Multichannel Regularized Recovery of Compressed Video Sequences

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

***Abstract*—In this paper, we propose a multichannel regularized recovery approach to ameliorate coding artifacts in compressed video. The major advantage of the proposed approach is that both temporal and spatial correlations in a video sequence can be exploited to complement the compressed video data. In particular, a temporal regularization term is introduced to enforce smoothness along the motion trajectories defined by the transmitted motion vectors for motion compensation. Several forms of temporal regularization with different computational complexity are considered. Based on the proposed approach, recovered images are obtained from the compressed data using the well-known gradient-projection algorithm. Moreover, an iterative algorithm is proposed for the determination of regularization parameters at the coder side. A number of numerical experiments using several H.261 and H.263 compressed streams are presented to evaluate the performance of the proposed recovery algorithms. Results from these experiments demonstrate that the use of temporal regularization can yield significant improvement in the quality of the recovered images—in terms of both visual evaluation and objective peak-signal-to-noise (PSNR) measure.**

I. Introduction

With the growing demand for efficient digital representation of images and video in a wide range of applications, image and video compression has attracted considerable attention in recent years. Evidence of this is the emergence of several international image and video coding standards such as JPEG [25], MPEG-1 [10], MPEG-2 [11], H.261 [12], H.263 [13], and most recently MPEG-4 [29]. What makes image and video compression possible is the fact that there exists a great deal of information redundancy in images and video, and every efficient coding scheme attempts to exploit this redundancy.

The information redundancy in a video sequence is divided into spatial redundancy and temporal redundancy. The spatial redundancy exhibits as the similarity of neighboring pixels in an image, whereas the temporal redundancy exhibits as similarity between neighboring frames in a video sequence. A popular video coding scheme is to use a transform to exploit the spatial redundancy and use motion-compensated prediction to take advantage of the temporal redundancy. For example, block discrete cosine transform (BDCT) and motion compensation are used in the existing video coding standards such as MPEG, H.261, and H.263.

An inherent problem with image and video compression is that it results in various undesirable coding artifacts, especially in low bit-rate applications. For example, coding artifacts such as "blocking," "ringing effects," and "mosquito noise" are well known to exist in JPEG and MPEG compressed images and video. In the literature, various postprocessing and recovery algorithms have been proposed to ameliorate these artifacts in compressed images. For example, in [26], [28], [19], [14], and [17], different filtering algorithms are used to reduce the artifacts in a compressed image, while in [27], [32], [33], [21][23], [18], and [34]–[37], image recovery approaches are proposed to reconstruct the compressed image so that it is free (or nearly free) of artifacts. The essence of these recovery algorithms is to reconstruct the image by taking advantage of, in addition to the received compressed data, the existence of strong correlation among the pixels of the image (i.e., spatial redundancy) either through assuming an underlying *a priori* probability model or by directly imposing a smoothness constraint on the image.

For the recovery of compressed video, a straightforward approach would be to treat each frame in a video sequence as an independent image and recover it separately from the other frames. However, such an approach is only suboptimal since it does not make use of the existence of strong interframe correlation (i.e., temporal redundancy) in a video sequence. It is well known from the work in other image sequence recovery problems that the incorporation of interframe correlation can lead to significant improvement in the quality of recovered images (see, for example, [3] and [24]. The work in [3] and [24] addressed restoration of blurred and noisy image sequences. In this paper, we address the problem of recovering compressed video sequences by utilizing both temporal and spatial correlations for the first time. The proposed approach is multichannel in nature in that, unlike other frame-based processing methods (see, for example, [9]), the image frames in a sequence are recovered jointly. The term *multichannel* is used here to reflect the fact that consecutive image frames can be viewed as observations of the same scene that are not identical (due to motion) but are strongly correlated [8]. Thus, consecutive image frames should be recovered simultaneously taking into account interframe correlations. The latter is enforced through smoothness along the motion trajectories defined by motion vectors that are readily available in the compressed data. Apart from a different problem being addressed, a major improvement of the work presented here over that in [3] is that a new form of temporal regularization is introduced that can accommodate general, subpixel accuracy motion. The approach in [3] was more restrictive and assumed integer accuracy motion. The proposed recovery algorithms are quite flexible and can be applied to all existing video coding standards such as MPEG and H.263.

Manuscript received May 2000; revised February 2001. This paper was recommended by Associate Editor V. Anastassopoulos.

M. G. Choi is with Actiontec Electronics Ltd, Sunnyvale, CA 94086 USA.

Y. Yang and N. P. Galatsanos are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA.

Publisher Item Identifier S 1057-7130(01)05229-6.

The rest of the paper is organized as follows: in Section II, we introduce some notation on video coding, particularly the coding scheme of block DCT and motion compensation. Then, video decoding is formulated as the solution to an ill-posed recovery problem using the principle of regularization. The regularization terms which characterize the spatial and temporal correlations are defined in Section III. The recovery algorithm is then derived in Section IV based on the gradient projection algorithm. A detailed analysis of the computational complexity of the proposed reconstruction algorithms is also presented in this section. In Section V, numerical results from the proposed recovery algorithms are presented, and issues regarding the implementation of the recovery algorithms are discussed. Moreover, a new approach for selecting the regularization parameters is presented in this section. Finally, conclusions are given in Section VI.

II. Background and Motivation

First, we review some basic notation in video coding. To keep focused, we will limit our discussion to motion-compensated transform coding schemes only. Notice that such coding schemes are currently widely used in existing coding standards such as MPEG, H.261, and H.263.

A. Motion-Compensated Video Coding

In motion-compensated video coding such as MPEG and H.263, the frames in a video sequence are broadly classified into intracoded frames and predictively coded frames. An intracoded frame, also known as an I-frame, is coded using a transform such as block DCT. A predictively coded frame is first predicted (either unidirectionally or bidirectionally) from the most recently reconstructed reference frame(s), and the prediction error is then further compressed by transform-based coding.

Some notation is due in order to quantitatively describe this coding process. Let $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L$ denote L frames of a video sequence, where $\mathbf{f}_l$ is a vector representation of frame l through a lexicographical ordering of its pixels. For frame $\mathbf{f}_l$, define

$$\bar{\mathbf{f}}_l \triangleq \begin{cases} \mathbf{f}_l, & \text{if } \mathbf{f}_l \text{ is intra coded} \\ \mathbf{f}_l - \hat{\mathbf{f}}_l, & \text{if } \mathbf{f}_l \text{ is predictively coded} \end{cases} \tag{1}$$

where $\hat{\mathbf{f}}_l$ denotes the motion-compensated prediction of the frame $\mathbf{f}_l$ from the most recently reconstructed reference frame(s). That is, $\bar{\mathbf{f}}_l$ is simply the motion-compensated error of the frame $\mathbf{f}_l$ when it is predictively coded. This prediction error $\bar{\mathbf{f}}_l$ is then further compressed by using transform-based coding (such as block DCT), denoted by T_B, followed by the quantizer, denoted by $\mathcal{Q}$, to obtain the compressed data, say $\bar{\mathbf{F}}_l$. In other words, $\bar{\mathbf{F}}_l$ is the quantized transform coefficients of $\bar{\mathbf{f}}_l$. In short, we have

$$\bar{\mathbf{F}}_l = \mathcal{Q} T_B \bar{\mathbf{f}}_l \tag{2}$$

for $l = 1, 2, \ldots, L$. The quantized data, along with the necessary motion information, are then entropy-coded and transmitted to the receiver.

B. Video Decoding as Video Recovery

The task at the receiver is to reconstruct the images $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L$ from the received data. Since quantization is typically a many-to-one mapping, i.e., the operator $\mathcal{Q}$ is not invertible, we can no longer determine exactly the original images $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L$ from the received quantized coefficients $\bar{\mathbf{F}}_1, \bar{\mathbf{F}}_2, \ldots, \bar{\mathbf{F}}_L$. In a classical decoder, these quantized coefficients are simply taken as the transform coefficients and operations opposite to the coding process are used to obtain the compressed images. That is, the frames $\mathbf{f}_l$, $l = 1, 2, \ldots, L$, are decoded as

$$\mathbf{g}_l \triangleq \begin{cases} T_B^{-1} \bar{\mathbf{F}}_l, & \text{if } \mathbf{f}_l \text{ is intra coded} \\ \hat{\mathbf{f}}_l + T_B^{-1} \bar{\mathbf{F}}_l, & \text{if } \mathbf{f}_l \text{ is predictively coded} \end{cases} \tag{3}$$

where T_B^{-1} denotes the inverse of the transform-based coding.

Since the quantized value of each coefficient specifies an interval that the exact value of that coefficient should belong to, the knowledge of the quantized coefficients $\bar{\mathbf{F}}_l$ defines the following set to which the frame $\mathbf{f}_l$ should belong [27], [32]:

$$C_l \triangleq \left\{ \mathbf{f}_l : \alpha_n^{\min} \le \left(T_B \bar{\mathbf{f}}_l \right)_n \le \alpha_n^{\max},\ n \in \mathcal{I} \right\} \tag{4}$$

where $(T_B \bar{\mathbf{f}}_l)_n$ is used to denote the n-th transform coefficient of $\bar{\mathbf{f}}_l$, $\alpha_n^{\min}$ and $\alpha_n^{\max}$ are the end-points of the quantization interval associated with this coefficient, and $\mathcal{I}$ denotes the index set of the transform coefficients in the frame.

Clearly, every element in C_l will result in the same quantized data as the original image $\mathbf{f}_l$ does. As a matter of fact, one can quickly verify that the decoded image $\mathbf{g}_l$ in (3) also belongs to C_l, yet it exhibits coding artifacts. Without additional knowledge, it is impossible to determine the original image exactly. Therefore, the recovery of the images $\mathbf{f}_l$, $l = 1, 2, \ldots, L$, from the received data becomes an ill-posed problem.

The method of regularization is proven effective for obtaining satisfactory solutions to ill-posed problems [30]. It has been applied successfully to solve a number of signal and image recovery problems. In particular, it is used in [32] for the reduction of artifacts in compressed images. According to this approach, an objective function is defined to consist of two terms whereby one term is used to express the fidelity of the solution to the available data and the other is to incorporate prior knowledge or impose desirable properties on the solution. The solution to the problem is then obtained through the minimization of this objective function. For the recovery of video from its compressed data using the method of regularization, the key is to define the objective function in such a way that its regularization terms can take advantage of, in addition to the received data, both the spatial domain and the temporal domain correlations in a video sequence. This will be the focus of the next section.

III. Multichannel Regularization Function

In multichannel signal recovery, the correlation properties between different channels of the signal are effectively exploited to recover the signal, see, for example [5], [6], [8], and [31]. An image sequence can be considered as a multichannel signal when the temporal dimension is viewed as an additional channel to the spatial dimension. In image sequence recovery problems,

it is well known that the use of both spatial smoothness and temporal smoothness along the direction of the motion yields better results than the use of spatial smoothness alone (see, for example, [3] and [1]). Consider L frames (channels) of a video sequence. Define

$$\mathbf{g}_M^{\mathrm{T}} \triangleq \left(\mathbf{g}_1^{\mathrm{T}}, \mathbf{g}_2^{\mathrm{T}}, \ldots, \mathbf{g}_L^{\mathrm{T}}\right)^{\mathrm{T}}, \quad \mathbf{f}_M^{\mathrm{T}} \triangleq \left(\mathbf{f}_1^{\mathrm{T}}, \mathbf{f}_2^{\mathrm{T}}, \ldots, \mathbf{f}_L^{\mathrm{T}}\right)^{\mathrm{T}} \quad (5)$$

where $\mathbf{g}_M$ and $\mathbf{f}_M$ denote the compressed images and the recovered images, respectively, the subscript M indicates the multichannel notation, and the supscript T denotes the transpose operation.

The existence of strong spatial and temporal correlations in $\mathbf{f}_M$ suggests the use of the following multichannel regularization functional for the recovery of $\mathbf{f}_M$ from its compressed data $\mathbf{g}_M$:

$$\mathbf{J}(\mathbf{f}_M) \triangleq \|\mathbf{g}_M - \mathbf{f}_M\|^2 + \lambda_s \mathbf{J}_s(\mathbf{f}_M) + \lambda_t \mathbf{J}_t(\mathbf{f}_M), \quad (6)$$

where three terms are involved, each of which is explained below.

- In the first term, $\|\cdot\|$ denotes the Euclidean norm. This term is used to enforce the fidelity of the recovered images $\mathbf{f}_M$ to the received data, which are represented in the spatial domain by the compressed image data $\mathbf{g}_M$.
- The constants λ_s and λ_t are called regularization parameters. Their role is to balance the influence of their associated regularization terms on the objective function.
- The term $\mathbf{J}_s(\mathbf{f}_M)$ in the objective function $\mathbf{J}(\mathbf{f}_M)$ in (6) is used to enforce spatial correlation in the recovered images. Note that the spatial smoothness property of images has been exploited in almost every proposed processing algorithm so far, ranging from simple filtering to the more sophisticated probability model-based approaches, for the reduction of compression artifacts, in [32]–[34], for example.
- The temporal domain correlation is exhibited as similarity between the neighboring frames in $\mathbf{f}_M$. The term $\mathbf{J}_t(\mathbf{f}_M)$ in (6) is defined to enforce smoothness along the motion trajectories in the frames of $\mathbf{f}_M$.

A. Spatial Domain Regularization

In block transform-based video coding such as in MPEG, H.261, and H.263, images are typically coded on a block-by-block basis, where a particular block is processed independent of the others. As a result, discontinuities occur at coding block boundaries. Indeed, it is observed in [32] that the local variation of the pixels at the coding block boundaries of an compressed image tends to be significantly larger than that of the pixels inside the coding blocks.

With this in mind, we define $\mathbf{J}_s(\mathbf{f}_M)$ to have the following form:

$$\mathbf{J}_s(\mathbf{f}_M) \triangleq \gamma \mathbf{J}_w(\mathbf{f}_M) + (1-\gamma)\mathbf{J}_b(\mathbf{f}_M) \quad (7)$$

where the first term $\mathbf{J}_w(\mathbf{f}_M)$ is used to enforce smoothness inside the coding blocks, and the second term $\mathbf{J}_b(\mathbf{f}_M)$ is used to enforce smoothness at the coding block boundaries. The constant γ in (7) is used to balance relative effectiveness of these two penalty terms.

1) Between-Block Regularization: The term $\mathbf{J}_b(\mathbf{f}_M)$ in (7) is based on the variation of an image at its coding block boundaries. It is further decomposed into two parts: the variation at the *vertical* block-boundaries and that at the *horizontal* block-boundaries. Specifically, for image frames $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L$ in $\mathbf{f}_M$, we have

$$\mathbf{J}_b(\mathbf{f}_M) \triangleq \sum_{l=1}^{L} \left(\|Q_{\mathrm{VB}}\mathbf{f}_l\|^2 + \|Q_{\mathrm{HB}}\mathbf{f}_l\|^2\right) \quad (8)$$

where Q_{VB}, Q_{HB} are operators that find the differences between adjacent columns and adjacent rows at the block boundaries of an image, respectively.

To clarify the point, assume that $\mathbf{f}$ is an $M \times N$ image with pixels $\mathbf{f}(i,j)$, $i = 1,2,\ldots, j = 1,2,\ldots,N$, and that a coding block size of 8×8 is used. Then the operator Q_{VB} is defined in the following fashion:

$$\left(Q_{\mathrm{VB}}\mathbf{f}\right)(i,j) \triangleq \begin{cases} \mathbf{f}(i,j) - \mathbf{f}(i,j+1), & \text{if } ((j))_8 = 0 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $(Q_{\mathrm{VB}}\mathbf{f})(i,j)$ denotes the value of $Q_{\mathrm{VB}}\mathbf{f}$ at (i,j), and $((j))_8$ denotes the value of j modulo 8. Note that $((j))_8 = 0$ only if the jth column of the image is at a coding block boundary. Then, we have

$$\|Q_{\mathrm{VB}}\mathbf{f}\|^2 = \sum_{i=1}^{M} \sum_{\substack{j=1 \\ ((j))_8=0}}^{N} \left[\mathbf{f}(i,j) - \mathbf{f}(i,j+1)\right]^2. \quad (10)$$

The quantity $\|Q_{\mathrm{VB}}\mathbf{f}\|^2$ captures the total variation between columns at all adjacent block boundaries of the image $\mathbf{f}$.

In a similar fashion, we can define the operator Q_{HB} to find the difference at the *horizontal* coding block boundaries of an image. Note that the operators Q_{VB} and Q_{HB} were used in [32] to define smoothness constraint sets to suppress the blocking artifact of a compressed image.

2) Within-Block Regularization: Similar to the definition of $\mathbf{J}_b(\mathbf{f}_M)$, the term $\mathbf{J}_w(\mathbf{f}_M)$ in (7) is based on the variation of an image inside its coding blocks. Specifically,

$$\mathbf{J}_w(\mathbf{f}_M) \triangleq \sum_{l=1}^{L} \left(\|Q_{\mathrm{VS}}\mathbf{f}_l\|^2 + \|Q_{\mathrm{HS}}\mathbf{f}_l\|^2\right) \quad (11)$$

where Q_{VS} and Q_{HS} are operators defined to find the differences between all the adjacent columns and adjacent rows inside the coding blocks of an image, respectively. That is, for an image $\mathbf{f}$,

$$\left(Q_{\mathrm{VS}}\mathbf{f}\right)(i,j) \triangleq \begin{cases} \mathbf{f}(i,j) - \mathbf{f}(i,j+1), & \text{if } ((j))_8 \neq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

Similarly, the operator Q_{HS} is defined for the rows of an image. Then, the quantity $\mathbf{J}_w(\mathbf{f}_M)$ as a whole characterizes the total variation inside all the coding blocks of the image frames in $\mathbf{f}_M$.

Observe that each term involved in (11) and (8) is a quadratic and convex function of the image vectors $\mathbf{f}_l$. As a result, the regularization terms $\mathbf{J}_w(\mathbf{f}_M)$ and $\mathbf{J}_b(\mathbf{f}_M)$ are quadratic and convex

functions of $\mathbf{f}_M$. This fact will be utilized in the derivation and implementation of the numerical algorithms later on.

B. Temporal Domain Regularization

Similar to the spatial domain regularization terms, we define the temporal domain regularization term $\mathbf{J}_t(\mathbf{f}_M)$ based on the differences between *temporally* neighboring pixels. The definition of $\mathbf{J}_t(\mathbf{f}_M)$, however, requires knowledge of the relative motion between the pixels in neighboring frames.

We consider two different motion compensation models in the following. The first model assumes that the motion vectors used are of integer precision. The second model, on the other hand, does not make use of this assumption and deals with the more general case of noninteger motion vectors. For easy reference, the first model will be referred to as *integer motion model* and the second as *noninteger motion model* in the rest of the paper. The use of the integer motion model is that it yields reduction in the complexity of the reconstruction algorithm.

First, let's introduce some standard notation for motion compensation. Consider two frames $\mathbf{f}_k$ and $\mathbf{f}_l$ in an image sequence. Let $\left(x_{k,l}^{(i,j)}, y_{k,l}^{(i,j)}\right)$ denote the motion vector of the pixel (i,j) in frame $\mathbf{f}_l$ with respect to its correspondence in the reference frame $\mathbf{f}_k$. Then the motion compensated estimate of frame $\mathbf{f}_l$ from the reference frame $\mathbf{f}_k$, denoted by $\hat{\mathbf{f}}_l^k$, is given by

$$\hat{\mathbf{f}}_l^k(i,j) = \mathbf{f}_k\left(i + x_{k,l}^{(i,j)}, j + y_{k,l}^{(i,j)}\right) \tag{13}$$

at pixel (i,j).

In general, the motion vector $\left(x_{k,l}^{(i,j)}, y_{k,l}^{(i,j)}\right)$ in (13) is of subpixel precision, and the term $\mathbf{f}_k\left(i + x_{k,l}^{(i,j)}, j + y_{k,l}^{(i,j)}\right)$ is computed from its neighboring pixels using linear interpolation [20]. As a result, (13) can be written in a compact matrix form as

$$\hat{\mathbf{f}}_l^k = \mathbf{M}_{k,l}\mathbf{f}_k \tag{14}$$

where the matrix $\mathbf{M}_{k,l}$ is used to denote the compensation operation of frame $\mathbf{f}_l$ from reference frame $\mathbf{f}_k$.

The *motion-compensated prediction error* of frame $\mathbf{f}_l$ from reference frame $\mathbf{f}_k$ is then given by

$$\|\mathbf{f}_l - \hat{\mathbf{f}}_l^k\|^2 = \|\mathbf{f}_l - \mathbf{M}_{k,l}\mathbf{f}_k\|^2. \tag{15}$$

Or alternatively, from (13), we have

$$\begin{aligned}\|\mathbf{f}_l - \hat{\mathbf{f}}_l^k\|^2 &= \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{f}_l(i,j) - \hat{\mathbf{f}}_l^k(i,j)\right)^2 \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{f}_l(i,j) - \mathbf{f}_k(i + x_{k,l}^{(i,j)}, j + y_{k,l}^{(i,j)})\right)^2.\end{aligned} \tag{16}$$

1) Integer Motion Model: Using the multichannel notion $\mathbf{f}_M$ introduced in (5), we define its motion-compensated estimate, denoted by $\hat{\mathbf{f}}_M$, as

$$\hat{\mathbf{f}}_M^{\mathrm{T}} \triangleq \left(\hat{\mathbf{f}}_1^{\mathrm{T}}, \hat{\mathbf{f}}_2^{\mathrm{T}}, \ldots, \hat{\mathbf{f}}_L^{\mathrm{T}}\right)^{\mathrm{T}} \tag{17}$$

where $\hat{\mathbf{f}}_l$, $l = 1, 2, \ldots, L$, is defined by

$$\hat{\mathbf{f}}_l(i,j) \triangleq \frac{1}{L-1}\sum_{k=1, k\neq l}^{L} \hat{\mathbf{f}}_l^k(i,j) \tag{18}$$

with $\hat{\mathbf{f}}_l^k$ defined in (13). In other words, $\hat{\mathbf{f}}_l$ is the simple average[1] of the motion-compensated estimates of frame $\mathbf{f}_l$ from other $(L-1)$ frames. Using the matrix notation in (14), we have

$$\hat{\mathbf{f}}_l \triangleq \frac{1}{L-1}\sum_{k=1, k\neq l}^{L} \mathbf{M}_{k,l}\mathbf{f}_k. \tag{19}$$

The regularization term $\mathbf{J}_t(\mathbf{f}_M)$ is then defined as

$$\mathbf{J}_t(\mathbf{f}_M) \triangleq \|\mathbf{f}_M - \hat{\mathbf{f}}_M\|^2 = \sum_{l=1}^{L}\|\mathbf{f}_l - \hat{\mathbf{f}}_l\|^2. \tag{20}$$

From (19), we can describe the term $\mathbf{J}_t(\mathbf{f}_M)$ using a multichannel linear operator $\mathbf{Q}_{\mathrm{MC}}$ such that

$$\mathbf{J}_t(\mathbf{f}_M) = \|\mathbf{Q}_{\mathrm{MC}}\mathbf{f}_M\|^2. \tag{21}$$

For the purpose of illustration, consider the case of $L = 3$. The operator $\mathbf{Q}_{\mathrm{MC}}$ is given by

$$\mathbf{Q}_{\mathrm{MC}} = \begin{bmatrix} \mathbf{I} & -\frac{1}{2}\mathbf{M}_{2,1} & -\frac{1}{2}\mathbf{M}_{3,1} \\ -\frac{1}{2}\mathbf{M}_{1,2} & \mathbf{I} & -\frac{1}{2}\mathbf{M}_{3,2} \\ -\frac{1}{2}\mathbf{M}_{1,3} & -\frac{1}{2}\mathbf{M}_{2,3} & \mathbf{I} \end{bmatrix} \tag{22}$$

where $\mathbf{I}$ is the identity matrix. Such a structure of $\mathbf{Q}_{\mathrm{MC}}$ will be utilized later on for the derivation of numerical algorithms.

2) Noninteger Motion Model: In order to handle the more general case of motion compensation such as that used in MPEG and H.263, we can alternatively define the temporal regularization term $\mathbf{J}_t(\mathbf{f}_M)$ directly based on frame-to-frame motion compensated prediction errors. Specifically, we define

$$\mathbf{J}_t(\mathbf{f}_M) \triangleq \sum_{l=1}^{L}\sum_{k=1, k\neq l}^{L}\left\|\mathbf{f}_l - \hat{\mathbf{f}}_l^k\right\|^2 \tag{23}$$

where $\hat{\mathbf{f}}_l^k$ is given in (13). In its definition, the term $\mathbf{J}_t(\mathbf{f}_M)$ captures the total variation of the pixels in each frame $\mathbf{f}_l$ from their correspondences in the other $(L-1)$ frames $\mathbf{f}_k$ along their motion trajectories.

Despite the assumptions made in the integer motion model, it is still of interest to examine how the definition of $\mathbf{J}_t(\mathbf{f}_M)$ in (23) to that in (20). For simplicity, consider again the case of $L = 3$. First, the definition in (23) leads to

$$\begin{aligned}\mathbf{J}_t(\mathbf{f}_M) = &\left\|\mathbf{f}_1 - \hat{\mathbf{f}}_1^2\right\|^2 + \left\|\mathbf{f}_1 - \hat{\mathbf{f}}_1^3\right\|^2 + \left\|\mathbf{f}_2 - \hat{\mathbf{f}}_2^1\right\|^2 \\ &+ \left\|\mathbf{f}_2 - \hat{\mathbf{f}}_2^3\right\|^2 + \left\|\mathbf{f}_3 - \hat{\mathbf{f}}_3^1\right\|^2 + \left\|\mathbf{f}_3 - \hat{\mathbf{f}}_3^2\right\|^2.\end{aligned} \tag{24}$$

[1]A weighted average may be a reasonable alternative to the simple average, where weighting is used to reflect the degree of correlation between two different frames (e.g., based on their temporal distance). Nevertheless, the simple average is used here in favor of its simplicity.

On the other hand, the definition in (20) gives

$$\begin{aligned}\mathbf{J}_t(\mathbf{f}_M) =& \left\|\mathbf{f}_1 - \frac{1}{2}\left(\hat{\mathbf{f}}_1^2 + \hat{\mathbf{f}}_1^3\right)\right\|^2 + \left\|\mathbf{f}_2 - \frac{1}{2}\left(\hat{\mathbf{f}}_2^1 + \hat{\mathbf{f}}_2^3\right)\right\|^2 \\ &+ \left\|\mathbf{f}_3 - \frac{1}{2}\left(\hat{\mathbf{f}}_3^1 + \hat{\mathbf{f}}_3^2\right)\right\|^2 \qquad (25)\\ =& \frac{1}{4}\left\|\mathbf{f}_1 - \hat{\mathbf{f}}_1^2\right\|^2 + \frac{1}{4}\left\|\mathbf{f}_1 - \hat{\mathbf{f}}_1^3\right\|^2 \\ &+ \frac{1}{2}\left(\mathbf{f}_1 - \hat{\mathbf{f}}_1^2\right)^{\mathrm{T}}\left(\mathbf{f}_1 - \hat{\mathbf{f}}_1^3\right) \\ &+ \frac{1}{4}\left\|\mathbf{f}_2 - \hat{\mathbf{f}}_2^1\right\|^2 + \frac{1}{4}\left\|\mathbf{f}_2 - \hat{\mathbf{f}}_2^3\right\|^2 \\ &+ \frac{1}{2}\left(\mathbf{f}_2 - \hat{\mathbf{f}}_2^1\right)^{\mathrm{T}}\left(\mathbf{f}_2 - \hat{\mathbf{f}}_2^3\right) \\ &+ \frac{1}{4}\left\|\mathbf{f}_3 - \hat{\mathbf{f}}_3^1\right\|^2 + \frac{1}{4}\left\|\mathbf{f}_3 - \hat{\mathbf{f}}_3^2\right\|^2 \\ &+ \frac{1}{2}\left(\mathbf{f}_3 - \hat{\mathbf{f}}_3^1\right)^{\mathrm{T}}\left(\mathbf{f}_3 - \hat{\mathbf{f}}_3^2\right). \qquad (26)\end{aligned}$$

Comparing (26) with (24), one quickly realizes that the former is simply the latter (up to a scale factor) with additional cross terms of the form $(1/2)\left(\mathbf{f}_l - \hat{\mathbf{f}}_l^k\right)^{\mathrm{T}}\left(\mathbf{f}_l - \hat{\mathbf{f}}_l^{k'}\right)$ where $k \neq k'$. Note that each cross term $\left(\mathbf{f}_l - \hat{\mathbf{f}}_l^k\right)^{\mathrm{T}}\left(\mathbf{f}_l - \hat{\mathbf{f}}_l^{k'}\right)$ is essentially the cross correlation between the compensation errors $\mathbf{f}_l - \hat{\mathbf{f}}_l^k$ and $\mathbf{f}_l - \hat{\mathbf{f}}_l^{k'}$. In practice, it is reasonable to expect that these compensation errors are weakly correlated. In this sense, the two seemingly different definitions for the temporal regularization term $\mathbf{J}_t(\mathbf{f}_M)$ will play essentially the same role in the recovery algorithms.

A note is that when fast scene changes occur in a sequence some of the pixels in one frame may not be predictable from its neighboring frames. In such a case, the temporal regularization term $\mathbf{J}_t(\mathbf{f}_M)$ in (23) should be adjusted accordingly by ignoring the affected pixels in computing the motion-compensated prediction errors. This also applies to the case of the integer motion model discussed earlier.

Finally, due to the quadratic and convex nature of the motion-compensated prediction error terms in the regularization term $\mathbf{J}_t(\mathbf{f}_M)$, both forms of $\mathbf{J}_t(\mathbf{f}_M)$ are also quadratic and convex in terms of $\mathbf{f}_M$.

IV. Multichannel Regularized Recovery Algorithms

Considering the definition of the spatial and temporal regularization terms defined in the previous section, we can rewrite the regularized objective function $\mathbf{J}(\mathbf{f}_M)$

$$\begin{aligned}\mathbf{J}(\mathbf{f}_M) =& \sum_{l=1}^{L} \|\mathbf{f}_l - \mathbf{g}_l\|^2 + \lambda_1 \mathbf{J}_w(\mathbf{f}_M) + \lambda_2 \mathbf{J}_b(\mathbf{f}_M) \\ &+ \lambda_3 \mathbf{J}_t(\mathbf{f}_M) \qquad (27)\end{aligned}$$

where $\lambda_1 \triangleq \gamma\lambda_s$, $\lambda_2 \triangleq (1-\gamma)\lambda_s$, and $\lambda_3 \triangleq \lambda_t$.

Now that each regularization term in $\mathbf{J}(\mathbf{f}_M)$ is defined, the recovery of video $\mathbf{f}_M$ from its compressed data can then be solved by directly minimizing this objective function. From our earlier discussion in Section II it is known that each original frame $\mathbf{f}_l$ belongs to the constraint set C_l defined by the received data in (4). Therefore, we seek solution to the following constrained problem

$$\min \mathbf{J}(\mathbf{f}_M) \text{ subject to } \mathbf{f}_l \in C_l, \quad l = 1, 2 \ldots, L. \qquad (28)$$

The individual constraint sets C_l on the frames $\mathbf{f}_l$, $l = 1, 2, \ldots, L$, can be written equivalently as

$$C_M \triangleq \{(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L) : \mathbf{f}_l \in C_l, \quad l = 1, 2 \ldots, L\}. \qquad (29)$$

Then, (28) can be equivalently written as

$$\min \mathbf{J}(\mathbf{f}_M) \text{ subject to } \mathbf{f}_M \in C_M. \qquad (30)$$

As pointed out in the previous section, all the terms $\mathbf{J}_w(\mathbf{f}_M)$, $\mathbf{J}_b(\mathbf{f}_M)$, and $\mathbf{J}_t(\mathbf{f}_M)$ are quadratic and convex in terms of $\mathbf{f}_M$. As a result, the objective function $\mathbf{J}(\mathbf{f}_M)$ is also quadratic and convex in terms of $\mathbf{f}_M$. Also, the sets C_l, $l = 1, 2 \ldots, L$, are closed and convex [32], [34], and consequently so is the set C_M. Therefore, the optimization problem in (30) is of the form of a convex functional under a convex constraint. It is well known that a problem in such a form has a unique well-defined solution, and can be found using a so-called iterative gradient-projection algorithm [22], i.e.,

$$\mathbf{f}_M^{k+1} = P_C\left(\mathbf{f}_M^k - \alpha \nabla \mathbf{J}\left(\mathbf{f}_M^k\right)\right), \quad k = 0, 1, 2, \ldots \qquad (31)$$

where P_C is the projection operator onto the set C_M, α a relaxation parameter that controls the rate of convergence of the iteration, $\mathbf{f}_M^k$ denotes the estimate of $\mathbf{f}_M$ after the kth iteration, and $\nabla \mathbf{J}\left(\mathbf{f}_M^k\right)$ is the gradient of $\mathbf{J}(\mathbf{f}_M)$ with respect to $\mathbf{f}_M$ evaluated at $\mathbf{f}_M^k$. The computation of the projector P_C and the gradient $\nabla \mathbf{J}\left(\mathbf{f}_M^k\right)$ is explained in detail in the following.

A. The Projection Operator P_C

From (29), it is easy to see that for $\mathbf{f}_M$ its projection $P_C\mathbf{f}_M$ onto the set C_M is simply given by the projections of the individual $\mathbf{f}_l$ onto their corresponding sets C_l. That is,

$$P_C\mathbf{f}_M = \left(P_{C_1}\mathbf{f}_1, P_{C_2}\mathbf{f}_2, \ldots, P_{C_L}\mathbf{f}_L\right). \qquad (32)$$

The individual projection of $\mathbf{f}_l$ onto the set C_l is well known [32], [34]. For T_B an orthonormal coding transform such as the block DCT, the projection $P_{C_l}\mathbf{f}_l$ is given by

$$P_{C_l}\mathbf{f}_l = \begin{cases} T_B^{-1}\bar{\mathbf{F}}_l, & \text{if } \mathbf{f}_l \text{ is intra-coded} \\ \bar{\mathbf{f}}_l + T_B^{-1}\bar{\mathbf{F}}_l, & \text{if } \mathbf{f}_l \text{ is predictively coded} \end{cases} \qquad (33)$$

where the nth coefficient of $\bar{\mathbf{F}}_l$ is given by

$$(\bar{\mathbf{F}}_l)_n = \begin{cases} \alpha_n^{\min}, & \text{if } \left(T_B\bar{\mathbf{f}}_l\right)_n < \alpha_n^{\min} \\ \alpha_n^{\max}, & \text{if } \left(T_B\bar{\mathbf{f}}_l\right)_n > \alpha_n^{\max} \\ \left(T_B\bar{\mathbf{f}}_l\right)_n, & \text{otherwise.} \end{cases} \qquad (34)$$

B. The Gradient $\nabla \mathbf{J}(\mathbf{F}_M)$

The gradient $\nabla \mathbf{J}(\mathbf{f}_M)$ can be expressed in terms of the partial derivatives of $\mathbf{J}(\mathbf{f}_M)$ with respect to the individual frames $\mathbf{f}_l$ as

$$\nabla \mathbf{J}(\mathbf{f}_M) = \left(\frac{\partial \mathbf{J}(\mathbf{f}_M)}{\partial \mathbf{f}_1}, \frac{\partial \mathbf{J}(\mathbf{f}_M)}{\partial \mathbf{f}_2}, \ldots, \frac{\partial \mathbf{J}(\mathbf{f}_M)}{\partial \mathbf{f}_L}\right). \qquad (35)$$

For a particular frame $\mathbf{f}_l$, $l = 1, 2, \ldots, L$,

$$\frac{\partial \mathbf{J}(\mathbf{f}_M)}{\partial \mathbf{f}_l} = 2(\mathbf{f}_l - \mathbf{g}_l) + \lambda_1 \frac{\partial}{\partial \mathbf{f}_l}\mathbf{J}_w(\mathbf{f}_M) + \lambda_2 \frac{\partial}{\partial \mathbf{f}_l}\mathbf{J}_b(\mathbf{f}_M) + \lambda_3 \frac{\partial}{\partial \mathbf{f}_l}\mathbf{J}_t(\mathbf{f}_M). \tag{36}$$

In the following, we consider separately the computation of the partial derivative terms $(\partial/\partial \mathbf{f}_l)\mathbf{J}_w(\mathbf{f}_M)$, $(\partial/\partial \mathbf{f}_l)\mathbf{J}_b(\mathbf{f}_M)$ and $(\partial/\partial \mathbf{f}_l)\mathbf{J}_t(\mathbf{f}_M)$ in (36).

1) The Gradients $\nabla \mathbf{J}_w(\mathbf{f}_M)$ and $\nabla \mathbf{J}_b(\mathbf{f}_M)$: From (8), we obtain

$$\frac{\partial \mathbf{J}_b(\mathbf{f}_M)}{\partial \mathbf{f}_l} = \frac{\partial}{\partial \mathbf{f}_l}\|Q_{\mathrm{HB}}\mathbf{f}_l\|^2 + \frac{\partial}{\partial \mathbf{f}_l}\|Q_{\mathrm{VB}}\mathbf{f}_l\|^2. \tag{37}$$

The partial derivative terms in (37) can be easily computed. Take the second term in (37) for example. From the definition of Q_{VB} in (9), we have

$$\frac{\partial}{\partial f_l(i,j)}\|Q_{\mathrm{VB}}\mathbf{f}_l\|^2 = \begin{cases} 2[\mathbf{f}_l(i,j) - \mathbf{f}_l(i,j+1)], & \text{if } ((j))_8 = 0 \\ 2[\mathbf{f}_l(i,j) - \mathbf{f}_l(i,j-1)], & \text{if } ((j))_8 = 1 \\ 0, & \text{otherwise .} \end{cases} \tag{38}$$

Similarly, $\nabla \mathbf{J}_w(\mathbf{f}_M)$the other terms in (40) can be derived. The details are omitted for brevity.

2) The Gradient $\nabla \mathbf{J}_t(\mathbf{f}_M)$: In Section III, two different forms for the temporal regularization term were proposed. In the following, we describe its gradient for both cases in detail. First, for the case of integer motion model

$$\frac{\partial \mathbf{J}_t(\mathbf{f}_M)}{\partial \mathbf{f}_M} = 2\mathbf{Q}_{\mathrm{MC}}^{\mathrm{T}}\mathbf{Q}_{\mathrm{MC}}\mathbf{f}_M. \tag{39}$$

The computation in (39) involves the matrix $\mathbf{Q}_{\mathrm{MC}}^{\mathrm{T}}$, which may require large memory. To avoid this difficulty, We propose a scheme to approximate this computation.

Recall the matrix structure for the operator $\mathbf{Q}_{\mathrm{MC}}$ in (22). Under the assumption that each motion compensation matrix $\mathbf{M}_{k,l}$ satisfies the following condition

$$(\mathbf{M}_{k,l})^{\mathrm{T}} = \mathbf{M}_{l,k} \tag{40}$$

the matrix $\mathbf{Q}_{\mathrm{MC}}$ becomes symmetric, i.e., $\mathbf{Q}_{\mathrm{MC}}^{\mathrm{T}} = \mathbf{Q}_{\mathrm{MC}}$. In such a case, the quantity in (39) can be easily computed since the operation of $\mathbf{Q}_{\mathrm{MC}}$ on $\mathbf{f}_M$ is simply the motion compensated prediction error for each frame.

It can be shown that the condition in (40) holds under the following assumptions: 1) each pixel from frame k is used only once to compensate a single pixel of frame l and 2) there is a one-to-one correspondence between the pixels of frames l and k. In practice, however, these two conditions are only *approximately* true because the motion vectors found through a motion estimation algorithm (such as block matching) may not preserve such a one-to-one correspondence between pixels in two frames. In our implementation, we make this approximation in favor of simplicity.

For the case of the noninteger motion model, the computation of the gradient $\mathbf{J}_t(\mathbf{f}_M)$ needs some detailed explanation. From the definition of $\mathbf{J}_t(\mathbf{f}_M)$ in (23)

$$\frac{\partial \mathbf{J}_t(\mathbf{f}_M)}{\partial \mathbf{f}_l} = 2\sum_{k=1, k\neq l}^{L}\left(\mathbf{f}_l - \hat{\mathbf{f}}_l^k\right) + \sum_{k=1, k\neq l}^{L}\frac{\partial\|\mathbf{f}_k - \hat{\mathbf{f}}_k^l\|^2}{\partial \mathbf{f}_l}. \tag{41}$$

The first term in (41) involves all the frames used for the prediction of $\mathbf{f}_l$, while the second term involves all the frames predicted from $\mathbf{f}_l$. The partial derivatives $(\partial\|\mathbf{f}_k - \hat{\mathbf{f}}_k^l\|^2/\partial \mathbf{f}_l)$ in the second term can be further computed as: for a particular pixel location (i,j),

$$\frac{\partial\left\|\mathbf{f}_k - \hat{\mathbf{f}}_k^l\right\|^2}{\partial \mathbf{f}_l(i,j)} = -2\sum_{(i',j')\in N_k^l(i,j)}\left(\mathbf{f}_k(i',j') - \hat{\mathbf{f}}_k^l(i',j')\right) \times \frac{\partial \hat{\mathbf{f}}_k^l(i',j')}{\partial \mathbf{f}_l(i,j)} \tag{42}$$

where $N_k^l(i,j)$ is the set of pixels in frame k that are motion compensated by $\mathbf{f}_l(i,j)$.

Clearly, the computation of the second term of the gradient in (41) can be cumbersome in that in computing $(\partial\|\mathbf{f}_k - \hat{\mathbf{f}}_k^l\|^2/\partial \mathbf{f}_l(i,j))$ some *bookkeeping* is necessary to keep track of the pixels in frame $\mathbf{f}_k$ that are predicted from pixel $\mathbf{f}_l(i,j)$. In practice, however, this task can be bypassed in favor of lower computational complexity by introducing an approximation to the temporal regularization term $\mathbf{J}_t(\mathbf{f}_M)$. Indeed, consider the definition of $\mathbf{J}_t(\mathbf{f}_M)$ for the case of noninteger motion model in (23)

$$\mathbf{J}_t(\mathbf{f}_M) = \sum_{l=1}^{L}\sum_{k=1, k\neq l}^{L}\left\|\mathbf{f}_l - \hat{\mathbf{f}}_l^k\right\|^2. \tag{43}$$

Recall that the term $\hat{\mathbf{f}}_l^k$ in $\mathbf{J}_t(\mathbf{f}_M)$ is used to denote the motion-compensated prediction of $\mathbf{f}_l$ from reference frame $\mathbf{f}_k$. If we approximate $\hat{\mathbf{f}}_l^k$, the motion-compensated prediction of $\mathbf{f}_l$ from $\mathbf{f}_k$, by using a most recently available (say, from the previous iteration) estimate of frame $\mathbf{f}_k$ instead of the true frame $\mathbf{f}_k$ itself, then the gradient $(\partial \tilde{\mathbf{J}}_t(\mathbf{f}_M)/\partial \mathbf{f}_l)$ in (41) can be approximated by

$$\frac{\partial \mathbf{J}_t(\mathbf{f}_M)}{\partial \mathbf{f}_l} \approx 2\sum_{k=1, k\neq l}^{L}\left(\mathbf{f}_l - \hat{\mathbf{f}}_l^k\right). \tag{44}$$

In the rest of this paper, this approximation is simply referred to as *approximate model*. The performance of this simplified approach will be discussed further in the experiments presented in the next section.

C. Analysis of Numerical Complexity of the Recovery Algorithms

The proposed recovery algorithms are iterative in nature. Nevertheless, it is still informative to analyze the major numerical computations involved in the implementation. For this

purpose, we will provide an approximate count of major arithmetic operations required for each iteration of the algorithms.

Combining the iteration in (31) and the gradient in (36), we see that during each iteration the following operation is needed for each frame $\mathbf{f}_l$:

$$\mathbf{f}_l^{k+1} = P_{C_l}\Big(\mathbf{f}_l^k - 2\alpha\left(\mathbf{f}_l^k - \mathbf{g}_l\right) - \alpha\lambda_1\frac{\partial}{\partial \mathbf{f}_l}\mathbf{J}_w(\mathbf{f}_M) - \alpha\lambda_2\frac{\partial}{\partial \mathbf{f}_l}\mathbf{J}_b(\mathbf{f}_M) - \alpha\lambda_3\frac{\partial}{\partial \mathbf{f}_l}\mathbf{J}_t(\mathbf{f}_M)\Big) \tag{45}$$

where each of the partial derivative term is evaluated at $\mathbf{f}_M = \mathbf{f}_M^k$.

A breakdown of the operations involved in (45) is as follows:

- The evaluation of the two terms $(\partial/\partial \mathbf{f}_l)\mathbf{J}_w(\mathbf{f}_M)$ and $(\partial/\partial \mathbf{f}_l)\mathbf{J}_b(\mathbf{f}_M)$ combined will require *only two additions (or subtractions to be exact) per pixel*. Note that the factor 2 in (38) can be combined with the multiplying constants in (45).
- From (39) and the discussions immediately after, it can be seen that for the term $(\partial/\partial \mathbf{f}_l)\mathbf{J}_t(\mathbf{f}_M)$ an equivalent of twice as many as operations associated with the operator $\mathbf{Q}_{\mathrm{MC}}$ are needed. Based on the structure of $\mathbf{Q}_{\mathrm{MC}}$ illustrated in (22), we see that the operation of $\mathbf{Q}_{\mathrm{MC}}$ can be accomplished using $(L-1)$ additions and 1 division. Thus, $(\partial/\partial \mathbf{f}_l)\mathbf{J}_t(\mathbf{f}_M)$ can be evaluated using *a total of* $2(L-1)$ *additions and 2 divisions*. Note that the operations associated with computing the motion vectors are not included here. Similarly, a computation count can also be obtained when the noninteger motion model or its simplified variant is used.
- The projection operation associated with P_{C_l}, as shown in (34), is dominated by the transformation of the image data to and from the transform domain. This is usually accomplished by a fast transform such as the fast DCT. The computational cost of the DCT depends on the specific implementation. In the algorithm reported in [15], for each $K \times K$ DCT transform, $K^2 \log_2 K$ real multiplications and $K(3K \log_2 K - K + 1)$ real additions are required. For $K = 8$, this corresponds to three multiplications and approximately eight additions per pixel. Note that both the forward and backward DCTs are required for P_{C_l}.

In summary, the total number of operations involved in each iteration of (51) is approximately as follows: 11 multiplications and $23 + 2(L-1)$ additions per pixel. For the case of $L = 5$, which is used in our experiments later on, this corresponds to 11 multiplications (6 of which is for DCT) and 31 additions (16 of which is for DCT) per pixel. Interestingly, this is roughly equivalent to four times the computational cost of decoding an I frame.

V. Experiments

In this section, we use a number of numerical experiments to test the proposed video recovery algorithms. We focus on two major issues in these experiments: 1) the impact of the use of various temporal regularization terms on the performance of the proposed recovery algorithms, and 2) the choice of their associated regularization parameters. In particular, we will compare the performance of the proposed multichannel recovery approach against that of spatial only recovery approach. In addition, we will propose a new approach for the determination of regularization parameters used in the objective function.

A. Experiments on Regularization Terms

The proposed recovery algorithms are tested using H.261/3 compressed video streams. Note, however, that these algorithms can be applied to MPEG compressed video streams as well. The following sequences are used: "Mother and Daughter (MD)," "Foreman," and "Carphone" sequences. The picture format is QCIF(Y: 176×144, U and V: 88×72) for all test sequences. A total of 30 frames from each selected sequence are used in our experiments. Three experiments are presented here. The frame rates used in these three experiments, from experiment one to experiment three, are 10, 30, and 10 frames/s, respectively.

As an objective metric the peak signal-to-noise ratio (PSNR) is used. This metric is defined by

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2 \cdot W}{\|\mathbf{f}_s - \mathbf{f}\|^2}\ \mathrm{dB} \tag{46}$$

where W is the total number of pixels in the image, and $\mathbf{f}_s$ and $\mathbf{f}$ are the original image and the recovered image, respectively.

Experiment 1: For this experiment,the proposed recovery algorithm in (31) was tested using H.261 compressed video streams. To demonstrate the merit of temporal regularization, the following different forms of regularization terms were considered:

1) Spatial-only regularization was used. That is, the temporal regularization term $\mathbf{J}_t(\mathbf{f}_M)$ was simply ignored in the objective function $\mathbf{J}(\mathbf{f}_M)$ in (6).
2) Temporal regularization based on the integer motion model as in (20) was used. In this experiment, motion vectors were estimated on a pixel-by-pixel basis by using a full search block matching algorithm from low-pass smoothed compressed frames. Also, $L = 5$ (i.e., 5-channel) was used in (19).
3) Temporal regularization based on the exact noninteger motion model as in (23) was used. Similar to case 2, pixel-by-pixel motion vectors were used for motion compensation, and $L = 5$ was used.
4) Similar to case 3, except that temporal regularization based on the approximate model as in (44) was used.

All four regularization approaches described above were applied to 30 frames of the "Foreman" sequence compressed at 31.69 kb/s and "Carphone" sequence compressed at 28.25 kb/s. The results for these two sequences are summarized in Figs. 1 and 2, respectively, where the PSNR values of the recovered frames for each case listed above are shown. Due to space limitation, we show in Fig. 3 the recovered images only for a particular frame (#236) of the "Foreman" sequence for case 1 and

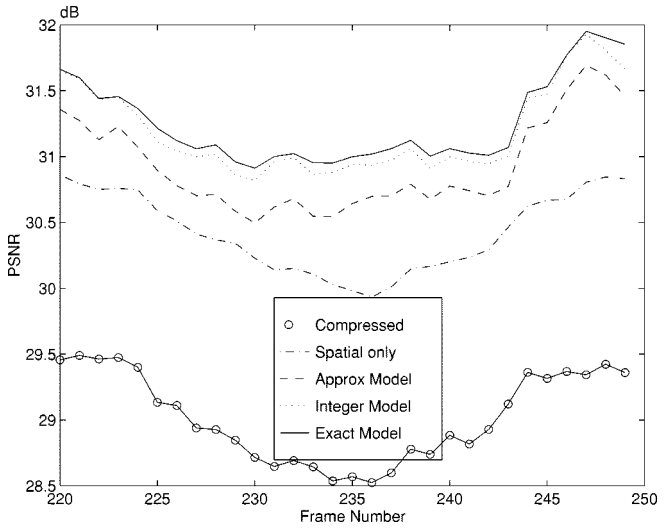


Fig. 1. PSNRs of recovered frames from the H.261 compressed "Foreman" sequence using four different regularization approaches: spatial only (dashed–dotted line), integer motion model (dotted line), noninteger motion model (solid line), and approximate model (dashed line).

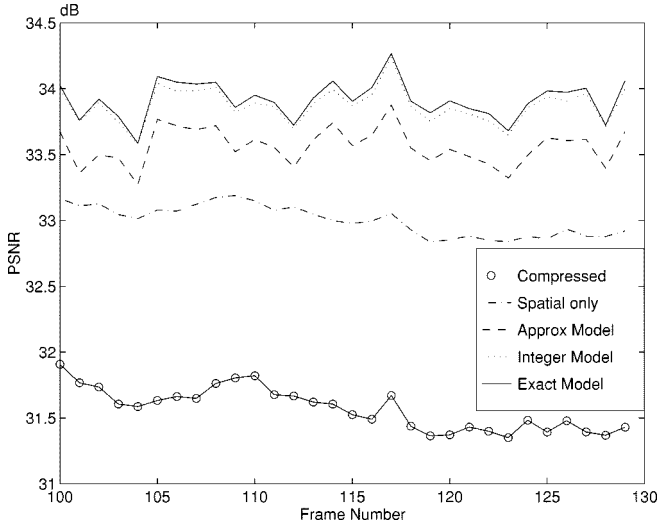


Fig. 2. PSNRs of recovered frames from the H.261 compressed "Carphone" sequence using four different regularization approaches: spatial only (dashed–dotted line), integer motion model (dotted line), noninteger motion model (solid line), and approximate model (dashed line).

case 4. Also shown for comparison in this figure are the compressed image and the original image.

Experiment 2: When there is little or no temporal correlation in a sequence, one would naturally expect that the multichannel approach should perform just as well as the spatial only regularization approach. Indeed, in such a case one can simply set $\lambda_3 = 0$ in (6) to discard temporal regularization.

In this experiment, the test stream used was the "Foreman" sequence coded at 38.6 kb/s using H.261 with five-frame skip. The same four regularization cases described in Experiment 1 were tested. The results are summarized in Fig. 4 where the PSNR values of the recovered frames are shown. From the results in Fig. 4 it seems that the spatial only regularization approach outperforms the multichannel approaches during the period from frame 240 to frame 280. A closer examination of the image sequence, however, reveals that frequent scene changes occur during this segment of the sequence. As a result, temporal regularization becomes less effective. As pointed out earlier, the regularization parameter λ_3 should be set zero in (6) in such a case. Nevertheless, in order to demonstrate its effect, we still kept λ_3 nonzero during this segment in this experiment. In such a case, the temporal regularization term $\mathbf{J}_t(\mathbf{f}_M)$ in (6) tends to assume a very large value because of less predictability between frames. As a result, this term dominates over the spatial regularization terms in $\mathbf{J}_t(\mathbf{f}_M)$ and makes them less effective.

In a practical implementation, one may utilize a scene-change detection mechanism to automatically adjust the value of λ_3 so as to control the effect of temporal regularization. Note that when a scene-change occurs the interframe prediction error is expected to rise drastically. This will likely cause a sudden increase in the number of intracoded macroblocks in a predictive frame (i.e., a P or B frame in MPEG coding). Therefore, one may detect the occurrence of a scene-change by simply monitoring any sudden increase in either the interframe prediction error or the number of intra-coded macroblocks in a frame. Of course, other more sophisticated schemes for detecting scene changes exist (e.g., see [2], [16]) and can also be employed.

Experiment 3: In this experiment, the proposed algorithms were tested using H.263 compressed streams. In these streams, the following modes of H.263 compression were allowed: unrestricted motion vector mode (annex-D), advanced prediction mode (annex-F), PB frame mode (annex-G), and syntax based arithmetic coding (annex-E) [13]. The recovery algorithms were tested again using various forms of temporal regularization—all those used in Experiment 1 except case 2 where the integer motion model was assumed. The motion vectors used were of half-pel precision. In addition to using reestimated motion vectors from the compressed images the algorithm was also tested using the available transmitted motion vectors.

The recovery results obtained using the "MD" sequence compressed at 10.67 kb/s are summarized in Fig. 5 where the PNSR values of the recovered frames are shown. In Fig. 6, recovered images are shown for a particular frame (#22) from the "MD" sequence for case 1 (spatial only regularization) and case 4 (noninteger motion model assumed using the reestimated motion field). Also shown in Fig. 6 for comparison are the original image and the compressed image of that frame.

Finally, it was observed in our experiments that the proposed gradient-projection recovery algorithms converged rather rapidly. All the results in the three experiments discussed above were obtained with less than 10 iterations of (31).

B. Experiments on Regularization Parameters

As in every regularized image recovery problem, the proper choice of these regularization parameters by itself is an interesting yet challenging problem [7]. However, we realize that there is an interesting differentiating aspect in the compressed video recovery problem from other traditional image recovery problems—the original images of the compressed video are known at the coder. An interesting question immediately arises: can we exploit this valuable information to determine the values of the regularization parameters at the coder? If so, then their values can be transmitted along with the compressed data as a minimal coding overhead (say, in the user data fields of the

(a) (b) (c) (d)

Fig. 3. Images of the 236th frame of the "Foreman" sequence: (a) the original image; (b) the compressed image; (c) the recovered image with spatial only regularization; and (d) the recovered image using regularization based on noninteger motion model.

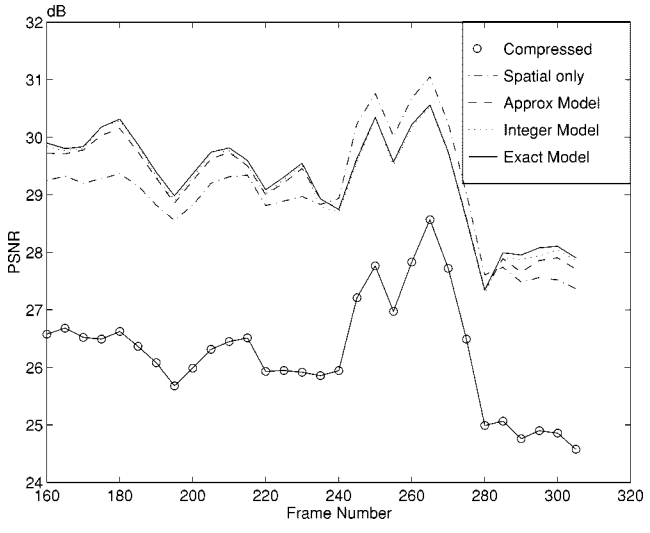


Fig. 4. PSNRs of recovered frames from the H.261 compressed "Foreman" sequence (with five-frame skip) using four different regularization approaches: spatial only (dashed–dotted line), integer motion model (dotted line), noninteger motion model (solid line), and approximate model (dashed line).

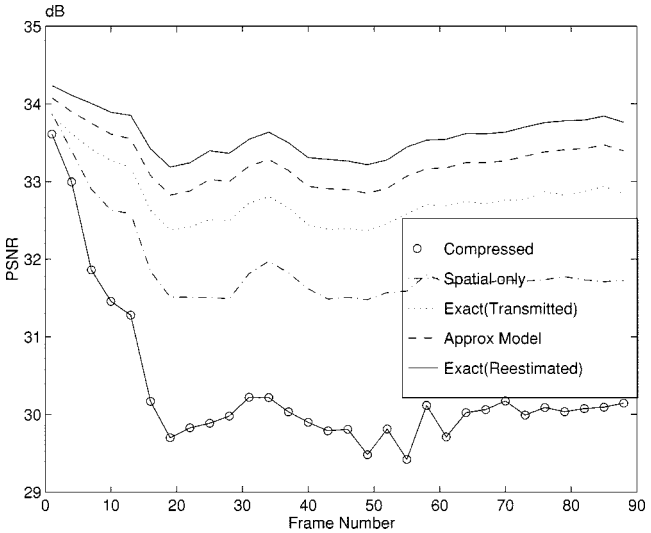
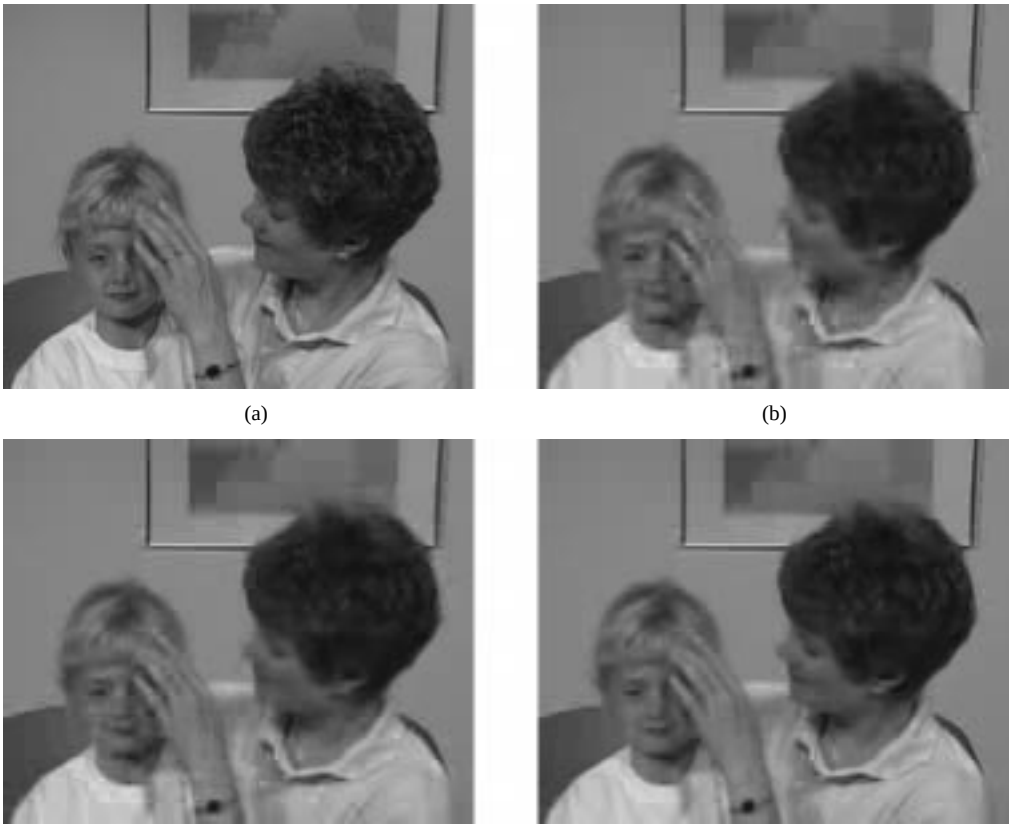


Fig. 5. PSNRs of recovered frames from the H.263 compressed "MD" sequence using three different regularization approaches: spatial only (dashed–dotted line), noninteger motion model using transmitted motion field (labeled as Exact(Transmitted), dotted line), noninteger motion model using reestimated motion field (labeled as Exact(Reestimated), solid line), and approximate model using the reestimated motion field (dashed line).

compressed data stream) so that they can be used in the decoder by the proposed compressed video recovery algorithms. This, of course, would serve as a valuable alternative to the traditional approaches for the determination of regularization parameters such as a trial-and-error approach. Motivated by this, we attempt to seek an answer to this question in the following.

Recall that our proposed recovery approach to compressed video was based on the solution to the constrained optimization of the objective function $\mathbf{J}(\mathbf{f_M})$ in (27). Clearly, the solution to this problem will depend on the regularization parameters λ_1, λ_2, and λ_3 in a nonlinear fashion. Let this solution be denoted by



(a) (b) (c) (d)

Fig. 6. Images of the 22nd frame of the "MD" sequence: (a) the original image; (b) the compressed image; (c) the recovered image with spatial only regularization; and (d) the recovered image using temporal regularization based on noninteger motion model using reestimated motion field.

$\mathbf{f}_M^*(\lambda_1, \lambda_2, \lambda_3)$. Also, let $\mathbf{f}_M^s$ denote the original image sequence in the coder, then a reasonable choice for λ_1, λ_2, and λ_3 seems to minimize the difference between $\mathbf{f}_M^*(\lambda_1, \lambda_2, \lambda_3)$, and $\mathbf{f}_M^s$. That is, we seek

$$(\lambda_1{}^*, \lambda_2{}^*, \lambda_3{}^*) = \arg \min_{\lambda_1, \lambda_2, \lambda_3} \|\mathbf{f}_M^s - \mathbf{f}_M^*(\lambda_1, \lambda_2, \lambda_3)\|^2. \quad (47)$$

Unfortunately, this is a difficult task due to the nonlinearity of $\mathbf{f}_M^*(\lambda_1, \lambda_2, \lambda_3)$. Instead, we propose the following iterative algorithm to approximate this solution:

1) Let iteration index $k = 0$, and let $(\lambda_1{}^0, \lambda_2{}^0, \lambda_3{}^0)$ and $\mathbf{f}_M^0$ denote the initial guess for $(\lambda_1{}^*, \lambda_2{}^*, \lambda_3{}^*)$ and $\mathbf{f}_M^*(\lambda_1^*, \lambda_2^*, \lambda_3^*)$, respectively.
2) During iteration k, define function $\mathbf{f}_M^{k+1}(\lambda_1, \lambda_2, \lambda_3) \triangleq \mathbf{f}_M^k - \alpha \nabla \mathbf{J}(\mathbf{f}_M^k)$. Note that $\nabla \mathbf{J}(\mathbf{f}_M^k)$ is a linear function of $(\lambda_1, \lambda_2, \lambda_3)$, hence so is $\mathbf{f}_M^{k+1}(\lambda_1, \lambda_2, \lambda_3)$. Next, solve for

$$(\lambda_1{}^{k+1}, \lambda_2{}^{k+1}, \lambda_3{}^{k+1}) = \arg \min_{\lambda_1, \lambda_2, \lambda_3} \left\|\mathbf{f}_M^s - \mathbf{f}_M^{k+1}(\lambda_1, \lambda_2, \lambda_3)\right\|^2. \quad (48)$$

   A closed-form solution for $(\lambda_1{}^{k+1}, \lambda_2{}^{k+1}, \lambda_3{}^{k+1})$ can be readily derived since $\mathbf{f}_M^{k+1}(\lambda_1, \lambda_2, \lambda_3)$ is linear in $(\lambda_1, \lambda_2, \lambda_3)$. We omit such details for brevity.
3) Compute

$$\mathbf{f}_M^{k+1} \triangleq P_C\left(\mathbf{f}_M^{k+1}\left(\lambda_1{}^{k+1}, \lambda_2{}^{k+1}, \lambda_3{}^{k+1}\right)\right). \quad (49)$$

4) If $\max\{|\lambda_i^{k+1} - \lambda_i^k|,\ i = 1, 2, 3\} < \epsilon$, a prescribed threshold, then stop; otherwise, let $k = k + 1$, go to step 2.

A few remarks are immediately in order.

1) The operations in (48) and (49) can be viewed as a two-step implementation of the gradient-projection operation in the recovery algorithm in (31), except that now an optimal value for $(\lambda_1, \lambda_2, \lambda_3)$ is searched in the step of (48).
2) From a computational point of view, the number of operations needed for computing (48) is only a fraction of that needed for (49). Thus, the computational complexity for calculating $(\lambda_1, \lambda_2, \lambda_3)$ in each iteration is of the same order as that of the recovery algorithm in each iteration.
3) It is observed from our extensive numerical experiments [4] that the iterates $\left(\lambda_1{}^k, \lambda_2{}^k, \lambda_3{}^k\right)$ generated in (48) always converge to a final value. More interestingly, it is observed that this final value is independent of both the choice of initial starting points $\left(\lambda_1{}^0, \lambda_2{}^0, \lambda_3{}^0\right)$ and $\mathbf{f}_M^0$, and choice of relaxation parameter α.
4) It is confirmed from our extensive numerical experiments that regularization parameters determined from this proposed algorithm yield quite satisfactory recovery results. In fact, it has been used in the three experiments presented above.

In the following, we present some numerical results to demonstrate: 1) the properties of the regularization parameters

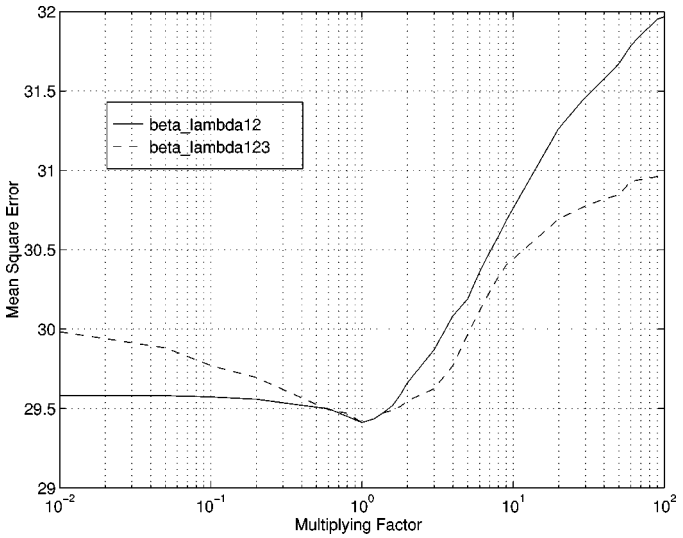


Fig. 7. MSEs of the recovered 116th frame of the H.261 compressed "Carphone" sequence when the regularization parameters are varied using the following schemes: (Plots of MSEs of the recovered 116th frame of the H.261 compressed "Carphone" sequence when the regularization parameters are varied using the following schemes: $(\lambda_1, \lambda_2, \lambda_3) = (\beta{\lambda_1}^*, \beta{\lambda_2}^*, \beta{\lambda_3}^*)$ (dashed line) and $(\lambda_1, \lambda_2, \lambda_3) = (\beta{\lambda_1}^*, \beta{\lambda_2}^*, {\lambda_3}^*)$ (solid line).

obtained from this iterative algorithm and 2) the sensitivity to the choice of regularization parameters of the performance of the proposed recovery algorithms.

Experiment 4: In this experiment, the iterative algorithm described above in (48) and (49) was first tested using the "Carphone" stream used in Experiment 1. The regularization parameters ${\lambda_1}^*$, ${\lambda_2}^*$, ${\lambda_3}^*$ were first obtained using this iterative algorithm for a particular frame (#116). The value of the relaxation parameter α was chosen to be 0.01. Then, to test the "optimality" of this solution, their values were varied using the following two arbitrarily chosen schemes: 1) $(\lambda_1, \lambda_2, \lambda_3) = (\beta{\lambda_1}^*, \beta{\lambda_2}^*, \beta{\lambda_3}^*)$ and 2) $(\lambda_1, \lambda_2, \lambda_3) = (\beta{\lambda_1}^*, \beta{\lambda_2}^*, {\lambda_3}^*)$, where β is a scaling parameter ranging from 10^{-2} to 10^{2}. The proposed recovery algorithms were then applied using these varied values. Shown in Fig. 7 are the mean square errors (MSEs) of the recovered image compared to the original frame versus the scaling parameter. The MSE is used here instead of the PSNR in order to show its low sensitivity to β. Note that in both cases, the minimal MSE was achieved when $\beta = 1$, i.e., when there was no deviation from the values determined by the iterative algorithm. The regularization approach used here was case 3 in Experiment 1, i.e., the noninteger motion model was used.

Similar results were obtained when the above procedures were applied to several other streams.

VI. Conclusion

In this paper, we proposed a multichannel regularization approach to address the video decoding problem. Temporal domain regularization is used, in addition to spatial domain regularization, to complement the transmitted data. The role of temporal regularization is to enforce smoothness along the motion trajectories defined by the transmitted motion field. Several forms of temporal regularization terms with different computational complexity are examined. In our proposed approach, the recovered images are obtained by using the well-known gradient-projection algorithm from the compressed video data. In addition, a novel iterative algorithm is proposed to address the classical problem of how to determine the regularization parameters. This algorithm takes advantage of the unique feature of video compression in that the original images are available at the coder.

A number of numerical experiments using H.261 and H.263 compressed streams were presented to demonstrate the performance of the proposed algorithms. Results from these experiments show that the proposed recovery approach can effectively exploit both the temporal and spatial correlations in an image sequence. It is observed that significant improvement can be obtained in the quality of the recovered images—in terms of both visual evaluation and objective PSNR measure. In particular, various forms of compression artifacts in the compressed images are greatly reduced in the recovered images.

References

[1] J. Brailean, R. Kleihorst, S. Efstratiadis, A. Katsaggelos, and R. Lagendijk, "Noise reduction filters for dynamic image sequences: a review," *Proc. IEEE*, vol. 83, pp. 1272–1292, Sept. 1995.

[2] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, pp. 844–851, Aug. 2000.

[3] M. G. Choi, N. P. Galatsanos, and A. K. Katsaggelos, "Multichannel regularized iterative restoration of motion compensated image sequences," *J. Visual Commun. Image Representation*, vol. 7, no. 3, pp. 244–258, Sept. 1996.

[4] M. G. Choi, "Regularization-based encoding and decoding of still-images and video," Ph.D. dissertation, Department of electrical and computer engineering, Illinois Institute of Technology, Chicago, IL, 1998.

[5] N. P. Galatsanos and R. T. Chin, "Digital restoration of multichannel images," *IEEE Trans. Acoustics, Speech Signal Processing*, vol. 37, Mar. 1989.

[6] N. P. Galatsanos, A. K. Katsaggelos, R. T. Chin, and A. D. Hillery, "Least squares restoration of multichannel images," *IEEE Trans. Signal Processing*, vol. 39, pp. 2222–2236, Oct. 1991.

[7] N. P. Galatsanos and A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *IEEE Trans. Image Processing*, vol. 1, pp. 322–336, July 1992.

[8] N. P. Galatsanos, M. N. Wernick, and A. K. Katsaggelos, "Multi-channel image recovery," in *The Handbook of Image and Video Processing*, A. Bovik, Ed. New York: Academic, 2000, pp. 161–174.

[9] M. C. Hong, C. M. Yon, and Y. M. Park, "An efficient real time algorithm to simultaneously reduce blocking and ringing artifacts in compressed video," in *Proc. Int. Conf. on Image Processing*, Kobe, Japan, 1999, pp. 899–903.

[10] "Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbits/s—Video,", Geneva, ISO/IEC 11 172-2, 1993.

[11] "Information technology—Generic Coding of Moving Pictures and Associated Audio: Video,", ISO/IEC 13 818-2, 1994.

[12] "Video Codec for Audiovisual Services at $p \times 64$ kbits,", ITU-T Recommendation H.261.

[13] "Video Coding for low bitrate communication,", ITU-T Recommendation H.263, 1997.

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

[15] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-32, pp. 1243–1245, Dec. 1984.

[16] S. W. Lee, Y. M. Kim, and S. W. Choi, "Fast scene change detection using direct feature extraction from MPEG compressed videos," *IEEE Trans. Multimedia*, vol. 2, pp. 240–254, Dec. 2000.

[17] Y. L. Lee, H. C. Kim, and H. W. Park, "Blocking effect reduction of JPEG images by signal adaptive filtering," *IEEE Trans. Image Processing*, vol. 7, Feb. 1998.

[18] 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.
[19] 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.
[20] H. Musmann and H. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.
[21] T. P. O'Rourke and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, p. 490499, Dec. 1995.
[22] J. M. Ortega and W. C. Reinbolt, *Iterative Solutions to Nonlinear Equations in Several Variables*. New York: Academic, 1970.
[23] 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.
[24] M. Ozkan, T. Erdem, M. I. Sezan, and A. M. Tekalp, "Efficient multiframe Wiener restoration of blurred and noisy image sequences," *IEEE Trans. Image Processing*, vol. 1, pp. 453–476, Oct. 1992.
[25] W. B. Pennebaker and J. L. Mitchel, *JPEG: Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
[26] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-34, pp. 1258–1267, Oct. 1986.
[27] 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.
[28] K. Sauer, "Enhancement of low bit-rate coded images using edge detection and estimation," *Computer Vision Graphics and Image Processing: Graphical Models and Image Processing*, vol. 53, no. 1, pp. 52–62, January 1991.
[29] T. Sikora, "MPEG digital video-coding standard," *IEEE Signal Processing Mag.*, vol. 15, pp. 82–100, Sept. 1997.
[30] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*. New York: Wiley, 1977.
[31] M. Wernick, J. Infusino, and M. Milosevic, "Fast spatio-temporal image reconstruction for dynamic PET," *IEEE Trans. Med. Imaging*, vol. 18, pp. 185–195, 1999.
[32] 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, no. 6, pp. 421–432, Dec. 1993.
[33] ——, "Projection-based spatially adaptive image reconstruction of block transform compressed images," *IEEE Trans. Image Processing*, vol. 4, pp. 896–908, July 1995.
[34] Y. Yang and N. Galatsanos, "Removal of compression artifacts using projections onto convex sets and line process modeling," *IEEE Trans. Image Processing*, vol. 6, pp. 1345–1357, Oct. 1997.
[35] S. Yang, Y.-H. Hu, and D. L. Tull, "Blocking effect removal using robust statistics and line process," in *Proc. IEEE 3rd Workshop on Multimedia Signal Processing*, 1999, pp. 315–320.
[36] S. Yang and Y.-H. Hu, "Coding artifacts removal using biased anisotropic diffusion," in *Proc. Int. Conf. Image Processing*, vol. 2, 1997, pp. 346–349.
[37] ——, "Blocking effect removal using regularization and dithering," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 1998, pp. 415–419.

Mun Gi Choi (S'91–M'92) received the B.S. degree in electrical engineering from the Seoul National University, Seoul, Korea, in 1980 and the M.S. and Ph.D degrees in electrical and computer engineering from Illinois Institute of Technology, Chicago, in 1993 and 1998, respectively.

From 1981 to 1982 and 1982 to 1990, he was with Dae Woo Telecommunication and Centel Electronics Company, Ltd, respectively. From 1997 to 2000, he had served as a senior technical staff of video R&D and wireless R&D at 3Com Corporation. Since January of 2001, he has been a principal of engineering at Actiontec Electronics Ltd, Sunnyvale, CA. His research interests are in the areas of video recovery, third generation of wireless multimedia communication, AAA for 3G. IP network, IP version 6, and multimedia solutions over broad-band network.

Yongyi Yang (M'97) received the B.S. and M.S. degrees in electrical engineering from Northern Jiaotong University, Beijing, China, in 1985 and 1988, respectively, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from Illinois Institute of Technology (IIT), Chicago, in 1992 and 1994, respectively.

He was previously on the faculty of the Institute of Information Science, Northern Jiaotong University. In 1995, he joined the Electrical Engineering Department at IIT, where he is currently an Assistant Professor. His research interests are in image recovery, medical imaging, image and video compression, general areas of signal processing, and applied mathematical and statistical methods. He is a co-author of *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics* (New York: Wiley, 1998).

Nikolas P. Galatsanos (S'84–M'85–SM'95) received the Diploma of Electrical Engineering from the National Technical University of Athens Greece in 1982 and the M.S.E.E. and Ph.D. degrees from the Electrical and Computer Engineering Department, University of Wisconsin-Madison in 1984 and 1989, respectively.

Since the fall of 1989, he has been on the faculty of the Electrical and Computer Engineering Department at the Illinois Institute of Technology, Chicago, where currently he is an Associate Professor and Graduate Program Director. His research interests include inverse problems for visual communication and medical imaging applications, and the application of vector space projections methods to signal and image processing problems. He has coedited a book titled *Image Recovery Techniques for Image and Video Compression and Transmission* (Boston, MA: Kluwer, 1998).

Dr. Galatsanos has served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and currently serves as an Associate Editor for the IEEE SIGNAL PROCESSING MAGAZINE.