A Unified Approach to Restoration, Deinterlacing and Resolution Enhancement in Decoding MPEG-2 Video

Forchhammer, Søren; Martins, Bo

Published in:
I E E E Transactions on Circuits and Systems for Video Technology

Link to article, DOI:
10.1109/TCSVT.2002.803227

Publication date:
2002

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Forchhammer, S., & Martins, B. (2002). A Unified Approach to Restoration, Deinterlacing and Resolution Enhancement in Decoding MPEG-2 Video. I E E E Transactions on Circuits and Systems for Video Technology, 12(9), 803-811. DOI: 10.1109/TCSVT.2002.803227

Transactions Letters

A Unified Approach to Restoration, Deinterlacing and Resolution Enhancement in Decoding MPEG-2 Video

Bo Martins and Søren Forchhammer

Abstract—**The quality and spatial resolution of video can be improved by combining multiple pictures to form a single superresolution picture. We address the special problems associated with pictures of variable but somehow parameterized quality such as MPEG-decoded video. Our algorithm provides a unified approach to restoration, chrominance upsampling, deinterlacing, and resolution enhancement. A decoded MPEG-2 sequence for interlaced standard definition television (SDTV) in 4 : 2 : 0 is converted to: 1) improved quality interlaced SDTV in 4 : 2 : 0; 2) interlaced SDTV in 4 : 4 : 4; 3) progressive SDTV in 4 : 4 : 4; 4) interlaced high-definition TV (HDTV) in 4 : 2 : 0; and 5) progressive HDTV in 4 : 2 : 0. These conversions also provide features as freeze frame and zoom. The algorithm is mainly targeted at bit rates of 4–8 Mb/s. The algorithm is based on motion-compensated spatial upsampling from multiple images and decimation to the desired format. The processing involves an estimated quality of individual pixels based on MPEG image type and local quantization value. The mean-squared error (MSE) is reduced, compared to the directly decoded sequence, and annoying ringing artifacts including mosquito noise are effectively suppressed. The superresolution pictures obtained by the algorithm are of much higher visual quality and have lower MSE than superresolution pictures obtained by simple spatial interpolation.**

Index Terms—**Deinterlacing, enhanced decoding, motion-compensated processing, MPEG-2, SDTV to HDTV conversion, video decoding.**

I. Introduction

MPEG-2 [1] is currently the most popular method for compressing digital video. It is used for storing video on digital versatile disks (DVDs) and it is used in the contribution and distribution of video for TV. We base this paper on the MPEG reference software encoder [2] for which a bit rate of 5–7 Mb/s yields a quality which is equivalent to (analog) distribution phase alternating line (PAL) TV quality. Lower bit rates are also used in TV distribution to save bandwidth and because professional encoders may provide better quality than the reference software.

At these bit rates, a sequence decoded from an MPEG-2 bitstream is of lower quality than the original digital sequence in terms of sharpness and color resolution but still acceptable (except for very demanding material). This overall reduction of quality is less annoying to a human observer than the artifacts typically found in compressed video. The most annoying artifacts are *ringing artifacts*[1] and in particular *mosquito noise*, which occurs when the appearance of the ringing changes from picture to picture.

The primary goal of this paper is to improve MPEG-2 decoding, or rather to postprocess the decoded sequence re-using information in the MPEG-2 bitstream to obtain a sequence of higher fidelity, especially with regard to the artifacts. The resulting output is a sequence in the same format as the directly decoded one, which in our case is interlaced standard TV in 4 : 2 : 0. In addition, we demonstrate how the approach can be used to obtain progressive (deinterlaced) or high-definition TV (HDTV) from the same bitstream. This also facilitates features such as frame freeze and zoom.

Previous work on postprocessing includes projections onto convex sets (POCS) [3] and regularization [4]. For low-bit-rate (high compression) JPEG-compressed still images and MPEG-1-coded moving pictures, the main artifact is *blocking*, i.e., visible discontinuities at coding block boundaries. This artifact can be dealt with efficiently using the POCS framework [5], as well as by other methods [6]. By regularization, POCS constraints can be combined with "soft" assumptions about the sequence. Thus, Choi *et al.* [4] restored very-low bit-rate video encoded by H.261 and H.263 according to the following desired (soft) properties: 1) smoothness across block boundaries; 2) small distance between the directly decoded sequence and the reconstructed sequence; and 3) smoothness along motion trajectories. Elad and Feuer [7] presented a unified methodology for superresolution restoration requiring explicit knowledge of parameters as warping and blurring. As this knowledge is not available in our case, we do not take the risk of processing based on estimating such parameters. Patti *et al.* [8] also addressed the superresolution problem in a general setting modeling the system components. They applied POCS performing projections for each pixel of each reference image in each iteration. Recently [9] this approach was modified to obtain superresolution from images of an MPEG-1 sequence captured by a specific video camera. Projections were carried

Manuscript received December 1, 1999; revised May 2, 2002. This work was supported in part by The Danish National Centre for IT Research. This paper was recommended by Associate Editor A. Tabatabai.

B. Martins was with the Department of Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark. He is now with Scientific-Atlanta Denmark A/S, DK-2860 Søborg, Denmark (e-mail: bo.martins@sciatl.com).

S. Forchhammer is with Research Center COM, 371, Technical University of Denmark, DK-2800 Lyngby, Denmark (e-mail: sf@com.dtu.dk).

Publisher Item Identifier 10.1109/TCSVT.2002.803227.

[1] Ringing artifacts are caused by the quantization error of high-frequency content, e.g., at edges. They appear as ringing adjacent to the edge.

1051-8215/02$17.00 © 2002 IEEE

out in the transform domain. Our goal is to develop simpler techniques (which could be combined with POCS).

The starting point of our work is the sequence decoded by an ordinary MPEG-2 decoder [2]. The material to be processed in this paper is of higher quality than MPEG-1 material or the low-bit-rate material of [4]. Consequently, there is a higher risk of degrading the material. Enforcing assumptions of smoothness of the material will almost surely lead to a decrease of sharpness. The basic idea of our restoration scheme is to apply a conservative form of filtering along motion trajectories utilizing the assumed quality of the pixels on each trajectory. The assumed quality of each pixel in the decoded sequence is given by the MPEG picture structure (i.e., what type of motion compensation is applied) and the quantization step size for the corresponding macroblock.

The algorithm has two steps. In the first step, a superresolution version (default is quadruple resolution) of each directly decoded picture[2] is constructed. In the second step, the superresolution picture is decimated to the desired format. Depending on the degree of decimation of the chrominance and luminance in the second step, the problem addressed is one of restoration, chrominance upsampling, deinterlacing, or resolution enhancement, e.g., conversion to HDTV. The aim in restoration is to enhance the decoding quality. For the other applications, the resolution is also enhanced.

In the first part of the upsampling, directly decoded pixels are placed very accurately in a superresolution picture before further processing. This approach is motivated by the fact that the individual pictures of the original sequence are undersampled [9], [10]. We do not want to trade resolution for improved peak signal-to-noise ratio (PSNR) by spatial filtering at this stage so the noise reducing filtering is deferred to the decimation step.

The paper is organized as follows. In Section II, a quality value is assigned to each pixel in the decoded sequence. Part one (upsampling) of our enhancement algorithm is described in Section III. The second part (decimation) is described in Section IV. Results on a number of test sequences are presented in Section V.

II. Processing Based on MPEG-Quality

MPEG-2 [1] partitions a picture into 16×16 blocks of picture material (*macroblocks*). A macroblock is usually predicted from one or more reference pictures. The different types of pictures are referred to as I, P, and B pictures. I pictures are intracoded, i.e., no temporal prediction. Macroblocks in P pictures may be unidirectionally predicted and macroblocks in B pictures may be uni- or bidirectionally predicted. (Macroblocks in B and P pictures may also be intracoded as macroblocks in I-pictures.)

The error block, resulting from the prediction, is partitioned into four luminance and two, four, or eight chrominance blocks of 8 × 8 pixels, depending on the format. For the 4 : 2 : 0 format, each macroblock has two chrominance blocks. The discrete cosine transform (DCT) is applied to each 8 × 8 block. The DCT coefficients are subjected to scalar quantization before being coded to form the bitstream.

A. Quality Measure for Pixels in an MPEG Sequence

From the MPEG code stream, the type (I, P, or B) and the quantization step size are extracted for each macroblock. Based on this information, we shall estimate a quality parameter for each pixel which is used in a motion-compensated (MC) filtering. MPEG specifies the code-stream syntax but not the encoder itself. Our work is based on the reference MPEG-2 software encoder [2], for which the quantizers may be characterized as follows. The nonintra quantizer used for DCT coefficient $C(i,\, j)$ is (very close to) a uniform quantizer with quantization step q_s and a deadzone of $2q_s$ around zero. The intra quantizer used for DCT coefficient $C(i,\, j)$ has a deadzone of 5/4 $q_s(i,\, j)$ around zero. For larger values, it is a uniform quantizer with quantization step $q_s(i,\, j)$, and the dequantizer reconstruction point has a bias of 1/8 $q_s(i,\, j)$ toward zero. In [2], as is usually the case, all DCT coefficients in all blocks are being quantized independently as scalars.

The mean-squared error (MSE) caused by the quantization depends on the distribution of $C(i,\, j)$. This distribution varies with the image content and is hard to estimate accurately. We may approximate the expected error by the expression for a uniform distribution of errors, within each quantization interval, resulting from a uniform quantizer with quantization step q_s applied to $C(i,\, j)$

$$E\left\{\left(C(i,\, j) - C\left(\hat{i},\, j\right)\right)^2\right\} = \frac{1}{12}\, q_s^2. \tag{1}$$

This expression may underestimate the error as it neglects the influence of the dead zone, and it may overestimate the error as the distribution of $C(i,\, j)$ is usually quite peaked around zero, especially for the high frequencies.

The DCT transform is unitary (when appropriate scaling is applied). Thus, the sum of squares over a block is the same in the DCT and spatial domains. Applying this to the quantization errors and introducing the expected values gives the following relationship for each DCT block:

$$\sum_{m=0,\, n=0}^{m=7,\, n=7} E\left\{\left(X(m,\, n) - \hat{X}(m,\, n)\right)^2\right\} = \sum_{i=0,\, j=0}^{i=7,\, j=7} E\left\{\left(C(i,\, j) - \hat{C}(i,\, j)\right)^2\right\} \tag{2}$$

where the DCT coefficients $C(i,\, j)$ are scaled as specified in [1] (Annex A) and $X(m,\, n)$ denotes the pixel value variables. As an approximation, we assume that the expected squared quantization errors are the same for all the pixel positions $(m,\, n)$ within the DCT block. Based on this assumption, the expected value of the squared error for pixel $x(m,\, n)$ is given by

$$E\left\{\left(X(m,\, n) - \hat{X}(m,\, n)\right)^2\right\} = \frac{1}{64} \sum_{i=0,\, j=0}^{i=7,\, j=7} E\left\{\left(C(i,\, j) - \hat{C}(i,\, j)\right)^2\right\} \tag{3}$$

for all $n,\, m$ within the DCT block having coefficients $C(i,\, j)$.

[2] In this paper, all pictures are field pictures.

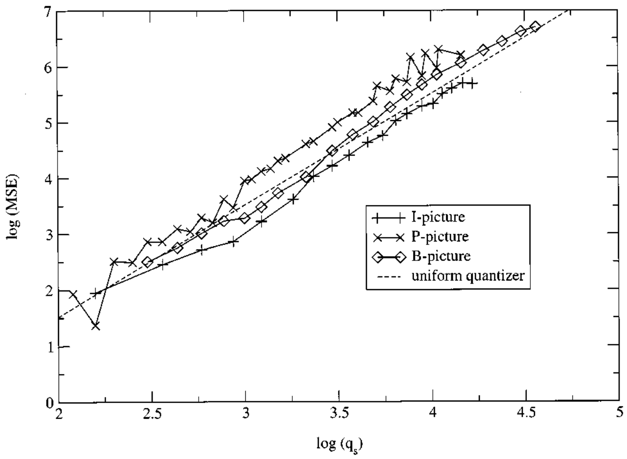

Fig. 1. MSE measured for sequence *table* as a function of the quantization step size q_s (depicted using natural logarithms). For intra pictures, q_s is defined as the quantization step size for the DCT coefficient at (1, 1).

Fig. 1 depicts the logarithm of the MSE as a function of $\log(q_s)$ for the luminance component of I, B, and P pictures. The figure reflects the fact that bidirectional prediction is better than unidirectional prediction, and that intra pictures and nonintra pictures are different. It is noted that we can use the expression (1) as a general approximation for the MSE of picture type y as long as we replace q_s with $q_s r_y$, where r_y is a constant which depends on the picture type, i.e.

$$E\left\{\left(X(m,\,n)-\hat{X}(m,\,n)\right)^2\right\}=\frac{1}{12}\,q^2, \qquad q\triangleq q_s r_y. \quad (4)$$

From the data in Fig. 1, we measure $r_B=1$, $r_P=1.2$, and $r_I=0.9$. These values are used in all the experiments reported. The intra and nonintra quantization matrices used [2] are different. This is, in part, addressed by the values of r_y. [The value of r_I was measured with q_s defined as the quantization step size for $C(1,\,1)$.] The normalized quantization parameters, q in (4) are used as the quality value we assign to each pixel within the block. This measure is only used for relative comparisons and not as an absolute measure. It could be improved by taking the specific frequency content into account, as well as the precise quantization for each coefficient.

In general, pixels in the interior of an 8 × 8 DCT block have a smaller MSE than pixels on the border. We could assign a different value of r_y for interior pixels and pixels on the border. Experiments lead to our decision of ignoring the small difference at our (high) bit rates and as an approximation use the same quality value (4) for all pixels in a block.

III. Upsampling to Superresolution Using Motion Compensation

To process a given (directly decoded) picture we combine the information from the current frame and the N_f previous frames and the N_f subsequent frames, where N_f is a parameter and each frame consists of two field pictures. We first describe how to align pixels of the current picture at time t with pixels of one of the reference pictures using motion estimation. Section III-A then describes how to combine the information from all the reference pictures to form a single superresolution picture at t. The term *superresolution picture* is used to refer to the initial MC upsampled high-resolution image. An overview of the algorithm is given in Fig. 2.

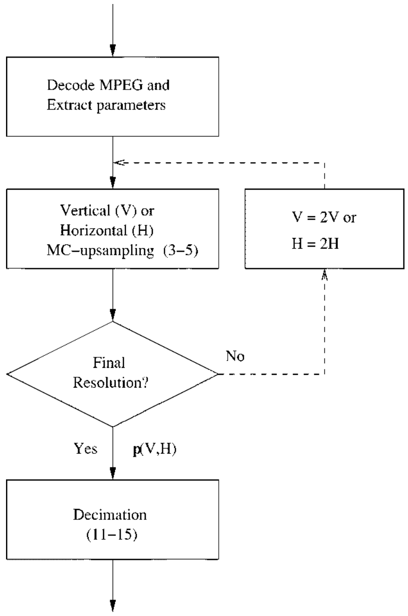

Fig. 2. Overview block diagram. MC upsampling alternates between doubling the resolution vertically and horizontally. Then final step is decimation to the desired format. Equation numbers are given in (). Dashed line marks control flow.

The motion field, relative to one of the reference pictures, is determined on the directly decoded sequence by block-based motion estimation using blocks of size 8 × 8. This block size is our compromise between larger blocks for robustness and smaller blocks for accuracy, e.g., at object boundaries. A motion vector is calculated at subpixel accuracy for each pixel x_0 of the current picture relative to the reference field picture considered. Based on the position of x_0 and the associated motion vector, one pixel x_r' shall be chosen in the reference picture.

The motion vector is found by searching the reference picture for the best match of the 8×8 block, which has x_0 positioned as the lower-right of the four center pixels. The displacements are denoted by $(m+\Delta m,\,n+\Delta n)$, where $(m,\,n)$ is the integer and $(\Delta m,\,\Delta n)$ the (positive) fractional part of the displacement relative to the position in the current picture. $m+\Delta m$ is the vertical displacement. For a given candidate vector $(m+\Delta m,\,n+\Delta n)$, each pixel x of the 8×8 block is matched against an estimated value $\hat{x}$ which is formed by bilinear interpolation of four neighboring pixels $x_r(m,\,n)$, $x_r(m+1,\,n)$, $x_r(m,\,n+1)$, and $x_r(m+1,\,n+1)$ in the reference picture

$$\begin{aligned}\hat{x}=\lfloor&(1-\Delta m)(1-\Delta n)\,x_r(m,n)+(1-\Delta m)\Delta n\,x_r(m,n+1)\\&+\Delta m(1-\Delta n)\,x_r(m+1,n)+\Delta m\Delta n\,x_r(m+1,n+1)\rfloor \quad (5)\end{aligned}$$

where $x_r(m,\,n)$ is the pixel in the reference picture displaced $(m,\,n)$ relative to the pixel x in the current picture. (The coordinate systems of the two pictures are aligned such that the positions of the pixels coincide with the lattice given by the integer coordinates.) The subpixel resolution of the motion field, specified vertically by V and horizontally by H, determines the allowed values of Δm and Δn: $\Delta m=0,\,1/V,\,\ldots,\,(V-1)/V$

and $\Delta n = 0, 1/H, \ldots, (H-1)/H$. The best motion vector $(m' + \Delta m', n' + \Delta n')$ is defined as the candidate vector that minimizes the sum of the absolute differences ($|x - \hat{x}|$) taken over the 64 pixels of the block. (How the set of candidate vectors is determined is described in Section III-C.) Let (m'_r, n'_r) be the absolute coordinate of pixel x'_r in the reference picture obtained by displacing the position of the current pixel x_0 by the integer part (m', n') of the best motion vector. The pixel value of x'_r is now perceived as a (quantized) sample value of a pixel at position $((m'_r - m' - \Delta m')V, (n'_r - n' - \Delta n')H)$ in a superresolution picture at time t which has V times the number of pixels vertically and H times the number horizontally relative to the directly decoded picture.

It is not sufficient, though, to find the best motion vector according to the matching criterion as there is no guarantee this is a *good* match. The following criteria is used to decide for each x'_r whether it shall actually be placed in the superresolution picture. We may look at the problem as a lossless data compression problem (inspired by the minimum description length principle [11]). Let there be two alternative predictive descriptions of the pixels of the current 8×8 block, one utilizing a block of the reference picture and one which does not. If the best compression method that utilizes the reference block is better than the best method which does not, then we rely on the match. In practice, we do not know the best data compression scheme, but instead some of the best compression schemes in the literature may be used. For lossless still-image coding, we use JPEG-LS [12]. For lossless compression utilizing motion compensation, we chose the technique in [13], which may be characterized as JPEG-LS with motion compensation. For simplicity, the comparison is based on the sum of absolute differences. The JPEG-LS predictor [12] is given by

$$\hat{x} = \begin{cases} \min(a, b), & \text{if } c \geq \max(a, b) \\ \max(a, b), & \text{if } c \leq \min(a, b) \\ a + b - c, & \text{otherwise} \end{cases} \tag{6}$$

where a denotes the pixel to the left of x, b denotes the pixel on top of x, and c the top-left pixel.

We compare the (intra picture) JPEG-LS predictor and the best MC bilinear predictor (5). If the former yields a better prediction of the pixels of the 8×8 surrounding block, we leave the superresolution pixel undefined (or unchanged) by not inserting (or modifying) a MC pixel at the position $((m'_r - m' - \Delta m')V, (n'_r - n' - \Delta n')H)$.

Checking the match reduces the risk of errors in the motion compensation process, e.g., at occlusions. Occlusions are also handled by performing the motion compensation in both directions time wise, and by performing motion compensation at pixel level. This leads to a fairly robust handling of occlusions to within 3–4 pixels of the edge.

A. Forming the Superresolution Picture

The superresolution picture is initially formed by mapping pixels from each of the reference pictures as described above. The implemented block-based motion-compensation scheme is described in Section III-C. If more than one reference pixel maps to the same superresolution pixel, the superresolution pixel is assigned the value of the reference pixel having the smallest value of the normalized quantization parameter q obtained from q_s and the picture type (4). If the pixels are of equal quality (q), the superresolution pixel is set equal to their average value. We do not define a MC superresolution pixel if the best (i.e., smallest) q is significantly larger than the normalized quantization value of the current macroblock in the directly decoded picture.

Pixels of the current directly decoded picture *a priori* have a higher validity than the reference pixels because the exact location in the current picture is known. Let x_2 be a pixel of the directly decoded picture at time t and x_1 a pixel from a reference picture aligned with x_2 within the uncertainty of the motion compensation. To estimate a new (superresolution) pixel value x at the original sample position of x_2, we calculate a weighted value $\hat{x}$ of x_1 and x_2 by

$$\hat{x} = h_1 x_1 + h_2 x_2. \tag{7}$$

The filter coefficients in (7) may be estimated in a training session using original data. The (MSE) optimal linear filter is given by solving the Wiener–Hopf equations

$$\begin{pmatrix} E\{X_1 X_1\} & E\{X_1 X_2\} \\ E\{X_1 X_2\} & E\{X_2 X_2\} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} E\{X X_1\} \\ E\{X X_2\} \end{pmatrix} \tag{8}$$

where X, X_1, and X_2 are the stochastic variables of the pixels in (7). The variables X_1 and X_2 represent quantized pixel values, whereas X represents a (superresolution) pixel at a sample position in the picture with the original resolution. The Wiener filter coefficients could, alternatively, be computed under the constraint that $h_1 + h_2 = 1$ in order to preserve the mean value. In our experiments on actual data applying (8), $h_1 + h_2$ was fairly close to 1, so we just proceeded with these estimates. Given enough training data, the second-order mean values in (8) could be conditioned on the quality of (x_1, x_2), i.e., (q_1, q_2), and on the types of the pictures of x_1 and x_2 as well as other MPEG parameters. In this paper, the picture type is reflected by (4) and the number of free parameters is reduced by fitting a smooth function to the samples $h_1(q_2/q_1)$. We choose the function below as it is monotonically increasing in q_2/q_1 from 0 to 1 and as its behavior can be adjusted by just two parameters as follows:

$$h_1 = 1 - (1 - \alpha)^{(q_2/q_1)^\beta} \tag{9}$$

$$h_2 = 1 - h_1. \tag{10}$$

The parameter α specifies the *a priori* weight that x_1 should carry. The parameter β specifies how much the difference in the qualities of x_1 and x_2 should influence x. The filter (9) has the property that for $0 \leq \alpha \leq 1$, $\beta \geq 0$ and $q_1, q_2 \geq 0$, we have $0 \leq h_1 \leq 1$.

The MC superresolution pixels, which do not coincide with the sample positions in the current image, maintain the quality value they were assigned in the reference picture. Pixels in the original sample positions $(x_2 = x(m, n))$, determined by (7), are assigned the quality value

$$q(m, n) = h_1 q_1 + h_2 q_2. \tag{11}$$

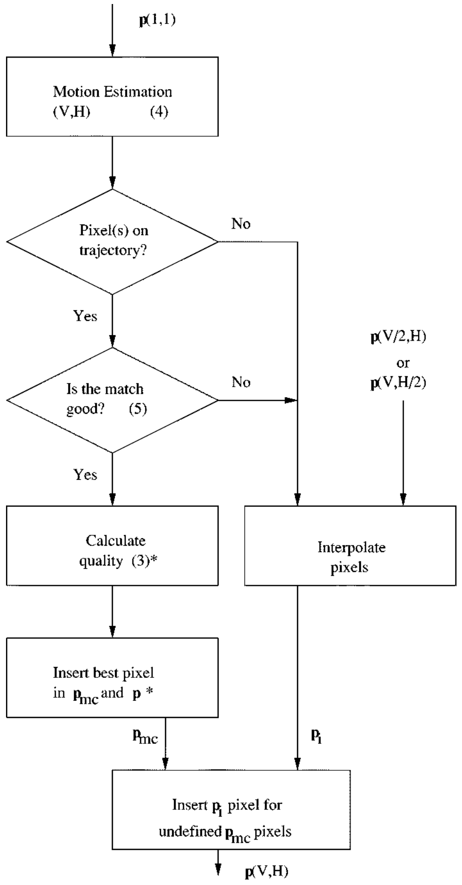

Fig. 3. Block diagram of MC upsampling doubling the vertical or horizontal resolution. Equation numbers are given in (). $*$Averages as expressed by (8)–(10) may also be used.

B. Completing the Superresolution Picture by Interpolation

A block diagram of the MC upsampling is given in Fig. 3. Let $\mathbf{p}_{\mathrm{mc}}(V,\,H)$ denote a superresolution picture created by MC upsampling as described previously. V and H specify the resolution of the motion compensation (5). Usually, some of the pixels of $\mathbf{p}_{\mathrm{mc}}(V,\,H)$ are undefined because there was no accurate match (of adequate quality) in any of the reference pictures. These pixels are assigned values from an interpolated superresolution picture $\mathbf{p}_i(V,\,H)$ having the same resolution as $\mathbf{p}_{\mathrm{mc}}(V,\,H)$. The resulting image is denoted $\mathbf{p}(V,\,H)$. The picture $\mathbf{p}_i(V,\,H)$ is created by a 2 : 1 spatial interpolation of the high-resolution picture $\mathbf{p}(V/2,\,H)$ (if $V = 2H$) or the high-resolution picture $\mathbf{p}(V,\,H/2)$ (if $V = H$). This upsampling alternates between horizontal and vertical 2 : 1 upsampling.

The upsampling process is first initialized by setting $\mathbf{p}(1,\,1)$ equal to the directly decoded picture which has the original resolution. Thereafter, the initialization is completed by defining $\mathbf{p}(2,\,1) = \mathbf{p}_i(2,\,1)$, where $\mathbf{p}_i(2,\,1)$ is created by spatial interpolation of $\mathbf{p}(1,\,1)$. Hereafter, $\mathbf{p}(2,\,2)$, $\mathbf{p}(4,\,2)$, and $\mathbf{p}(4,\,4)$[3] may be created in turn building up the resolution, alternating between horizontal and vertical 2 : 1 upsampling.

The odd samples being interpolated in the upsampled picture are obtained with a symmetric finite-impulse response (FIR) filter h_c used in the software coder [2] for 4 : 2 : 2–4 : 4 : 4 conversion

$$h_c = (\ldots, 0, 0.082, -0.203, 0.621, 0.621, -0.203, 0.082, 0, \ldots). \tag{12}$$

Each pixel of the resulting superresolution picture $\mathbf{p}(V,\,H)$ is assigned the attribute of whether it was determined by motion compensation $\mathbf{p}_{\mathrm{mc}}(V,\,H)$ or interpolation $\mathbf{p}_i(V,\,H)$. The MC pixels also maintain their quality value determined by (4) [and possibly modified by (11)] as an attribute.

C. Speedup of Motion Compensation

The following scheme is applied to speed up the estimation of the $4N_f + 1$ high-resolution motion fields that are required for the $4N_f + 1$ reference pictures relative to the current picture. The very first motion field (estimating the displacement of pixels of the other field of the current frame relative to the current field picture) is found by an exhaustive search within a small rectangular window (±3 vertically and ±7 horizontally). For each of the remaining $4N_f$ reference pictures, we initially *predict* the motion field before actually estimating the field by a search over a reduced set of candidate motion vectors. The motion field is initially predicted from the previously estimated motion fields using linear prediction, simply extrapolating the motion based on two motion vectors taken from two previous fields. (The offset in relative pixel positions between fields of different parity is taken into account in the extrapolation. After this the motion vectors implicitly takes care of the parity issue.) Having the predicted motion field (truncated to integer precision), we collect a list over the J most common motion vectors appearing in the predicted motion field. Thereafter, the search is restricted to the small set of this list for the integer part $(m,\,n)$ of the motion vector in (5). All $H \cdot V$ fractional values of a motion vector are combined with the J integer vectors on the list. Consequently, the final motion vector search consists of trying out $J \cdot H \cdot V$ vectors. This way, we hope to track the motion vectors at picture level without requiring the tracking locally. Thus, even with a small initial search area, between the two fields of a frame, the magnitude of the motion vectors on the list may increase considerably with no explicit limit to the magnitude. Very fast motion, exceeding the initial search area between two fields of the same frame, is not captured though. In the experiments, we use a fixed-size ($J = 35$) candidate list. The size of the list can be adjusted according to different criterias. As an example, including all motion vectors on the list with an occurrence count greater than some threshold T in the predicted motion field reduces the risk of overlooking the motion vector of an object composed of more than T pixels, as a motion vector is estimated for each pixel. An additional increase in speed for higher-resolution motion fields $(H > 2 \vee V > 2)$ is obtained by letting them be simple subpixel refinements of the motion field found for $H = V = 2$. The processing time for creating the high-resolution motion field is proportional to $J \cdot 4 + H \cdot V$ instead of $J \cdot H \cdot V$, i.e., approximately a reduction by a factor of four for the usual resolution $(H = V = 4)$. As the size of the list with the updated vectors is fixed, the complexity is also proportional to the number of pictures specified by N_f.

[3] The block-based motion-estimation method applied does not warrant higher precision of the motion field.

In order to keep the algorithmic complexity down, we base the decisions in the enhancement algorithm on analysis of the luminance component only, always performing the same operations on a chrominance pixel as the corresponding luminance pixel. Additionally, no special action is taken at the picture boundaries apart from zero padding. The original motion vectors coming with the bit stream were disregarded as a higher resolution is desired. They could be used though, e.g., by including them on the list of predicted motion vectors.

IV. Decimation

The upsampling procedure only performed quality-based filtering for pixels located on the same motion trajectory (within our accuracy). In this section, we state a downsampling scheme applying quality based spatial filtering of the superresolution pictures. The filter coefficient for each pixel should reflect the quality and the spatial distance of the pixel. The quality attributes are dependent on the MPEG quantization (4) and whether the pixel is MC or interpolated. For all possible combinations of quality attributes within the filter window, the optimal filter could be determined given enough training data. Instead, we take the simpler approach of first assigning individual weights to each pixel depending on its attributes relative to the current pixel and then normalizing the filter coefficients.

A two-dimensional linear filter g is applied to the samples of the superresolution picture $\mathbf{p}$ $(=\mathbf{p}(V, H))$ in the vicinity of each sample position (m_0, n_0) in the resulting output image of lower resolution. The filter is a product of a symmetric vertical filter, a symmetric horizontal filter and a function reflecting the quality. The weight of the pixel $p(m, n)$ at (m, n) in $\mathbf{p}$ is

$$\begin{aligned} &g(m, n; m_0, n_0, V, H) \\ &\qquad = K\, g_V(|m - m_0|)\, g_H(|n - n_0|)\, w(m, n). \end{aligned} \tag{13}$$

In this expression, the weight $w(m, n)$ is a function of the quality attributes of the pixel $p(m, n)$ and K is a normalizing factor $(\sum g = 1)$. The 1-D filters g_V and g_H, reflecting the spatial distance, are defined as follows:

$$g_1 = (\ldots, 0, 0, 1, 0, 0 \cdots) \tag{14}$$

$$g_2 = (\ldots, 0, a, 1, a, 0 \cdots) \tag{15}$$

$$g_4 = g_2 * g_2 = (\ldots, a^2, 2a, 1 + 2a^2, 2a, a^2 \cdots). \tag{16}$$

It is noticed that the support of the low-pass filter is $(H + 1) \cdot (V + 1)$ superresolution pixels or approximately the area of one low-resolution pixel. This very small region of support is chosen to reduce the risk of blurring across edges in the decimation process. Furthermore, the value of a should be quite small because very often the individual pictures are undersampled. In the experiments, we use the parameter value $a = 0.1$.

The function $w(m, n)$ (13), reflecting the quality, depends on whether $p(m, n)$ and $p(m_0, n_0)$ are MC superresolution pixels or whether they were found through interpolation. When both pixels are MC [i.e., defined by $\mathbf{p}_{\mathrm{mc}} = \mathbf{p}_{\mathrm{mc}}(V, H)$], their relative quality parameters are used to determine the weight of $p(m, n)$. If one of the pixels is obtained by interpolation, a constant is used for the weight

$$w(m, n) = \begin{cases} \dfrac{w_0}{\gamma}\, \gamma^{(q(m, n)/q(m_0, n_0))^{\delta}} \\ \qquad p(m, n),\ p(m_0, n_0) \in \mathbf{p}_{\mathrm{mc}} \\ 1, \qquad p(m, n) \notin \mathbf{p}_{\mathrm{mc}}, \\ w_0, \qquad p(m, n) \in \mathbf{p}_{\mathrm{mc}},\ p(m_0, n_0) \notin \mathbf{p}_{\mathrm{mc}} \end{cases} \tag{17}$$

where w_0, δ, and γ are parameters.

The parameter $w_0(=w(m_0, n_0))$ specifies the *a priori* worth of a MC $(\mathbf{p}_{\mathrm{mc}})$ pixel compared to an interpolated $(\mathbf{p}_i = \mathbf{p}_i(V, H))$ pixel. The last case in (17), where there is no MC superresolution $(\mathbf{p}_{\mathrm{mc}})$ pixel at the output sample position (m_0, n_0), may occur in conversion to HDTV and in chrominance upsampling. Restoring SDTV there will always be the directly decoded pixel at (m_0, n_0) ensuring a defined pixel in $\mathbf{p}_{\mathrm{mc}}$ at (m_0, n_0).

The parameter γ is a global parameter (set to 0.5) whereas δ is inversely proportional to a local estimate [within a region of size $(H + 1)(V + 1)$] of the variance of the superresolution picture at (m_0, n_0). w_0 is set to 6. The structurally simple downsampling filter specified by (13)–(17) only has the four parameters (a, γ, δ, w_0). The downsampling filter also attenuates noise, e.g., from (small) inaccuracies in the motion compensation. (Larger inaccuracies in the motion compensation are largely avoided by checking the matches and only operating on a reduced list of candidate motion vectors.)

V. Results

Four sequences were encoded: *table, mobcal, tambour-sdtv,* and *tambour-hdtv*. The extremely complex *tambour* sequence has been used both as interlaced SDTV and in HDTV format. For SDTV, the format is 4:2:0 PAL TV, i.e., the luminance frame size is 720 × 576 and the frame rate is 25 frames/sec. For HDTV, the resolution is doubled horizontally and vertically.

The parameters (α, β) of the filter expression (9) are estimated using a small number of frames of the sequence *mobcal*. Calculating the Wiener filter (8), we assume implicitly that the "original" pixels of the superresolution picture taken at the sample positions are equal to the original (low-resolution) pixels of the SDTV test sequence. This yields the curve $h(q_2/q_1)$ depicted in Fig. 4. Fitting the filter parameters of (9) to this curve yields $\alpha = 0.25$ and $\beta = 0.5$. These parameters are used in the processing of all the test sequences. Besides the curve $h_1(q_2/q_1)$ based on average values over all q_1, curves of $h_1(q_2/q_1)$ were recorded for different fixed values of q_1. These curves differ from the average in shape, as well as in level, e.g., expressed by the value for $q_1 = q_2$, i.e., $h_1(1)$. For most of the occurrences, q_2/q_1 was close to 1. The irregular shape of the curves for larger values of q_2/q_1 reflects the sparse statistics and due to this the dependency on the specific data that was used for estimating the Wiener filter. The overall level of h_1 was observed to increase with increasing q_1, reflecting the fact that the motion estimation inaccuracy becomes less important when the quantization error is large.

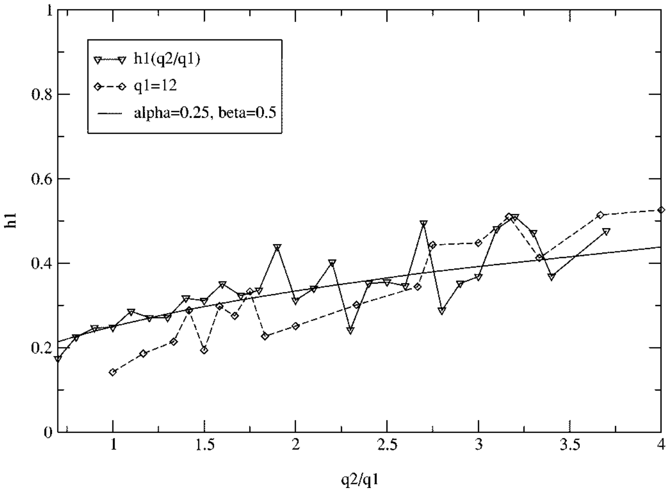

Fig. 4. Wiener filter coefficient h_1 as a function of q_2/q_1 for a piece of *mobcal*. The smooth function is the filter expression obtained by fitting α and β. The curves are $h_1(q_2/q_1)$ for all q_1 and for a small fixed value of $q_1(=12)$.

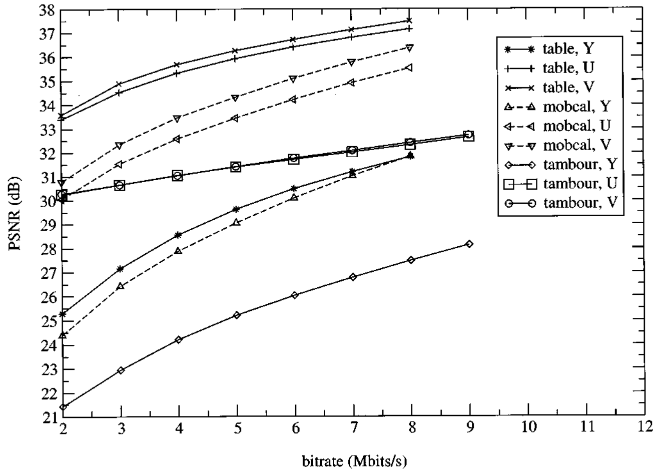

Fig. 5. PSNR of directly decoded sequences as a function of the bit rate.

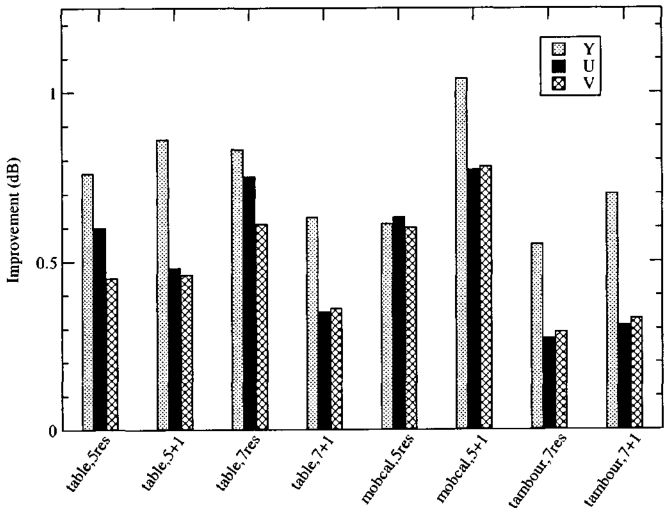

Fig. 6. Average improvement in PSNR for luminance and chrominance for all sequences (res) using parameters $H = V = 4$ and $N_f = 5$. The result of increasing the bit rate by 1 Mb/s (+1) is given for comparison.

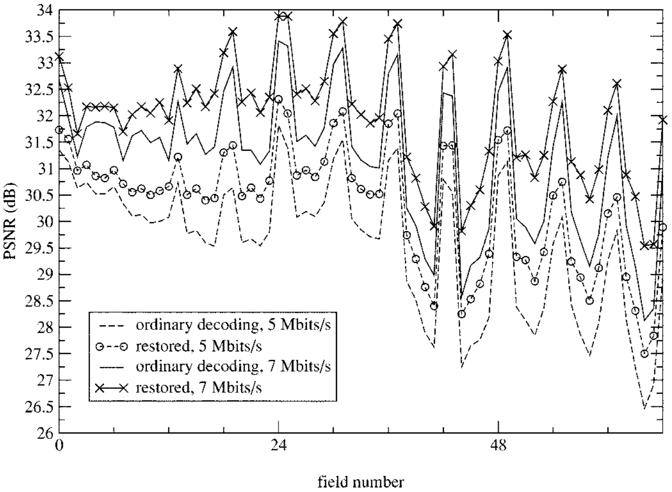

Fig. 7. PSNR measured for sequence *table* (luminance). The GOP consists of 24 pictures.

A. MSE Results

Fig. 5 shows the PSNR of the directly decoded sequences (for the first 33 frames in each sequence, which is the part being used in the tests). The average PSNR improvement for the sequences using our algorithm is depicted in Fig. 6. For comparison, the improvement obtained by increasing the coded bit rate by 1 Mb/s is also shown. Over these sequences, the average improvement achieved by our algorithm is roughly the same as the improvement obtained by increasing the bit rate by 1 Mb/s.

Figs. 7–9 show the PSNR for the individual pictures in the sequence. (The group of picture (GOP) structure consists of 12 frames and thereby 24 pictures: I/P, B/B, B/B, P/P, B/B, B/B, P/P $\cdots$). It is remarkable that the directly decoded sequences display such different characteristics: for *table* and *tambour*, the P pictures have much better PSNR than B pictures, while for *mobcal* this is not so. The restoration algorithm improves all pictures, regardless of their directly decoded quality. The magnitude of the improvement depends on two factors: 1) the relative quality of the directly decoded picture compared to the surrounding pictures and 2) to which degree the temporal redundancy was exploited during MPEG-2 coding. Consequently, the largest improvement (up to 1.7 dB) is recorded for the I pictures of *mobcal*. The P-pictures of *mobcal* being relatively poor and only unidirectionally predicted also display high improvement (about 1 dB). The B pictures of *table* being much worse than the corresponding P pictures display the highest improvement (about 1 dB) for this sequence. Whereas the algorithm generally improves poor pictures the most, some areas may be so poor (e.g., due to occlusions), that the algorithm fails to improve them. This is a consequence of the conservative strategy of requiring a good block match in the reference picture in order for it to influence the current picture. This is also the reason why *tambour* displays a relatively modest improvement and why *table* at 7 Mb/s has a larger improvement than *table* at 5 Mb/s.

In Fig. 10, the influence of the upsampling factors H and V, as well as N_f, is depicted. The superresolution picture is constructed using four $N_f + 2$ field pictures, namely the current field and the four $N_f + 1$ reference field pictures. The results are evaluated by the average improvement in PSNR reconstructing

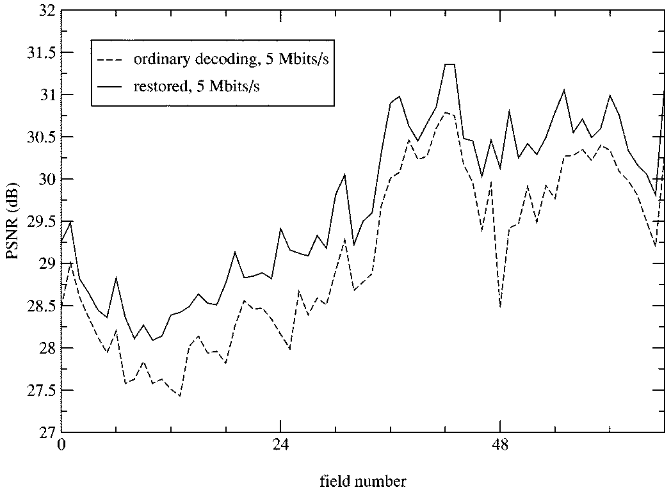

Fig. 8. PSNR measured for sequence *mobcal* (luminance). The GOP consists of 24 pictures.

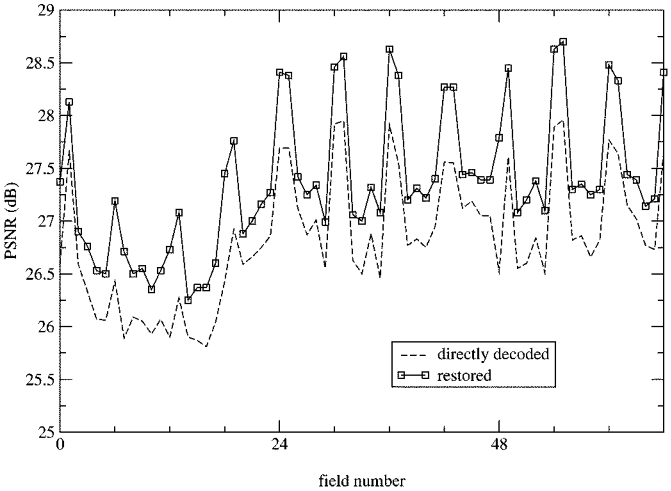

Fig. 9. PSNR measured for sequence *tambour* (luminance). The GOP consists of 24 pictures.

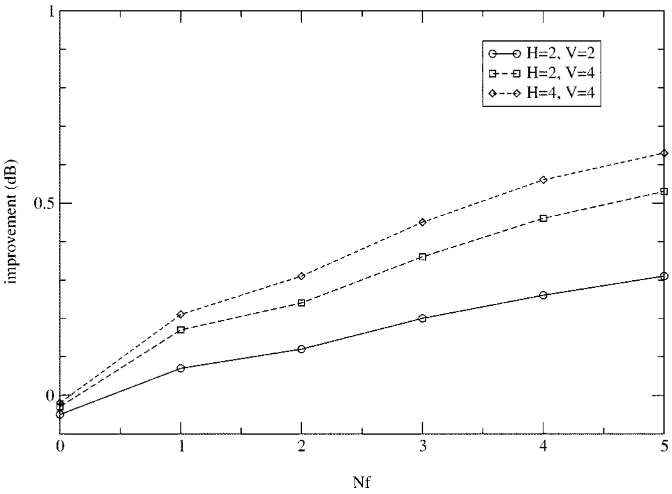

Fig. 10. Average PSNR improvement of the luminance for a GOP of *mobcal* (pictures 24–47) encoded at 5 Mb/s as a function of N_f and the upsampling factors (V, H).

SDTV measured over one GOP of *mobcal*. It is noted that, in this test, the improvement increases with the upsampling factor, implying that the accuracy of the motion field is very important at these bit rates. It is also noticed that the improvement increases with the number of reference pictures. For $N_f = 5$, 11 full frames are used, i.e., almost half a second of video in the restoration of each picture. In this relatively large span of time, the scene is geometrically warped to some extent. The fact that far-away pictures can contribute to the improvement implies that our mechanism for excluding bad matches (see Section III) works satisfactorily.

The algorithm is almost progressive in N_f, as it starts with the nearby reference pictures and works its way to the far-away pictures. We can get the benefit of the restoration little by little, actually traversing along the curves in Fig. 10. This might be useful for freeze-frame applications. The only increase in algorithmic complexity is that we have to perform the decimation multiple times, which only accounts for a minor part of the processing time.

For *tambour*, we can measure the performance of restoration to HDTV. The PSNR is only measured for the even fields. (This is in order to exclude the effect of a resampling in the measurements.) For *tambour* coded at 7 Mb/s, the restoration method gave a 0.76-dB PSNR improvement for the luminance in comparison to simple spatial upsampling of the directly decoded pictures. For the latter method, the upsampling filter of (12) was used for calculating odd samples of the even field.

B. Panel Tests

The sequences were presented for a panel of eight (PAL TV) expert viewers. Each viewer was seated at a fixed distance between two and six screen heights. The sequences were displayed on a 50-Hz interlaced high-fidelity TV using split screen in 20 tests in all. The viewers made blind pair-wise comparisons of the directly decoded, the restored, and the original sequence. In each pair-wise comparison, they scored $(-1, 0, +1)$ indicating the best $(+1)$ and the worst (-1) of the two or equal quality (0). They were also asked to judge sharpness, artifacts, etc.

The reconstructed sequences were overall rated as equally good or better than the corresponding directly decoded sequence (with an average overall score of 0.5 on the -1 to $+1$ scale). The overall evaluation was highly correlated with the degree the artifacts were evaluated to be reduced in the restored sequences. The sharpness was also evaluated to be improved by the restoration but less noticeable.

In a comparison between a directly decoded (*table*) sequence coded at 7 Mb/s and a restored sequence coded at 5 Mb/s, the panel judged the sequences to be of equal quality overall. Some viewers observed that the 7 Mb/s sequence was sharper.

Using our method for upsampling a decoded sequence to HDTV produced acceptable results for *mobcal* and *table*. The restored HDTV sequence of the very complex *tambour* was too bleak and lacked details though. For all sequences, our results were visually significantly better than simple spatial upsampling.

Deinterlacing was tested in a frame-freeze setting viewing single images of a progressive sequence. The images obtained by our enhancement algorithm were also evaluated as being of acceptable quality. Figs. 11–13 show part of an image of mobcal resulting from deinterlacing to progressive format. Fig. 11 depicts the result of using simple upsampling of a directly decoded

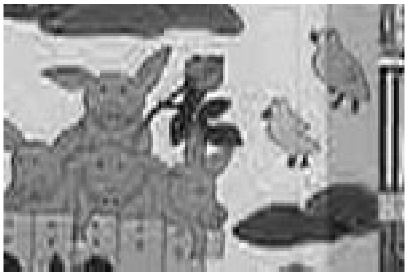

Fig. 11. Direct progressive SDTV. Mobcal at 5 Mbs/s. Part of the I-picture (frame 24, top field). PSNR $= 28.5$ dB.

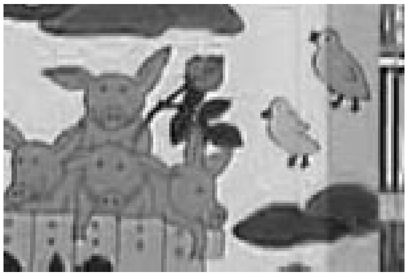

Fig. 12. Enhanced to progressive SDTV. Mobcal at 5 Mb/s. Part of the I-picture (frame 24, top field). PSNR $= 30.1$ dB.

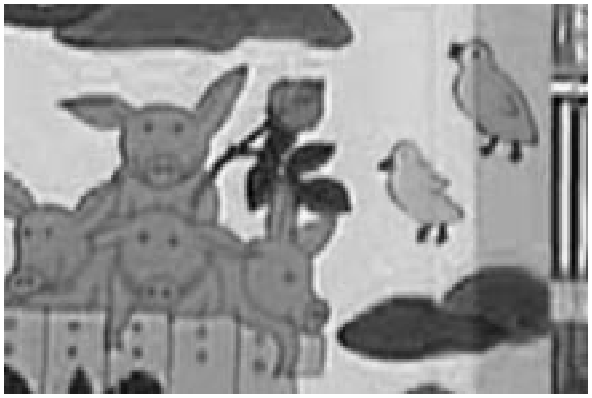

Fig. 13. Enhanced to progressive HDTV in 4 : 2 : 0. Mobcal at 5 Mb/s. Part of the I picture (frame 24, top field).

sequence. Figs. 12 and 13 depict the results of our enhancement to progressive SDTV and HDTV images, respectively.

VI. Conclusion

We have achieved a significant improvement of the decoding quality of MPEG-2 encoded sequences coded at bit rates that are usually considered to provide good quality for distribution. The algorithm is based on MC spatial upsampling from multiple pictures and decimation to the desired format. The processing involves an estimated quality of individual pixels. The quality is estimated from MPEG-2 code streams in our work. Improved MPEG-2 decoding and MPEG-2 SDTV to HDTV conversion were demonstrated. The quality is improved both for moving pictures and for the individual still pictures. Measured by MSE, the improvement roughly corresponds to the improvement obtained by incrementing the bit rate by 1 Mb/s. Subjective tests suggest that the performance of the algorithm is even better than this because it efficiently suppresses mosquito noise, the main artifact at the bit rates used in these tests. The algorithm is conceptually simple but the computational demand is high as it is based on high-accuracy estimation of a dense motion field. An initial application could be progressive improvement of frame freeze for displaying and printing single images based on deinterlacing and possibly upsampling. For these applications, the technique could be combined with POCS.

References

[1] ISO/IEC 13 818-2, "Information technology—Generic coding of moving pictures and associated audio information—Part 2: Video," International Standard (MPEG-2), 1996.

[2] MPEG Group. (1996) MSSG MPEG-2 video software encoder, TM5. [Online]. Available: URL http://www.mpeg.org/MPEG/MSSG

[3] H. Stark and Y. Yang, *Vector Space Projections*. New York: Wiley, 1998.

[4] Y. Yang, M. Choi, and N. Galatsanos, "New results on multichannel regularized recovery of compressed video," in *Proc. ICIP '98*, vol. 1, Oct. 1998, pp. 391–395.

[5] 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.

[6] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Lett.*, vol. 5, pp. 33–35, Feb. 1998.

[7] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Processing*, vol. 6, pp. 1646–1658, Dec. 1997.

[8] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Trans. Image Processing*, vol. 6, pp. 1064–1076, Aug. 1997.

[9] A. J. Patti and Y. Altunbasak, "Super-resolution image estimation for transform coded video with application to MPEG," in *Proc. ICIP*, 1999, pp. 179–183.

[10] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[11] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, Singapore: World Scientific, 1989.

[12] JPEG-LS, "IS 14 495-1, lossless and near-lossless coding of continuous tone still images (JPEG-LS)," ISO/IEC International Standard, 1998.

[13] B. Martins and S. Forchhammer, "Lossless compression of video using motion compensation," in *Proc. 7th Danish Conf. Pattern Recognition and Image Analysis*, 1998, pp. 59–67.