitrary starting point $\mathbf{x}_0 \in \mathbf{H}$ will converge weakly to a point of $C_0$.

In a Hilbert space $\mathbf{H}$, a sequence $\{\mathbf{x}_n\}$ is said to converge *weakly* to a point $\mathbf{x}^*$ if the inner product $\langle \mathbf{x}_n, \mathbf{y} \rangle$ converges to $\langle \mathbf{x}^*, \mathbf{y} \rangle$ for every $\mathbf{y} \in \mathbf{H}$. In a finite-dimensional vector space $\mathbf{H}$, as is always the case when we perform computer implementations, weak convergence is equivalent to strong convergence, i.e., the more familiar form of $\lim_{n\to\infty} \|\mathbf{x}_n - \mathbf{x}^*\| = 0$.

The above theorem was first proved by Gubin, Polyak, and Raik in 1967 [1]. Later in 1982, Youla and Webb provided an alternative proof [7]. Youla's approach is based on an important result by Opial in the fixed-point theory of nonexpansive operators [4], which is closely followed in [9].

The iterative algorithm in (5) is generally known as the *convex projection algorithm* or simply the *POCS algorithm*. In particular, when the projector $P_i$ is used instead of $T_i$ for each set, the POCS algorithm in (5) reduces to

$$\mathbf{x}_{n+1} = P_m P_{m-1} \cdots P_1 \mathbf{x}_n. \tag{6}$$

The algorithm in (6) is better known as the *alternating projections*, or *successive projections algorithm*, and is most frequently used in the literature.

In its most general form, the POCS method for solving a practical problem has the following framework: We want to recover or determine an unknown quantity about which some information is known in the form of constraints. The unknown quantity is treated as a vector in a Hilbert space, and the known constraints are described in the form of closed convex sets in this space. Without loss of generality, assume that there are a total of $m$ such sets $C_1, C_2, \cdots, C_m$ available. Each set is usually associated with a single constraint, although sometimes it is convenient to include multiple constraints in a single set. Then, the intersection of all these sets, say $C_0 = \bigcap_{i=1}^{m} C_i$, will contain all the possible solutions to the problem because each solution satisfies all the available information about the unknown. To find such a solution, we apply the iterative POCS algorithms.

The key to the successful application of the POCS method is the definition of the appropriate constraint sets $C_1, C_2, \cdots, C_m$ to describe the available information. This is the creative part of the problem. The computation of the projection $P_i$ may be technically challenging but can usually be achieved, especially when the sets are defined in an amenable form. Interested readers are referred to [9] for various types of constraint sets and application examples. For extensions of this theory to non-convex sets, inconsistent sets, and parallel projections the interested reader is referred to [10–15].

# 3 Projections onto Convex Sets–Based Image Recovery from Compressed Data

## 3.1 Overview of the Methodology

In this Section we apply the POCS method to solving the problem of image recovery from compressed image data. In image compression, a source image is first converted to a series of digital symbols and then sent to the receiver (or stored in a storage medium). The receiver then tries to recover the source image that produced the received digital symbols. Unfortunately, this process is noninvertible (we deal here with lossy compression), in that the original source image can no longer be recovered exactly from the received digital symbols (owing to the use of quantization for compression efficiency).

To recover the source image from the compressed data, we will define two types of constraints for this problem: The first type is defined directly from the available compressed data, and the other type is defined in the form of *prior knowledge* about the source image. The latter is introduced to complement the available data such that the recovered image will not only faithfully reproduce the received data at the receiver, but also be free of undesirable artifacts caused by the compression process (i.e., quantization). While this framework is applicable to almost every known compression scheme, for clarity we will focus below on a block DCT-based compression scheme (popularized by JPEG and MPEG).

## 3.2 The JPEG Discrete Cosine Transform Algorithm

The JPEG DCT encoding/decoding scheme is shown in Fig. 2. The entropy encoding (decoding) process is ignored in this discussion since it is an information lossless process. The following steps are involved:

1. The image to be encoded is partitioned into contiguous $8 \times 8$-pixel square blocks;
2. Each block is transformed by the DCT given by

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{j=0}^{7} \sum_{k=0}^{7} f(j,k) \cos\left[\frac{(2j+1)u\pi}{7}\right] \cos\left[\frac{(2k+1)v\pi}{16}\right], \tag{7}$$

where $C(u) = 0$ for $u = 0$, and $C(u) = 1$ otherwise, and likewise for $C(v)$. Note that the DCT produces an array of 64 coefficients $F(u, v), u = 0, 1, \cdots, 7$; $v = 0, 1, \cdots, 7$;

3. The DCT coefficients are normalized using a user-defined normalization array $Q(u, v)$, which is the same for all blocks. The elements in the normalization array determine the quantization step size; Larger values correspond to larger quantization steps. The *human visual system* (HVS) contrast sensitivity function can be used as a guide in determining the elements of $Q(u, v)$. Thus, components in $F(u, v)$ that are important from a perceptual point of view would be weighted by the corresponding components in $Q(u, v)$ in a fashion that would enhance that particular component with respect to others;
4. The normalized and quantized DCT value is given by

$$\tilde{F}(u,v) = \text{round}\left[\frac{F(u,v)}{Q(u,v)}\right], \tag{8}$$

where *round* is short for rounding to the nearest integer. The rounding step is information *lossy*;

5. The set of quantized DCT coefficients $\{\tilde{F}(u, v)\}$ are further encoded using a *lossless* encoding algorithm such as Huffman encoding. As stated earlier, we shall ignore this step since the Huffman decoding at the receiver identically reproduces the coefficients of Equation (8);
6. At the receiver, after Huffman decoding, the received coefficients are "dequantized" by the operation

$$\hat{F}(u,v) = \tilde{F}(u,v)Q(u,v), \tag{9}$$

for $u=0, 1, \cdots, 7$; $v=0, 1, \cdots, 7$, and for each block;

7. The dequantized coefficients $\{\hat{F}(u, v)\}$ for each block are used to construct a subimage that is one block of the reconstructed image. The subimage is obtained via the inverse block DCT (BDCT)

$$\hat{f}(j,k) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)C(v)\hat{F}(u,v) \cos\left[\frac{(2j+1)u\pi}{16}\right] \cos\left[\frac{(2k+1)v\pi}{16}\right], \tag{10}$$

for $j=0, 1, \cdots, 7$; $k=0, 1, \cdots, 7$. The blocks are then reassembled in their respective order and the entire image is thus available for display.

As we can see, in both the encoding and decoding processes, the image is processed on a block-by-block basis. The existing correlation among the neighboring blocks is never taken into consideration in either of the two processes. Hence, the decoded image exhibits discontinuities at the block boundaries, which are known as blocking artifacts. As an example, Fig. 3A shows the $512 \times 512$ gray-scale "Lena" image, of which only the center $256 \times 256$ section is shown for clarity; Fig. 3B shows the same image when compressed using the JPEG algorithm at 0.24 bits per pixel (bpp). The blocking artifacts are clearly visible in this image.

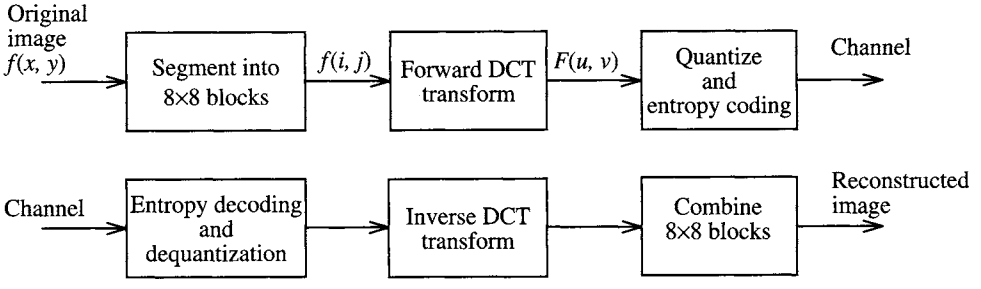


**FIGURE 2** The JPEG discrete consine transform encoding and decoding process.

(A) (B)

**FIGURE 3** Example of compression artifacts: (**A**) original Lena image and (**B**) JPEG compressed image at 0.24 bits per pixel.

## 3.3 Vector Notation

For simplicity, in the rest of the chapter we treat an $N \times N$ digital image as an $N^2 \times 1$ vector $\mathbf{f}$ by scanning either along rows or columns. This is called lexicographic ordering (by row or column) and the resulting vector can be viewed as a vector in the Euclidean space $R^{N^2}$.

Similarly, the corresponding BDCT coefficients of $\mathbf{f}$, obtained using Equation (7), can be arranged as a vector $\mathbf{F}$ in $R^{N^2}$. Thus, the BDCT functions as a linear mapping from $R^{N^2}$ to $R^{N^2}$. Let $B$ denote the corresponding matrix of this transformation. That is,

$$\mathbf{F} = B\mathbf{f}. \tag{11}$$

Due to the unitary property of the two-dimensional (2D) DCT transform for each block, the BDCT matrix is also unitary and the inverse transform satisfies $B^{-1} = B^T$. Then the inverse DBCT can be written as

$$\mathbf{f} = B^T\mathbf{F}. \tag{12}$$

As already stated, in JPEG coding, each DCT coefficient is quantized according to Equation (8). This quantization process can be described mathematically by an operator, say $Q$, from $R^{N^2}$ to $R^{N^2}$ such that

$$\tilde{\mathbf{F}} = Q[\mathbf{F}]. \tag{13}$$

Using this notation, the whole process from the original image $\mathbf{f}$ to the compressed data $\tilde{\mathbf{F}}$ can be conveniently written as

$$\tilde{\mathbf{F}} = Q[B\mathbf{f}]. \tag{14}$$

In the following, we let $\tilde{T}$ denote the concatenation of $B$ and $Q$ for additional simplicity. Then, Equation (14) can be rewritten as

$$\tilde{\mathbf{F}} = \tilde{T}[\mathbf{f}]. \tag{15}$$

With the above notation, the image decoding problem can be stated as the following: Given the data $\tilde{\mathbf{F}}$, find an $\mathbf{f}$ that satisfies Equation (15). A seemingly straightforward approach would be to solve for $\mathbf{f}$ from Equation (15) by directly inverting $\tilde{T}$. Unfortunately, due to the *many-to-one* nature of $Q$, $\tilde{T}$ is also a many-to-one mapping. Therefore, Equation (15) is not readily invertible. In general, there are many images that satisfy Equation (15). This is the basis of the projection method that we describe below.

## 3.4 Constraint Set Based on Transform Coefficients

From Equation (15) we can define a fundamental constraint set as follows:

$$C_T' = \left\{\mathbf{f} \in R^{N^2}\colon \tilde{T}[\mathbf{f}] = \tilde{\mathbf{F}}\right\}. \tag{16}$$

In other words, set $C_T'$ consists of *all* possible images that can identically reproduce the received compressed data. Equivalently, $C_T'$ can be written as

$$C_T' = \left\{\mathbf{f} \in R^{N^2}\colon (Q[B\mathbf{f}])_n = \tilde{F}_n,\ n \in \mathrm{I}\right\}, \tag{17}$$

where $\mathbf{I}$ is the index set of all BDCT coefficients, and $\tilde{F}_n$ denotes the $n$th component of the vector $\tilde{\mathbf{F}}$. Clearly, $C_T'$ effectively defines all the received information (i.e., all the quantized BDCT coefficients) about the unknown source image.

In general, $C_T'$ is not a closed set in $R^{N^2}$ owing to the fact that the quantization intervals associated with the coefficients $\tilde{F}_n$ are not closed. Instead, we use the closure of this set by adding the end points to these quantization intervals. The resulting set is then defined as follows:

$$C_T = \left\{\mathbf{f} \in R^{N^2}: F_n^{\min} \le (B\mathbf{f})_n \le F_n^{\max},\ n \in \mathbf{I}\right\}, \tag{18}$$

where $F_n^{\min}$, $F_n^{\max}$ are the end points of the quantization interval determined from the knowledge of $\tilde{F}_n$.

It is easy to show that $C_T$ is convex. The projection onto $C_T$ can be derived as follows: Consider a vector $\mathbf{g} \notin C_T$. Since BDCT is unitary, for any point $\mathbf{f} \in C_T$ we have

$$\|\mathbf{g} - \mathbf{f}\|^2 = \|B\mathbf{g} - B\mathbf{f}\|^2 = \sum_{n \in \mathbf{I}} [(B\mathbf{g})_n - (B\mathbf{f})_n]^2. \tag{19}$$

Thus, $\|\mathbf{g} - \mathbf{f}\|$ is minimized when $\mathbf{f}$ satisfies the following:

$$(B\mathbf{f})_n = \begin{cases} F_n^{\min} & \text{if } (B\mathbf{g})_n < F_n^{\min} \\ F_n^{\max} & \text{if } (B\mathbf{g})_n > F_n^{\max} \\ (B\mathbf{g})_n & \text{otherwise.} \end{cases} \tag{20}$$

for each $n$. Therefore, the projection of $\mathbf{g}$ is computed as $P_T\mathbf{g} = B^T\mathbf{F}$, where the components of $\mathbf{F}$ are determined according to Equation (20).

## 3.5 Constraints Based on Prior Knowledge

As stated earlier, during the encoding process a source image is first divided into nonoverlapping blocks, which are then transformed and quantized independently. This creates artificial discontinuities across neighboring blocks in the resulting image, referred to as blocking artifacts. Consequently, we need to introduce prior knowledge into the decoding process to restore the continuity in the image. That is, we need to enforce some degree of smoothness across neighboring blocks. As in all POCS algorithms, this prior knowledge must be described in the form of a constraint set, which is the creative part of a POCS method. While there might be a number of possibilities, one particularly effective technique is to directly bound the total variation across the block boundaries of the image.

Below we describe the variation separately for image columns and rows at the block boundaries. For convenience, we denote the vectors formed by the different columns of an $N \times N$ image $\mathbf{f}$ as $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$. We introduce a linear operator $Q_c$ such that $Q_c\mathbf{f}$ yields the difference between neighboring columns at the block boundaries of $\mathbf{f}$. For example, for the case of $N = 512$ and $8 \times 8$ blocks,

$$Q_c\mathbf{f} = \begin{bmatrix} \mathbf{f}_8 - \mathbf{f}_9 \\ \mathbf{f}_{16} - \mathbf{f}_{17} \\ \vdots \\ \mathbf{f}_{504} - \mathbf{f}_{505} \end{bmatrix}. \tag{21}$$

Therefore the norm of $Q_c\mathbf{f}$

$$\|Q_c\mathbf{f}\| = \left[\sum_{i=1}^{63} \|\mathbf{f}_{8\cdot i} - \mathbf{f}_{8\cdot i+1}\|^2\right]^{\frac{1}{2}} \tag{22}$$

is a measure of the total variation in image intensity between the boundary columns of adjacent blocks. For clarity, the operator $Q_c$ is illustrated in Fig. 4.

Accordingly, we introduce the following constraint set

$$C_c = \left\{\mathbf{f} \in R^{N^2}:\ \|Q_c\mathbf{f}\| \le E\right\}, \tag{23}$$

where $E$ is a scalar upper bound that defines the allowable amount of total variation of the boundary columns. The choice of $E$ is discussed later in the recovery algorithm section.

It is easy to demonstrate that the set $C_c$ is closed and convex. The projection $P_c\mathbf{g}$ of a vector $\mathbf{g} \notin C_c$ onto $C_c$ is computed as follows: Let $\{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_N\}$ denote the column vectors of $\mathbf{g}$. Then the column vectors $\{\mathbf{g}_1^*, \mathbf{g}_2^*, \cdots, \mathbf{g}_N^*\}$ of $P_c\mathbf{g}$ are given by

$$\mathbf{g}_i^* = \begin{cases} \alpha\mathbf{g}_i + (1-\alpha)\mathbf{g}_{i+1} & \text{for } i = 8, 16, 24, \cdots \\ (1-\alpha)\mathbf{g}_{i-1} + \alpha\mathbf{g}_i & \text{for } i = 9, 17, 25, \cdots \\ \mathbf{g}_i & \text{otherwise,} \end{cases} \tag{24}$$

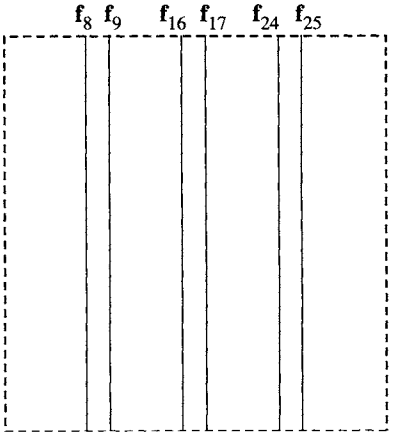


FIGURE 4 Reducing blocking artifacts by enforcing continuity between neighboring columns at block boundaries.

where $\alpha$ is a constant computed as

$$\alpha = \frac{1}{2}\left[1 + \frac{E}{\|Q_c \mathbf{g}\|}\right]. \tag{25}$$

Clearly, $\frac{1}{2} < \alpha < 1$ for $\mathbf{g} \notin C_c$.

Note that the projection in Equation (24) is remarkably simple. It modifies only the columns at the block boundaries, and leaves the rest of the columns unchanged.

The projection in Equation (24) is derived as follows: We seek $\mathbf{f} \in C_c$ such that $\|\mathbf{g} - \mathbf{f}\|$ is minimized. Toward this goal, we construct the Lagrange functional

$$J(\mathbf{f}) = \|\mathbf{f} - \mathbf{g}\|^2 + \lambda\left[\|Q_c \mathbf{f}\|^2 - E^2\right]. \tag{26}$$

With $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$, $\{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_N\}$ denoting the columns of $\mathbf{f}$ and $\mathbf{g}$, respectively, the Lagrange functional becomes

$$J(\mathbf{f}) = \sum_{j=1}^{N} \|\mathbf{f}_j - \mathbf{g}_j\|^2 + \lambda\left[\sum_{i=1}^{\lfloor (N-1)/8 \rfloor} \|\mathbf{f}_{8\cdot i} - \mathbf{f}_{8\cdot i+1}\|^2 - E^2\right], \tag{27}$$

where $\lfloor x \rfloor$ denote the largest integer no larger than $x$.

Setting the partial derivatives $\frac{\partial}{\partial \mathbf{f}_j} J(\mathbf{f}) = 0$, we obtain

$$\begin{aligned} \mathbf{f}_{8\cdot i} - \mathbf{g}_{8\cdot i} + \lambda(\mathbf{f}_{8\cdot i} - \mathbf{f}_{8\cdot i+1}) &= \mathbf{0} \\ \mathbf{f}_{8\cdot i+1} - \mathbf{g}_{8\cdot i+1} - \lambda(\mathbf{f}_{8\cdot i} - \mathbf{f}_{8\cdot i+1}) &= \mathbf{0} \end{aligned} \tag{28}$$

for $i = 1, 2, \cdots, \lfloor (N-1)/8 \rfloor$, and $\mathbf{f}_j - \mathbf{g}_j = \mathbf{0}$ for the rest of the columns $j$.

From Equation (28) we can easily obtain

$$\begin{aligned} \mathbf{f}_{8\cdot i} &= \alpha \mathbf{g}_{8\cdot i} + (1-\alpha)\mathbf{g}_{8\cdot i+1}, \text{ and} \\ \mathbf{f}_{8\cdot i+1} &= (1-\alpha)\mathbf{g}_{8\cdot i} + \alpha \mathbf{g}_{8\cdot i+1}, \end{aligned} \tag{29}$$

where $\alpha \triangleq \frac{1+\lambda}{1+2\lambda}$. The constant $\alpha$ is then solved from the constraint that $\|Q_c \mathbf{f}\|^2 = E^2$.

In a similar fashion, a set $C_r$ (subscript $r$ is for *row*) and projector $P_r$, that captures the intensity variations between the neighboring rows of adjacent blocks can be defined. For brevity, the details are omitted here.

Besides the sets previously defined, another useful set is the set that captures the information about the range of the pixel values of an image. For example, for an 8-bit $N \times N$ gray-scale image, this set is defined by

$$C_a = \left\{\mathbf{f} \in R^{N^2} : 0 \le f(i,j) \le 255,\ 1 \le i, j \le N\right\}, \tag{30}$$

where the subscript $a$ is a reminder that it is a constraint upon the *amplitude* of the image. This set is closed and convex. The projection of a vector $\mathbf{g}$ onto $C_a$ is given by $\mathbf{g}^* = P_a \mathbf{g}$ with its components defined by

$$g^*(i,j) = \begin{cases} 0, & \text{if } g(i,j) < 0 \\ 255, & \text{if } g(i,j) > 255, \\ g(i,j), & \text{otherwise.} \end{cases} \tag{31}$$

Let us now summarize the various constraint sets introduced above:

- $C_T$, the constraint set on DCT coefficients, with projector $P_T$;
- $C_c$, the constraint set on the total variation at the *column* boundaries, with projector $P_c$;
- $C_r$, the constraint set on the total variation at the *row* boundaries, with projector $P_r$;
- $C_a$, the constraint set on the allowable range of pixel amplitudes of the image, with projector $P_a$.

Using these convex sets, a POCS-based recovery algorithm can be defined in the decoder to reconstruct the compressed images.

## 3.6 Recovery Algorithm

The perceptive reader might have conjectured that by reducing the discontinuity across the block boundary, new discontinuities might be introduced in the columns (rows) adjacent to the original boundary columns (rows). That is, as the discontinuities at the block boundaries are ameliorated by projections, secondary discontinuities might be produced that "propagate" away from the boundary. The solution to this problem is to also define continuity constraints similar to $C_c$ ($C_r$) for pairs of adjacent columns (rows) that are away from the block boundaries. These constraints are repeated until all eight columns (rows) are ameliorated. We use the symbol $\tilde{P}_c(\tilde{P}_r)$ to denote the composition projector that sequentially reduces both the blocking artifacts at the boundary columns (rows) and the induced column (row) discontinuities within the block.

With $\tilde{P}_c$ and $\tilde{P}_r$ defined as above, the POCS recovery algorithm can be described as follows:

- Set $\mathbf{f}_0 = \hat{\mathbf{f}}$, where $\hat{\mathbf{f}}$ is the JPEG reconstructed image by the inverse BDCT matrix, i.e., $\hat{\mathbf{f}} = B^T \hat{\mathbf{F}}$;
- For $k = 1, 2, \cdots$, compute $\mathbf{f}_k$ from

$$\mathbf{f}_k = P_a \tilde{P}_c \tilde{P}_r P_T \mathbf{f}_{k-1}; \tag{32}$$

- Continue the iteration until $\|\mathbf{f}_k - \mathbf{f}_{k-1}\|$ is less than some prescribe threshold value.

To be complete, recall that the definition of the constraint set $C_c$ involves the constant upper bound $E$. In our experiments [16], it was determined from the JPEG reconstructed image $\hat{\mathbf{f}}$ in the following fashion: with $\{\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \cdots, \hat{\mathbf{f}}_N\}$ denoting the columns of $\hat{\mathbf{f}}$,

$$E = \frac{1}{7}\sum_{k=1}^{7}\left[\sum_{i=1}^{\lfloor (N-1)/8 \rfloor}\left\|\hat{\mathbf{f}}_{8i+k} - \hat{\mathbf{f}}_{8i+k+1}\right\|^2\right]^{\frac{1}{2}}. \quad (33)$$

This is the average variation between pairs of adjacent columns that are inside the coding blocks. Of course, one may further reduce this upper bound so that a smoother image can be recovered. In a similar fashion, the upper bound for the constraint set $C_r$ can be determined from pairs of adjacent rows in $\hat{\mathbf{f}}$.

As an application example, Fig. 5 shows the recovered image using this algorithm from the compressed data shown earlier in Fig. 3B. As can be seen, blocking artifacts have been greatly suppressed in this image. As an objective distance measure between a recovered image $\mathbf{g}$ and the original source image $\mathbf{f}$, the peak signal-to-noise ratio (PSNR) was used. For $N \times N$ images with [0, 255] gray-level range, PSNR is defined as

$$\text{PSNR} = 10\log_{10}\frac{255^2 \times N^2}{\|\mathbf{g} - \mathbf{f}\|^2}. \quad (34)$$

The PSNR of the POCS recovered image in Fig. 5 is 30.30 dB. In contrast, the PSNR of the JPEG image in Fig. 3B is 29.58 dB. Thus, the image recovered by the POCS algorithm is also closer to the original source image.

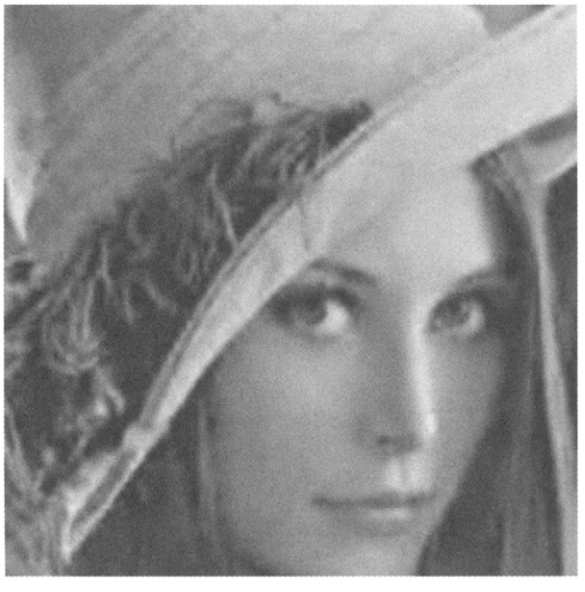

**FIGURE 5** Projection onto covex sets recovery of the compressed Lena image, peak signal-to-noise ratio = 30.30 dB.

Finally, it was found in our experiments that the POCS algorithm converged very quickly, typically requiring less than ten iterations. A detailed analysis on the computational cost of this algorithm can be found in [17], which concludes that for $512 \times 512$ images with $8 \times 8$ DCT blocks, the POCS algorithm typically requires 10 to 17 times the computations of a conventional JPEG decoder without any postprocessing.

## 3.7 Adaptive Processing

While the constraint set $C_c$ defined in Equation (23) effectively suppresses discontinuities between neighboring blocks, the image pixels at the block boundaries are processed in a uniform fashion by the projection in Equation (24). In practice, it might be more desirable to use spatially adaptive processing for the following reasons. First, the statistics of a natural image are typically nonstationary in the spatial domain, and it would be beneficial to exploit the local statistics of the image by a processing algorithm. Second, due to the frequency masking effect, compression artifacts in an image are not perceived uniformly by the HVS. For example, blocking artifacts in a smooth region are typically more visible than that in a texture (or edge) region in an image (e.g., Fig. 3). Thus, it becomes challenging to further incorporate this prior knowledge into the POCS recovery algorithm.

Toward this end, we modify the definition of the constraint set $C_c$ as follows:

$$C_{wc} = \left\{\mathbf{f} \in R^{N^2}: \|WQ_c\mathbf{f}\| \le E\right\}, \quad (35)$$

where $W$ is a diagonal matrix applied to the difference term $Q_c\mathbf{f}$ at the block boundaries. For a $512 \times 512$ image and $8 \times 8$ DCT blocks, $Q_c\mathbf{f}$ is a $(512 \cdot 63) \times 1$ vector, and correspondingly, $W = diag(w_2, w_2, \cdots, w_{512 \cdot 63})$, where diagonal elements $w_i$, $i = 1, 2, \cdots, 512 \cdot 63$ are defined according to the local statistics of the image. Consequently, the quantity $\|WQ_c\mathbf{f}\|$ corresponds to a weighted version of the between-block variation adjusted by the image statistics.

To better understand how the weights $w_i$ in $W$ are defined, let's first examine the projection onto the new set $C_{wc}$. It is straightforward to show that $C_{wc}$ is both convex and closed. For convenience, assume $512 \times 512$ images and $8 \times 8$ DCT blocks. Furthermore, we rewrite the diagonal matrix $W$ in a block form as

$$W = \begin{bmatrix} \mathbf{w}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{w}_{63} \end{bmatrix}, \quad (36)$$

where each sub-matrix $\mathbf{w}_k$, $k=1,2,\cdots,63$, is written in the form

$$\mathbf{w}_k = \begin{bmatrix} w_1^k & 0 & \cdots & 0 \\ 0 & w_2^k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{512}^k \end{bmatrix}. \tag{37}$$

That is, the diagonal elements in $\mathbf{w}_k$ are related to those in $W$ by the relation

$$w_j^k = w_{512\cdot(k-1)+j}, \text{ for } j = 1,2,\cdots,512. \tag{38}$$

Accordingly, the weighted variation term $\|WQ_c\mathbf{f}\|$ can be written as $\|WQ_c\mathbf{f}\| = \left[\sum_{k=1}^{63} \|\mathbf{w}_k(\mathbf{f}_{8\cdot k} - \mathbf{f}_{8\cdot k+1})\|^2\right]^{\frac{1}{2}}$, with $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$ denoting the column vectors of $\mathbf{f}$.

Consider an image vector $\mathbf{g} \notin C_c$. Its projection $P_{wc}\mathbf{g}$ onto $C_{wc}$ is derived as follows: We seek $\mathbf{f} \in C_{wc}$ such that $\|\mathbf{g} - \mathbf{f}\|$ is minimized, for which we construct the Lagrange functional

$$J(\mathbf{f}) = \|\mathbf{f} - \mathbf{g}\|^2 + \lambda\left[\|WQ_c\mathbf{f}\|^2 - E^2\right]. \tag{39}$$

Omitting the derivation details, we can write the columns $\{\mathbf{g}_1^*, \mathbf{g}_2^*, \cdots, \mathbf{g}_N^*\}$ of the resulting projection $P_{wc}\mathbf{g}$ as

$$\begin{aligned} \mathbf{g}_{8\cdot k}^* &= \frac{1}{2}(\mathbf{g}_{8\cdot k} + \mathbf{g}_{8\cdot k+1}) + \frac{1}{2}(1 + 2\lambda\mathbf{w}_k^T\mathbf{w}_k)^{-1}(\mathbf{g}_{8\cdot k} - \mathbf{g}_{8\cdot k+1}) \\ \mathbf{g}_{8\cdot k+1}^* &= \frac{1}{2}(\mathbf{g}_{8\cdot k} + \mathbf{g}_{8\cdot k+1}) - \frac{1}{2}(1 + 2\lambda\mathbf{w}_k^T\mathbf{w}_k)^{-1}(\mathbf{g}_{8\cdot k} - \mathbf{g}_{8\cdot k+1}), \end{aligned} \tag{40}$$

for $k=1,2,\cdots,63$; and the rest of the columns of $\mathbf{g}$ remain unchanged. The constant $\lambda$ is determined from the condition that $\|WQ_c\mathbf{g}^*\| = E$, which we will further discuss below in detail.

The spatially adaptive nature of the projector $P_{wc}$ can be readily seen from Equation (40). Indeed, the matrix term $(1 + 2\lambda\mathbf{w}_k^T\mathbf{w}_k)^{-1}$ can be written as

$$\begin{bmatrix} \frac{1}{1+2\lambda(w_1^k)^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{1+2\lambda(w_2^k)^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{1+2\lambda(w_{512}^k)^2} \end{bmatrix}. \tag{41}$$

Thus, the difference between the neighboring pixels at the boundaries will be reduced more in areas where the weighting factors $w_j^k$ (i.e., $w_i$) are large than in areas where they are small. In particular, consider the following two extreme cases: when $w_j^k = \infty$, the corresponding neighboring pixels after projection will become simply their average, which corresponds to maximal smoothing; on the other hand, when $w_j^k = 0$, the corresponding pixels will remain unchanged, hence no smoothing occurs.

## 3.8 Choice of Weight Factors in $W$ for Adaptive Processing

As discussed above, the choice of the weight factors in $W$ should reflect the local statistics of the image and the perceptual properties of the HVS. For an image the intensity at pixel location $(i, j)$ can be modeled as a random variable with mean $\mu_{i,j}$ and variance $\sigma_{i,j}^2$. The mean serves as a measure of the local brightness, and the variance serves as a measure of the local detail at location $(i, j)$.

Thus, from the operation of the projector $P_{wc}$, the weights should be a decreasing function of the local variance $\sigma_{i,j}^2$. An example of such a function is

$$w(i, j) = \frac{1}{1 + \sigma_{i,j}}, \tag{42}$$

where 1 is added to the denominator to avoid mathematical anomalies when $\sigma_{i,j}^2 = 0$. A range-compressed form of this function is given by

$$w(i, j) = \ln\left[1 + \frac{1}{1 + \sigma_{i,j}}\right]. \tag{43}$$

In our experiments [17], we noticed that the blocking artifacts were more visible in bright than in dark areas when the image was viewed on a monitor. A function which captures this property is

$$w(i, j) = \ln\left[1 + \frac{\sqrt{\mu_{i,j}}}{1 + \sigma_{i,j}}\right]. \tag{44}$$

We acknowledge that an exact form for the weight factors is difficult to determine as it is likely dependent on the medium and viewing conditions under which the images are presented. Nevertheless, the functions given above were found to yield improved results for image recovery from compressed data.

## 3.9 Determination of Constant $\lambda$ in the Projector $P_{wc}$

As stated previously, the constant $\lambda$ is solved from the condition $\|WQ_c\mathbf{g}^*\| = E$, or equivalently, $\|WQ_c\mathbf{g}^*\|^2 = E^2$. From Equation (40) we obtain

$$\mathbf{g}^*_{8\cdot k} - \mathbf{g}^*_{8\cdot k+1} = (1 + 2\lambda \mathbf{w}_k^T \mathbf{w}_k)^{-1}(\mathbf{g}_{8\cdot k} - \mathbf{g}_{8\cdot k+1}). \quad (45)$$

Thus, we have

$$Q_c\mathbf{g}^* = (1 + 2\lambda W^T W)^{-1} Q_c\mathbf{g}. \quad (46)$$

Letting $d_i$, $i = 1, 2, \cdots, 512\cdot 63$, denote the components of the vector $Q_c\mathbf{g}$, we can write the condition $\|WQ_c\mathbf{g}^*\|^2 = E^2$ as

$$\sum_{i=1}^{512\cdot 63} \frac{w_i^2 d_i^2}{(1 + 2\lambda w_i^2)^2} = E^2. \quad (47)$$

Clearly, Equation (47) can have many roots in $\lambda$. Thus, the immediate question that arises is which of these roots corresponds to the actual projection in Equation (40). It has been shown that [17] Equation (47) has one and only one root $\lambda^*$ that is positive, and that it is this positive root that gives the projection in Equation (40).

The next question is how to find this positive root $\lambda^*$. Clearly, numeric methods have to be applied. In particular, if we let $\Phi(\lambda) = \sum_{i=1}^{512\cdot 63} \frac{w_i^2 d_i^2}{(1+2\lambda w_i^2)^2} - E^2$, it can be further shown [17] that the iterates generated by the Newton's method

$$\lambda_{k+1} = \lambda_k - \Phi(\lambda_k)/\Phi'(\lambda_k), \quad k = 0, 1, 2, \cdots \quad (48)$$

with $\lambda_0 = 0$ will always converge to the positive root $\lambda^*$ in the following manner: $\lambda_{k+1} > \lambda_k$ and $|\lambda_{k+1} - \lambda^*| < |\lambda_k - \lambda^*|$. In experiments, we found that the algorithm in (47) converged rather quickly. Typically only a few iterations would yield adequate accuracy.

## 3.10 The Recovery Algorithm

So far we have focused on the set $C_{wc}$, which is modified from the original set $C_c$ such that it enforces spatially adaptive continuity between the neighboring *columns* of adjacent blocks in an image. Similarly, we can define a set $C_{wr}$ for the neighboring *rows* of adjacent blocks. Furthermore, these constraints can also be extended to pairs of adjacent columns (rows) that are away from the block boundaries in order to suppress any secondary discontinuities that might arise from the projections. These constraints are repeated until all eight columns (rows) are considered. Having defined all these new constraint sets, they are substituted into the POCS recovery algorithm previously defined in Equation (32).

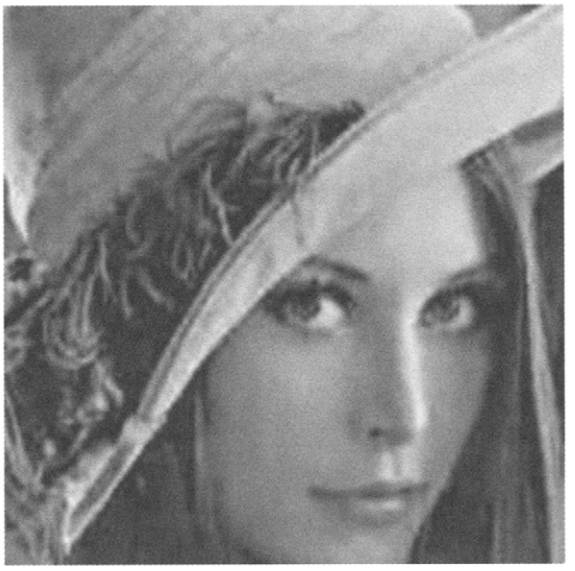

**FIGURE 6** Projection onto complex sets recovery of the compressed Lena image using adaptive processing, peak signal-to-noise ratio = 30.43 dB.

As an example, we show in Fig. 6 the recovered image using these spatially adaptive constraint sets from the same compressed data shown earlier in Fig. 3B. Compared to the previous result in Fig. 6, most of the high-frequency features such as textures and edges of the image are notably better preserved in Fig. 5. In addition, the PSNR of this image is 30.43 dB, which is also slightly higher than that of the image in Fig. 6. In our experiments, Equation (43) was used for determining the weight factors, for which the local image mean and variance were estimated directly from the compressed data. Interested readers are referred to [17] for more details.

# 4 Maximum A Posteriori Estimation for Postprocessing of Compressed Images

Let us denote by vector $\mathbf{g}$ and $\mathbf{f}$ the observed and the unknown image is treated as a deterministic quantity, the maximum likelihood (ML) is commonly used to obtain an estimate of it. In this case, the probability density function or likelihood function $p(\mathbf{g}/\mathbf{f})$ is formed. The ML estimator of $\mathbf{f}$ is obtained by maximizing either the likelihood, that is, $\mathbf{f}_{ML} = \arg\max_f\{p(\mathbf{g}/\mathbf{f}, \boldsymbol{\theta})\}$, or the log likelihood function, that is, $\mathbf{f}_{ML} \stackrel{f}{=} \arg\max_f\{\log[p(\mathbf{g}/\mathbf{f}, \boldsymbol{\theta})]\}$. The implicit assumption here is that the unknown parameters $\boldsymbol{\theta}$ ?? The ML estimator has been used extensively and has many desirable properties

(see, for example, [93] for more details). Its main shortcoming, however, is that it is based exclusively on the observation-unknowns relationship and does not allow for the incorporation of *prior knowledge* about the unknowns which are treated as deterministic parameters.

Frequently in many problems of interest there arise situations where from the observations $\mathbf{g}$ alone one cannot define precisely the unknowns. In other words, the observations-to-unknowns relation is one-to-many. This is also the case in the problem of inferring the original image from the compressed data. In other words, due to quantization as explained previously, there is an infinite number of images not necessarily similar in appearance that all satisfy the compressed data.

To ameliorate this difficulty, in MAP estimation $\mathbf{f}$ is modeled to be a sample of a random field. Prior knowledge about them is assumed in the form of the prior PDF $p(\mathbf{f})$, which is assumed known. The MAP estimator is obtained by maximizing the *posterior* according to $\mathbf{f}_{MAP} = \underset{f}{\text{argmax}}\{p(\mathbf{f}/\mathbf{g})\}$. Since the posterior is not known, Bayes rule is used and the posterior is written as $p(\mathbf{f}/\mathbf{g}) = \frac{p(\mathbf{g}/\mathbf{f})p(\mathbf{f})}{p(\mathbf{g})}$. Since the denominator of the posterior does not depend on $\mathbf{f}$ the MAP estimator is found as

$$\mathbf{f}_{MAP} = \arg\max_{f}\{p(\mathbf{f}/\mathbf{g})\} = \arg\max_{f}\{p(\mathbf{g}/\mathbf{f})p(\mathbf{f})\}. \quad (49)$$

If the log is used

$$\mathbf{f}_{MAP} = \arg\max_{f}\{\log(p(\mathbf{g}/\mathbf{f})) + \log(p(\mathbf{f}))\}. \quad (50)$$

Thus, to find the MAP estimator ones needs the following: (a) to define the likelihood function, that is, the probabilistic relationship between the observations and the unknowns; and (b) to define a prior on the unknowns to introduce desired *a priori* properties to the estimate of the unknowns.

## 4.1 Definition of Likelihood

Using the same notation as for the POCS methodology the original $N \times N$ digital image $\mathbf{f}$ that needs to be estimated is an $N^2 \times 1$ vector in an $R^{N^2}$ Euclidean space by scanning either along rows or columns. The observation can be either assumed to be the transmitted BDCT coefficients or the image reconstructed by them. If the observations $\mathbf{g}$ are assumed to be the BDCT coefficients they are given by

$$\mathbf{g} = Q[B\mathbf{f}]; \quad (51)$$

alternatively, if the observations are assumed to be the reconstructed image they are given by

$$\mathbf{g} = B^T Q[B\mathbf{f}]; \quad (52)$$

where $B$ and $Q$ represent the BDCT transform and the quantization operators, respectively.

The MAP formulation requires the definition of the likelihood $p(\mathbf{g}/\mathbf{f})$. Various definitions have been used for that. In [19], the BDCT coefficients are used as observations $\mathbf{g}$ and the likelihood is defined as

$$p(\mathbf{g}/\mathbf{f}) = \begin{cases} 1 & \text{if} \quad \mathbf{g} = Q[B\mathbf{f}] \\ 0 & \text{if} \quad \mathbf{g} \neq Q[B\mathbf{f}] \end{cases}. \quad (53)$$

Since

$$\log p(\mathbf{g}/\mathbf{f}) = \begin{cases} 0 & \text{if} \quad \mathbf{g} = Q[B\mathbf{f}] \\ -\infty & \text{if} \quad \mathbf{g} \neq Q[B\mathbf{f}] \end{cases} \quad (54)$$

the MAP estimator in Equation (50) can be written as the constrained optimization problem

$$\mathbf{f}_{MAP} = \arg\min_{f \in C_T}\{-\log(p(\mathbf{f}))\} \quad (55)$$

where $C_T$ is the set of images that compress to $\mathbf{g}$, [i.e., this set is identical to the set defined in Equation (18)].

## 4.2 Definition of Image Prior

Crucial to the success of the recovery algorithm is the definition of the prior $p(\mathbf{f})$ on the recovered image $\mathbf{f}$. In [19], a non-Gaussian Markov random field (MRF) model was used. Such models have been shown to successfully model both the smooth regions and discontinuities present in images. A Gibbs distribution is used to explicitly describe the distribution of MRFs. A Gibbs distribution is a distribution that can be expressed in the form

$$p(\mathbf{f}) = \frac{1}{Z}\exp\left(-\sum_{c \in C} V_c(\mathbf{f})\right) \quad (56)$$

where $Z$ is a normalizing constant, $V_c(\mathbf{f})$ is any function of a local group of points $c$ and $C$ is the set of all such local groups. More specifically the model used in this work has the form

$$p(\mathbf{f}) = \frac{1}{Z}\exp\left(-\frac{1}{\lambda}\sum_{c \in C} \rho_T(\mathbf{d}_c^T\mathbf{f})\right), \quad (57)$$

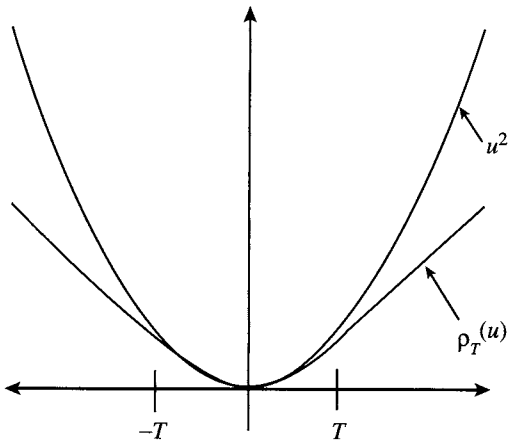


**FIGURE 7** Huber minimax function $\rho_c()$ superimposed on quadratic function.

where $\lambda$ is a positive scalar, $\mathbf{d}_c$ is a difference operator at location $c$, and $\rho_T(\cdot)$ is a function given by

$$\rho_T(u) = \begin{cases} u^2, & |u| \le T, \\ T^2 + 2T(|u| - T), & |u| > T. \end{cases} \tag{58}$$

The function $\rho_c()$ superimposed on a quadratic is shown in Fig. 7.

The difference operator $\mathbf{d}_c$ is defined as

$$\left[\mathbf{d}_c^T \mathbf{f}\right]_{c \in C} = \left[\mathbf{f}_{m,n} - \mathbf{f}_{k,l}\right] \text{ with } k, l \in \mathrm{N}_{m,n} \text{ and } 1 \le m, n \le N, \tag{59}$$

with $\mathrm{N}_{m,n}$ the nearest eight neighbors of pixel $(m,n)$ and $N$ the dimension of the image. Since $\rho_{T(u)}$ is convex, this particular form of the MRF results in a convex optimization problem when used in MAP estimation. Therefore, such MAP estimates will be unique and can be computed efficiently. The function $\rho_{T(u)}$ is known as the Huber minimax function [20] and for that reason this statistical model is called the Huber Markov random field (HMRF) model. The use of $\rho_{T(u)}$ in the prior does not penalize large differences as heavily as a quadratic function used in a Gaussian model. Thus, the generation of edges is "encouraged" more than in a Gaussian MRF model. Note also that the Gaussian model is a special case of the HMRF model.

## 4.3 Image Reconstruction Algorithm

The MAP estimate can now be written as

$$\begin{aligned} \mathbf{f}_{MAP} &= \arg\min_{f \in C_T} \left\{ \sum_{c \in C} V_c(\mathbf{f}) \right\} \\ &= \arg\min_{f \in C_T} \left\{ \sum_{1 \le m,n \le N} \sum_{k,l \in \mathrm{N}_{m,n}} \rho_T\left(\mathbf{f}_{m,n} - \mathbf{f}_{k,l}\right) \right\}. \end{aligned} \tag{60}$$

As a result of the choice of the image model, this results in a convex (but not quadratic) constrained optimization, which can be solved using iterative techniques. Since the set $C_T$ is convex, the "gradient projection" algorithm is known to converge to the constrained minimum of this problem [Mpertsekas]. Such an algorithm is given by

$$\begin{aligned} \mathbf{f}^{k+1} &= \tilde{\mathbf{f}}^k + a^k \mathbf{r}^k \\ \tilde{\mathbf{f}}^{k+1} &= P_T\left(\mathbf{f}^{k+1}\right), \end{aligned} \tag{61}$$

where the index $k$ denotes the iteration, the operator $P_T$ the projector onto the set $C_T$ given in Equation (20), the vector $\mathbf{r}^k$ the gradient with respect to $\mathbf{f}$ of the objective function $\sum_{c \in C} V_c(\mathbf{f})$, and the scalar $a^k$ is the step along the gradient. The gradient of the Huber function can be computed in a closed form in [19]. The choice of the step size is based on a quadratic approximation of the nonquadratic function in (60).

The standard Lena image compressed at 0.2644 bpp by the JPEG algorithm was used to demonstrate this algorithm. The result of JPEG decompression of this image is shown in Fig. 8A. The result of postprocessing this image is with the MAP algorithm previously presented shown in Fig. 7B. The number of iterations needed to sufficiently reduce the artifacts is dependent on the severity of the degradation. More iterations are required for convergence with more severe degradation. Notice that the blocking effects have been removed in the postprocessed image. This can be most easily seen in the shoulder and background regions. Notice that while the discontinuities due to the blocking effects have been smoothed the sharp discontinuities in the original image such as along the hat brim have been preserved.

# 5 Conclusions

In this chapter, we presented two methodologies that have been used for recovery-based postprocessing of compressed images. Since the purpose of this chapter was tutorial rather than a review, the methodologies presented here by no means exhaust the topic. However, they are representative of the work in this field. Other projection-based recovery methods have used different smoothness constraint sets to capture both local and global smoothness properties of images. Recovery methods based on Bayesian methodologies have also been applied to this problem. For example, in [25] the hierarchial Bayesian methodology has been applied for obtaining estimates of both the original image and the hyper parameters required to determine the utilized image and noise models. A review of the postprocessing of compressed images problem is presented in [26].

(A) (B)

**FIGURE 8** **A:** Compressed Lena at 0.2644 bpp, and **B:** postprocessed with T = 1.

## References


[1] L. G. Gubin, B. T. Polyak, and E. V. Raik, "The method of projections for finding the common point in convex sets," *USSR Comput. Math. Phys.* 7, 1–24 (1967).

[2] L. M. Bregman, "Finding the common point of convex sets by the method of successive projections," *Dokl. Akad. Nauk. USSR* 162, 487–490 (1965).

[3] I. Halperin, "The product of projection operators," *Acta Sci. Math.* 23, 96–99 (1960).

[4] Z. Opial, "Weak convergence of the sequence of successive approximation for nonexpansive mappings," *Bull. Am. Math. Soc.* 73, 591–597 (1967).

[5] J. von Neumann, *Functional Operators*, vol. II of *Ann. Math. Stud.*, vol. 22, no. 55, (Princeton University Press, Princeton, NJ, 1950).

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

[7] D. C. Youla and H. Webb, "Image restoration by the method of convex projections: Part 1 - theory," *IEEE Trans. Med. Imaging* 1, 81–94 (1982).

[8] Henry Stark, Ed., *Image Recovery: Theory and Application* (Academic Press New York, 1987).

[9] Henry Stark and Yongyi Yang, *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Networks, and Optics* (John Wiley & Sons, New York, 1998).

[10] M. Goldburg and R. J. Marks II, "Signal synthesis in the presence of inconsistent set of constraints," *IEEE Trans. Circuits Syst.* 332, 647–663 (1985).

[11] D. C. Youla and V. Velasco, "Extensions of a result on the synthesis of signals in the presence of inconsistent constraints," *IEEE Trans. Circuits Syst.* 33, 465–468 (1986).

[12] A. Levi and H. Stark, "Signal restoration from phase by projections onto convex sets," *J. Opt. Soc. Am.* 73, 810–822 (1983).

[13] G. Pierra, "Decomposition through formalization in a product space," *Math. Prog.* 28, 96–115 (1984).

[14] P. L. Combettes and H. Puh, "Iterations of parallel convex projections in Hilbert spaces," *Numer. Funct. Anal. Optim.* 15, 225–243 (1994).

[15] P. L. Combettes, "Inconsistent signal feasibility problems: least-squares solutions in a product space," *IEEE Trans. Signal Process.* 42, 2955–2966 (1994).

[16] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circ. Syst. Video Tech.* 3, 421–432 (1993).

[17] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially-adaptive reconstruction of block transform compressed images," *IEEE Trans. Image Process.* 4, 896–908 (1995).

[18] Y. Yang and N. P. Galatsanos, "Removal of compression artifacts using projections onto convex sets and line process modeling," *IEEE Trans. Image Process.* 6, 1345–1357 (1997).

[19] T. P. O'Rourke and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Trans. Circuits Syst. Video Technol.* 5, 490–499 (1995).

[20] P. J. Huber, *Robust Statistics* (John Wiley, New York, 1981).

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

[22] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Process.* 18, 36–58 (2001).

[23] A. K. Katsaggelos and N. Galatsanos, eds, *Signal Recovery Techniques for Image and Video Compression and Transmission* (Kluwer Academic, New York, 1998).

[24] D. P. Bertsekas, Convex Analysis and Optimization, Athena Scientific, New Hampshire, 2003.

[25] J. Mateos, A. K. Katsaggelos, and R. Molina, "A Bayesian approach for the estimation and transmission of regularization parameters for reducing blocking artifacts," *IEEE Trans. Image Process.* 9, 1200–1215 (2000).

[26] M. Shen and C. Kuo, "Review of post processing techniques for compression artifact removal," *J. Vis. Commun. Image Rep.* 9, 2–14 (1998).