Multiframe Blocking-Artifact Reduction for Transform-Coded Video

Bahadir K. Gunturk, Yucel Altunbasak, and Russell M. Mersereau

***Abstract*—A major drawback of block-based still-image or video-compression methods at low rates is the visible block boundaries that are also known as blocking artifacts. Several methods have been proposed in the literature to reduce these artifacts. Most are single image methods, which do not distinguish between video and still images. However, video has a temporal dimension that can lead to better reconstruction if utilized effectively. In this paper, we show how to combine information from multiple frames to reduce blocking artifacts. We derive constraint sets using motion between neighboring frames and quantization information that is available in the video bit stream. These multiframe constraint sets can be used to reduce blocking artifacts in an alternating-projections scheme. They can also be included in existing set-theoretic algorithms to improve their performance by narrowing down the feasibility set. Experimental results show the effectiveness of using these multiframe constraint sets.**

***Index Terms*—Blocking-artifact reduction, dequantization, multiframe image reconstruction, POCS.**

I. Introduction

TRANSFORM coding is a popular and effective compression method for both still images and video sequences, as is evident from its widespread use in international media coding standards such as MPEG, H.263, and JPEG. The motion-compensated image (or the image itself) is divided into blocks and each block is independently transformed by a 2-D orthogonal transform to achieve energy compaction. The most commonly used transform is the discrete cosine transform (DCT). After the block transform, the transform coefficients undergo a quantization step. At low bit-rates, the DCT coefficients are coarsely quantized. This coarse quantization with independent quantization of neighboring blocks gives rise to blocking artifacts: visible block boundaries.

Blocking-artifact reduction methods can be classified into three distinct groups according to their reconstruction approaches. The first group uses low-pass filtering; the filters can be space-invariant [1] or space-varying [2], [3]. The main problem with low-pass filtering is the over-smoothing of images. References [4] and [5] try to avoid this problem by first decomposing images into their frequency subbands, and then filtering the block boundaries in high-frequency subbands. The second group of artifact-reduction methods are statistical estimation methods [6]–[8]. These assume a probabilistic model, and apply maximum *a posteriori* probability (MAP) technique to reduce the artifacts. The final group of methods consists of set-theoretic reconstruction methods [9]–[14]. These methods define constraint sets using observed data or prior knowledge of the solution, and try to reconstruct the original image by using a projections onto convex sets (POCS) technique. The success of set-theoretic methods depends highly on the constraint sets, whose intersection gives the feasibility set: the set of all acceptable solutions [15]. If the constraint sets do not have a small feasibility set, POCS-based algorithms do not perform well. Moreover, incorrect constraint sets can prevent convergence or lead to incorrect solutions.

Temporal information adds another dimension to these methods for video sequences. Ironically, this information is not used effectively, if used at all, for blocking-artifact reduction in video. One method that does use temporal information was proposed by Park and Lee [16]. It makes use of motion vectors to extract the blocking semaphores and employs adaptive spatial filtering to reduce the artifacts. Another method [17] uses space-varying spatial filtering followed by a motion-compensated nonlinear filter.

In this paper, we propose a way to incorporate temporal information in blocking-artifact reduction for video sequences. The proposed method constructs convex constraint sets using the motion between the frames and the quantization information extracted from the video bit stream. This allows us to impose additional constraints on a particular frame using the quantization information of the neighboring frames. The constraint sets defined this way are solely based on the observed data, unlike the smoothness constraint sets that are based on a prior image smoothness assumption. The use of these additional constraint sets improves the performance of existing set-theoretic reconstruction algorithms by narrowing down the feasibility set. It is also possible to reduce blocking artifacts by using only these constraint sets without any smoothness assumption.

In Section II, existing constraint sets used in blocking-artifact reduction methods are reviewed. Multiframe constraint sets are derived in Section III. The details of the POCS-based algorithm are explained in Section IV. Section V presents the experimental results, and conclusions are discussed in Section VI.

II. Existing Constraint Sets Used in Blocking-Artifact Reduction Algorithms

Set-theoretic reconstruction techniques produce solutions that are consistent with the information arising from the observed data or prior knowledge about the solution. Every single piece of information is associated with a constraint set in the solution space, and the intersection of these sets represents the space of acceptable solutions [15]. By using more valid constraint sets, the feasibility set can be made smaller, which means getting closer to the original signal. In the blocking-artifact reduction problem, the well-known constraint set is the quantization-bound

Manuscript received January 6, 2001; revised December 7, 2001. This work was supported in part by the National Science Foundation Award CCR-0113681 and by the Office of Naval Research (ONR) under Award N000140110619. This paper was recommended by Associate Editor A. Tabatabai.

The authors are with the Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: bahadir@ece.gatech.edu; yucel@ece.gatech.edu; rmm@ece.gatech.edu).

Publisher Item Identifier S 1051-8215(02)04284-2.

information. When DCT coefficients are quantized, the exact values are lost, but the upper and lower bounds within which the original DCT coefficients lie can be determined using the quantization step sizes. The quantization constraint set was first used by Zakhor to reduce blocking artifacts [9], and became common to all POCS-based methods [9]–[14]. Another constraint set is based on the range of pixel intensities. For an 8-bit representation, this range is 0–255. These two constraint sets are based on the observed data, but they are not enough for still-image reconstruction *since the decoded image already lies in the feasibility set formed from these two constraint sets.* Therefore, additional constraint sets must be defined based on prior knowledge of the original images. One such constraint set is the smoothness constraint set, and it is used in all set-theoretic blocking-artifact reduction methods. Zakhor assumed that images are bandlimited, and the high-frequency components above a certain cutoff frequency are caused by blocking artifacts [9]. There are two problems with this approach: 1) this is not always a valid assumption since images may contain high-frequency components and 2) an ideal low-pass filter cannot be implemented. Zakhor used a 3×3 low-pass filter, which over-smooths images when applied repetitively, since it is not a projection operator. (Although Zakhor's method is not a POCS-based method in its implementation, we found it more appropriate to classify it among the POCS-based methods because of its design.) A real smoothness-projection operator was proposed by Yang *et al.* [10]. That method constrains the difference between neighboring blocks. They later improved their algorithm by applying adaptive smoothness constraint sets: directional smoothness-constraint sets that do not over-smooth images [11], [12]. Paek *et al.* [13] assumed that the global frequency characteristics of two adjacent blocks are similar to the local ones, and tried to detect and remove the undesired high-frequency components. Other constraint sets, such as ones aimed at reducing ringing artifacts, have also been used in blocking-artifact reduction algorithms [12].

III. Multiframe Constraint Sets

As explained in the previous section, set-theoretic single-frame blocking-artifact reduction methods must define smoothness constraint sets. However, smoothness constraint sets may not always be consistent with the original image. For video sequences there is another way to narrow down the feasibility set: temporal information. In this section, we show how to define additional constraint sets using motion between the frames and the quantization-bound information at those frames. We first establish the relation between two frames using motion. We then consider the MPEG compression stages to define the constraint sets.

We start with the intensity conservation assumption along the motion trajectories. Let $f(x_1, x_2, t)$ denote the intensity of a continuous spatio–temporal video signal at spatial coordinate (x_1, x_2) at time t. Pixel intensities of any two video frames can be related to each other through the motion vectors.[1] Denoting $(M_1, M_2) \equiv (M_1(x_1, x_2, t_k; t_j), M_2(x_1, x_2, t_k; t_j))$ as the motion mapping between the frames at times t_k and t_j, we can write

$$f(x_1, x_2, t_k) = f(x_1 - M_1, x_2 - M_2, t_j). \tag{1}$$

We now proceed by discretizing this relation. Discrete signals are obtained by sampling their continuous versions with a space-time lattice Λ_s. In order to re-obtain continuous signals, discrete signals must be interpolated with an interpolant. Denoting $f(n_1, n_2, j)$ as the sampled version of $f(x_1, x_2, t_j)$, and $h_r(x_1, x_2)$ as the interpolant, this relation can be written as

$$\begin{aligned} f(x_1, x_2, t_j) &= \left[\sum_{(n_1, n_2)} f(n_1, n_2, j)\delta(x_1 - n_1, x_2 - n_2)\right] * h_r(x_1, x_2) \\ &= \int \sum_{(n_1, n_2)} f(n_1, n_2, j)\delta(\xi_1 - n_1, \xi_2 - n_2) \\ &\quad \cdot h_r(x_1 - \xi_1, x_2 - \xi_2)\, d\xi_1\, d\xi_2 \\ &= \sum_{(n_1, n_2)} h_r(x_1 - n_1, x_2 - n_2) f(n_1, n_2, j) \end{aligned} \tag{2}$$

where $\delta(\cdot, \cdot)$ is the 2-D impulse function, and "$*$" represents the convolution operation. The spatial coordinates (n_1, n_2) and the frame number j are integers. The interpolant $h_r(x_1, x_2)$ can be a zero-order, bilinear, or higher-order filter. Substituting (2) into (1), we get

$$f(x_1, x_2, t_k) = \sum_{(n_1, n_2)} h_r(x_1 - M_1 - n_1, x_2 - M_2 - n_2) \cdot f(n_1, n_2, j). \tag{3}$$

Since we are only dealing with digital video, we evaluate $f(x_1, x_2, t_k)$ at integer locations (l_1, l_2). The discrete kth frame is then written as:

$$f(l_1, l_2, k) = \sum_{(n_1, n_2)} h_r(l_1 - M_1 - n_1, l_2 - M_2 - n_2) \cdot f(n_1, n_2, j). \tag{4}$$

In order to simplify the notation and emphasize the point that (M_1, M_2) is a function of frames k and j, corresponding to times t_k and t_j, respectively, we define $h(l_1, l_2, k; n_1, n_2, j) \equiv h_r(l_1 - M_1 - n_1, l_2 - M_2 - n_2)$, and write (4) as

$$f(l_1, l_2, k) = \sum_{(n_1, n_2)} h(l_1, l_2, k; n_1, n_2, j) f(n_1, n_2, j). \tag{5}$$

Equation (5) gives the relation between two different frames. Now we model the operations that take place in the process of MPEG compression (i.e., motion compensation, block-DCT calculation, and quantization). Motion compensation is simply the subtraction of an offset value from $f(l_1, l_2, k)$. Denoting $f_m(l_1, l_2, k)$ as the motion-compensated frame and $\hat{f}(l_1, l_2, k)$ as the predicted frame, we write:

$$f_m(l_1, l_2, k) = f(l_1, l_2, k) - \hat{f}(l_1, l_2, k). \tag{6}$$

This motion-compensated frame is then divided into 8×8 blocks, and each block is separately transformed by a DCT.

[1] Note that these motion vectors are not the MPEG motion vectors. They are used to relate a reference frame to neighboring frames, and therefore, they need to be dense and accurate.

Denoting $F_m(m_1, m_2, k)$, $F(m_1, m_2, k)$, and $\hat{F}(m_1, m_2, k)$ as the block-DCT's of $f_m(l_1, l_2, k)$, $f(l_1, l_2, k)$, and $\hat{f}(l_1, l_2, k)$, respectively, we can write this relation as

$$F_m(m_1, m_2, k) = F(m_1, m_2, k) - \hat{F}(m_1, m_2, k). \quad (7)$$

Here, $F(m_1, m_2, k)$ are the block-DCT coefficients. This block-DCT stage is followed by the quantization process, which can be modeled by the addition of a quantization error $Q(m_1, m_2, k)$. Denoting $d_q(m_1, m_2, k)$ as the quantized DCT coefficients, this can be written as

$$\begin{aligned} d_q(m_1, m_2, k) &= F_m(m_1, m_2, k) + Q(m_1, m_2, k) \\ &= F(m_1, m_2, k) - \hat{F}(m_1, m_2, k) + Q(m_1, m_2, k). \quad (8) \end{aligned}$$

Equation (5) gave the relation between the kth and jth frames. By combining that relation with the MPEG compression relation given in Equation (8), we will be able define constraint sets on the jth frame using the quantization information of another frame k.

Before continuing, we write the block-DCT relation for $f(l_1, l_2, k)$ explicitly. Denoting $L(m) = 8\lfloor m/8 \rfloor$ as the limit function, and $(\cdot)_8$ as the $modulo\,8$ operator, the block-DCT of $f(l_1, l_2, k)$ can be written as

$$F(m_1, m_2, k) = \sum_{l_1=L(m_1)}^{L(m_1)+7} \sum_{l_2=L(m_2)}^{L(m_2)+7} \cdot K((m_1)_8, (m_2)_8; l_1, l_2) f(l_1, l_2, k) \quad (9)$$

where $K((m_1)_8, (m_2)_8; l_1, l_2)$ is the DCT kernel, which is given by

$$K(m_1, m_2; l_1, l_2) = k_{m_1} k_{m_2} \cdot \cos\left(\frac{\pi(2l_1+1)m_1}{16}\right) \cos\left(\frac{\pi(2l_2+1)m_2}{16}\right) \quad (10)$$

with k_{m_1} and k_{m_2} being the normalization constants

$$k_{m_i} = \begin{cases} \frac{1}{2\sqrt{2}}, & m_i = 0 \\ \frac{1}{2}, & m_i \neq 0 \end{cases} \quad \text{for } i = \{1, 2\}. \quad (11)$$

Now we will combine (5), (8), and (9). Substituting (5) into (9), and changing the order of summations gives

$$F(m_1, m_2, k) = \sum_{(n_1, n_2)} \left(\sum_{l_1=L(m_1)}^{L(m_1)+7} \sum_{l_2=L(m_2)}^{L(m_2)+7} K((m_1)_8, (m_2)_8; l_1, l_2) \cdot h(l_1, l_2, k; n_1, n_2, j) f(n_1, n_2, j) \right). \quad (12)$$

Defining

$$h_K(m_1, m_2, k; n_1, n_2, j) \equiv \sum_{l_1=L(m_1)}^{L(m_1)+7} \sum_{l_2=L(m_2)}^{L(m_2)+7} K((m_1)_8, (m_2)_8; l_1, l_2) \cdot h(l_1, l_2, k; n_1, n_2, j), \quad (13)$$

we can write (12) as

$$F(m_1, m_2, k) = \sum_{(n_1, n_2)} h_K(m_1, m_2, k; n_1, n_2, j) f(n_1, n_2, j). \quad (14)$$

Substituting (14) into (8), we get

$$d_q(m_1, m_2, k) = \sum_{(n_1, n_2)} h_K(m_1, m_2, k; n_1, n_2, j) f(n_1, n_2, j) - \hat{F}(m_1, m_2, k) + Q(m_1, m_2, k). \quad (15)$$

Equation (15) is the key in this paper. It shows the connection between jth frame $f(n_1, n_2, j)$ and the quantized DCT coefficients $d_q(m_1, m_2, k)$ of another frame k. By using the quantization bound information about $d_q(m_1, m_2, k)$, we are able to define constraint sets on $f(n_1, n_2, j)$. Although the exact value of $Q(m_1, m_2, k)$ is not known, the range within which the original DCT coefficient lies can be extracted from the MPEG bit stream. Defining $b_l(m_1, m_2, k)$ and $b_u(m_1, m_2, k)$ as the lower and upper bounds of the DCT coefficient at spatio–temporal location (m_1, m_2, k), the constraint set $C(m_1, m_2, k)$ can be written as n (16), shown at the bottom of the page.

This equation shows how to define constraint sets on any frame j using the quantization information from another frame k. Therefore, we can use any number of constraint sets while reconstructing $f(n_1, n_2, j)$. These constraint sets are based on the observed data unlike the smoothness constraint sets. By projecting the "blocky" frame onto these multiframe constraint sets, the blocking artifacts can be reduced significantly. These constraint sets can also be used with other set-theoretic blocking-artifact reduction algorithms. The next section gives the projection operator corresponding these constraint sets, and provides the details of the algorithm.

IV. Multiframe Blocking-Artifact Reduction

In order to construct the constraint sets on a reference frame j, we first compute the motion-compensated kernel function (MCKF) $h_K(m_1, m_2, k; n_1, n_2, j)$ between the reference frame j and another neighboring frame k. This requires accurate motion estimates, since incorrect motion estimation may lead to constraint sets that are not consistent with the original block-artifact-free frame. (It should be noted that the motion

$$C(m_1, m_2, k) = \left\{ f(n_1, n_2, j) \colon \left[\sum_{(n_1, n_2)} h_K(m_1, m_2, k; n_1, n_2, j) f(n_1, n_2, j) - \hat{F}(m_1, m_2, k) \right] \in [b_l(m_1, m_2, k), b_u(m_1, m_2, k)] \right\} \quad (16)$$

TABLE I
HIERARCHICAL BLOCK-MATCHING PARAMETERS

Level	Max. Displ.		Window Size		Filter Size		Step Size	Accuracy
	hor.	vert.	hor.	vert.	hor.	vert.		
1	13	13	64	64	5	5	14	1
2	5	5	28	28	5	5	6	1
3	2	2	12	12	3	3	3	0.25

vectors available in the MPEG bit stream are generally not accurate enough, since they are not selected to measure true object motion. Therefore we need to compute true motion vectors after decoding the frames. The details of the motion estimation and MCKF computation are given in Section V.) Once the MCKF is computed, frame j is projected onto the constraint sets using the projection operator $P_{C(m_1, m_2, k)}[\cdot]$, as in (17), shown at the bottom of the page where $f \bullet h_K \equiv \sum_{(n_1, n_2)} h_K(m_1, m_2, k; n_1, n_2, j) f(n_1, n_2, j)$. The dependencies on (m_1, m_2, k) are dropped from b_l, b_u, and $\hat{F}$ for convenience. Since this operation is valid for any k, we can construct an arbitrary number of sets $C(m_1, m_2, k)$ to constrain the solution space. We also note that (16) and (17) are valid regardless of the macroblock mode used in encoding. For intramode macroblocks, no motion compensation is done and, therefore, $\hat{f}(l_1, l_2, k)$ is zero. For intermode macroblocks, $\hat{f}(l_1, l_2, k)$ can be predicted from the previous intracoded frame or bidirectionally.

One way of implementing the algorithm is as follows.

1) Choose the reference frame j to be reconstructed.
2) Choose another frame k, and compute the motion vectors from frame k to frame j. (Do not use the motion vectors extracted from the MPEG bit stream.)
3) Determine the MCKF $h_K(m_1, m_2, k; n_1, n_2, j)$, and define constraint sets $C(m_1, m_2, k)$ according to (16) for each pixel site (m_1, m_2, k) where the motion estimation is believed to be accurate. Do not define a constraint set at a pixel where there is a little confidence in the motion estimate.
4) For all sites (m_1, m_2, k) where the constraint sets have been defined, project the reference frame j onto the constraint set $C(m_1, m_2, k)$ using (17).
5) Make sure that the reconstructed image has pixel intensities between 0 to 255. If there are any of them out this range, project them to the closest bound (0 or 255).
6) Stop, if the stopping criterion is reached; else, choose another frame k, and go to step 2.

V. RESULTS

To demonstrate the efficacy of the proposed method, several experiments with real video sequences have been performed. Before presenting these experimental results we need to clarify the details concerning motion estimation and the computation of the MCKF.

A. Estimating Motion

Motion estimation is an important part of the algorithm since it directly affects the constraint sets. If motion is not estimated accurately, incorrect constraint sets can be imposed on the solution. In our experiments we used the hierarchical block matching (HBM) algorithm of Bierling [18] to compute the nonuniform translational motion between frames. The assumption of locally translational motion is quite effective when warping effects are small. We used three levels of hierarchy with mean absolute difference (MAD) as the matching criterion between measurement blocks. The parameters are given in Table I. In that table, the maximum horizontal/vertical displacement gives the search range in terms of number of pixels. The window size is the size of the matching blocks where the MAD is computed. The filter size is the support of a Gaussian low-pass filter that is applied before determining the motion vectors. The variance of the Gaussian filter is set to one-half the support size. The step size is the distance between neighboring pixels for which motion is estimated. The motion vectors for the pixels in between are bilinearly interpolated from these computed motion estimates. (We did not compute motion vectors at every pixel in order to speed up the implementation.) In the final level of estimation, motion vectors are sought with one-quarter-pixel accuracy by subsampling pixels. It should also be noted that in order not to impose any incorrect constraint set on the reconstructed frame, we did not use the constraint sets at locations (m_1, m_2, k) where the MAD was greater than a threshold. In the experiments we chose that threshold as four.

$$P_{C(m_1, m_2, k)}[f(n_1, n_2, j)] = \begin{cases} f(n_1, n_2, j) + \dfrac{(b_l + \hat{F} - f \bullet h_K) h_K(m_1, m_2, k; n_1, n_2, j)}{\sum\limits_{(n_1, n_2)} |h_K(m_1, m_2, k; n_1, n_2, j)|^2}, & f \bullet h_K < b_l + \hat{F} \\ f(n_1, n_2, j) + \dfrac{(b_u + \hat{F} - f \bullet h_K) h_K(m_1, m_2, k; n_1, n_2, j)}{\sum\limits_{(n_1, n_2)} |h_K(m_1, m_2, k; n_1, n_2, j)|^2}, & f \bullet h_K > b_u + \hat{F} \\ f(n_1, n_2, j), & \text{elsewhere} \end{cases} \quad (17)$$

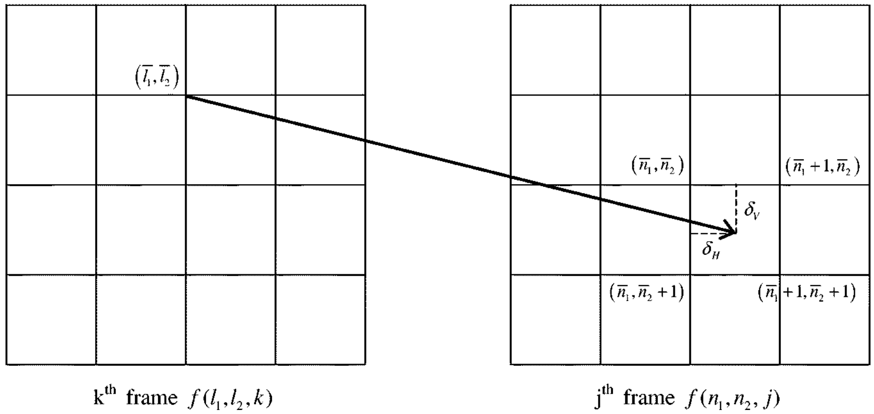

Fig. 1. Computing the MCKF.

B. Computing the MCKF

In order to find the MCKF $h_K(m_1, m_2, k; n_1, n_2, j)$, we first need to determine the interpolant $h_r(x_1, x_2)$. In our experiments we used a bilinear interpolant. Referring to Fig. 1, if the motion vector from spatio–temporal location $(\bar{l}_1, \bar{l}_2, k)$ points to the jth frame at coordinates $(\overline{n}_1 + \delta_H, \overline{n}_2 + \delta_V)$, then we can write the following relation:

$$f(\bar{l}_1, \bar{l}_2, k) = (1-\delta_H)(1-\delta_V) f(\overline{n}_1, \overline{n}_2, j) + (\delta_H)(1-\delta_V) \cdot f(\overline{n}_1 + 1, \overline{n}_2, j) + (1-\delta_H)(\delta_V) f(\overline{n}_1, \overline{n}_2 + 1, j) + (\delta_H)(\delta_V) f(\overline{n}_1 + 1, \overline{n}_2 + 1, j) \tag{18}$$

where $\delta_H \in [0, 1)$, and $\delta_V \in [0, 1)$. We used the notation $(\bar{l}_1, \bar{l}_2)$ and $(\overline{n}_1, \overline{n}_2)$ for the spatial coordinates to distinguish them from the generic coordinates (l_1, l_2) and (n_1, n_2). Equation (18) reveals how to find the mapping $h(l_1, l_2, k; n_1, n_2, j)$ in (5) for all coordinates. After finding $h(l_1, l_2, k; n_1, n_2, j)$, all we need to do is take the block-DCT as in (13) to find the MCKF $h_K(m_1, m_2, k; n_1, n_2, j)$.

C. Experimental Results

We compressed the Susie test sequence at 112 kbits/s using an MPEG-1 encoder. The original frame 13 and the compressed frame 13 are given in Figs. 2 and 3. We then computed the motion between frame 13 and the frames 14–17. (Picture modes for frames 13–17 are P, B, B, P, and B, respectively.) Frame 13 was projected onto constraints sets defined using frames 14–17. After one iteration, the image in Fig. 5 was obtained. When compared to the compressed frame (Fig. 3) and Zakhor's method with one iteration (Fig. 4), there is a significant reduction in blocking artifacts (without over-smoothing) in the multiframe method, but there are still regions with visible block boundaries. From a set-theoretic point of view, this means that the feasibility set is not small enough to get close to the original image. The results can be improved by imposing other constraint sets. These additional constraint sets can be constraint sets from other frames as we proposed, or constraint sets such as assuming spatial smoothness. In this case, we chose to impose a smoothness constraint. Fig. 6 shows the result if Zakhor's smoothness filter is also included in the iteration. The same experiments are repeated for the Foreman sequence. The results are shown in Figs. 7–11.

Fig. 2. Susie original.

Fig. 3. Susie compressed at 112 kbits/s.

VI. Discussion

The proposed multiframe blocking-artifact reduction method exploits temporal information encapsulating the motion be-

Fig. 4. Zakhor's method (one iteration).

Fig. 5. Multiframe (one iteration).

Fig. 6. Multiframe + Zakhor (one iteration).

Fig. 7. Foreman original.

Fig. 8. Foreman compressed at 112 kbits/s.

Fig. 9. Zakhor's method (one iteration).

tween frames and the quantization bounds available in the video bit stream. It defines convex constraint sets based on the observed data and uses an alternating projections scheme to reconstruct the original image. It is a general tool in the sense that it can easily be combined with other set-theoretic methods. However, the method requires accurate motion estimates, and it has high computational complexity. This is not a significant drawback for offline applications where video quality is the

Fig. 10. Multiframe (one iteration).

Fig. 11. Multiframe + Zakhor (one iteration).

main concern. It can also be used in real-time applications by implementing the algorithm as part of a dedicated hardware solution in set-top boxes.

REFERENCES

[1] H. C. Reeves and J. S. Lim, "Reduction of blocking effects in image coding," *Opt. Eng.*, vol. 23, pp. 34–37, 1984.

[2] G. 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.

[3] K. Sauer, "Enhancement of lower bit-rate coded images using edge detection and estimation," *Comput. Vis. Graph. Image Processing: Graphical Models Image Processing*, vol. 53, no. 1, pp. 52–62, Jan. 1991.

[4] Z. Xiong, M. T. Orchard, and Y.-Q. Zhang, "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 433–437, Apr. 1997.

[5] H. Choi and T. Kim, "Blocking-artifact reduction in block-coded images using wavelet-based subband decomposition," *IEEE Trans. Circuits Syst. Video Technolo.*, vol. 10, pp. 801–805, Aug. 2000.

[6] R. L. Stevenson, "Reduction of coding artifacts in transform image coding," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 5, 1993, pp. 401–404.

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

[8] T. Ozcelik, J. C. 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, 1995.

[9] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 91–95, 1992.

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

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

[12] Y. Yang and N. P. 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.

[13] H. Paek, R. Kim, and S. Lee, "On the POCS-based postprocessing technique to reduce blocking artifacts in transform coded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 358–367, June 1998.

[14] Y. Jeong, I. Kim, and H. Kang, "A practical projection-based postprocessing of block-coded images with fast convergence rate," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 617–623, June 2000.

[15] P. L. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, pp. 182–208, Feb. 1993.

[16] H. W. Park and Y. L. Lee, "A postprocessing method for reducing quantization effects in low bit-rate moving picture coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 161–171, Feb. 1999.

[17] C. Derviaux, F.-X. Coudoux, M. G. Gazalet, and P. Corlay, "Blocking artifact reduction of DCT coded image sequences using visually adaptive postprocessing," in *Proc. Int. Conf. Image Processing*, vol. 2, 1996, pp. 5–8.

[18] M. Bierling, "Displacement estimation by hierarchical blockmatching," in *Proc. SPIE Visual Communications and Image Processing '88*, 1988, pp. 942–951.