Region-based super-resolution for compression

Article *in* Multidimensional Systems and Signal Processing · September 2007

DOI: 10.1007/s11045-007-0019-y

Rodrick Molina
Universidad Internacional SEK
71 PUBLICATIONS **920** CITATIONS

Aggelos Katsaggelos
Northwestern University
784 PUBLICATIONS **18,028** CITATIONS

Gustavo Marrero Callico
Universidad de Las Palmas de Gran Canaria
148 PUBLICATIONS **1,001** CITATIONS

Multidim Syst Sign Process
DOI 10.1007/s11045-007-0019-y

Region-based super-resolution for compression

**D. Barreto · L. D. Alvarez · R. Molina ·
A. K. Katsaggelos · G. M. Callicó**

Received: 15 January 2006 / Revised: 3 May 2006 /
Accepted: 1 June 2006

Abstract Every user of multimedia technology expects good image and video visual quality independently of the particular characteristics of the receiver or the communication networks employed. Unfortunately, due to factors like processing power limitations and channel capabilities, images or video sequences are often downsampled and/or transmitted or stored at low bitrates, resulting in a degradation of their final visual quality. In this paper, we propose a region-based framework for intentionally introducing downsampling of the high resolution (HR) image sequences before compression and then utilizing super resolution (SR) techniques for generating an HR video sequence at the decoder. Segmentation is performed at the encoder on groups of images to classify their blocks into three different types according to their motion and texture. The obtained segmentation is used to define the downsampling process at the encoder and it is encoded and provided to the decoder as side information in order to guide the SR process. All the components of the proposed framework are analyzed in detail. A particular implementation of it is described and tested experimentally. The experimental results validate the usefulness of the proposed method.

D. Barreto (✉) · G. M. Callicó
Research Institute for Applied Microelectronics, IUMA, University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain
e-mail: dbarreto@iuma.ulpgc.es

L. D. Alvarez · R. Molina
Departamento de Ciencias de la Computación e I.A, Universidad de Granada,
18071 Granada, Spain
e-mail: rms@decsai.ugr.es

A. K. Katsaggelos
Department of Electrical Engineering and Computer Science, Northwestern University,
2145 Sheridan Road, Evanston, IL 60208-3118, USA
e-mail: aggk@ece.northwestern.edu

1 Introduction

Video compression has been the enabling technology behind the multimedia revolution we are experiencing. There have been a number of successful video compression standards (i.e., MPEG-2, MPEG-4, H.264), while there are on-going standardization efforts (i.e., SVC and the recently announced H.265). All existing standards define the decoder, i.e., the syntax of the bitstream which will make it decodable by the decoder. Pre- and post-processing techniques are therefore not considered part of the standard and represent the add-on value provided by the implementer of the standard. More importantly they are considered as separate steps, for example, the pre-processor typically does not interact with the rate controller (an exception is presented in Karunaratne, Segall, & Katsaggelos, 2001). Furthermore, as is justifiable in most cases, an open-loop system is considered, i.e., the pre-processor does not have knowledge of the post-processor, and vice versa. When a codec providing spatial scalability is considered (i.e., MPEG-4, H.264), then the upsampling process represents the normative part.

The term *super-resolution* is typically used in the literature to describe the process of obtaining a high resolution (HR) image or a sequence of HR images from a set of low resolution (LR) observations. This term has been applied primarily to spatial and temporal resolution enhancement (a comprehensive classification of spatio-temporal super-resolution (SR) problems is provided by Borman, 2004) and more recently to spectral together with spatial resolution enhancement (see Molina, Mateos, & Katsaggelos, 2006 for a short review). In these two contexts SR is used to ameliorate the undesirable reduction in resolution introduced by the imaging system. However, the intentional downsampling during pre-processing and the application of SR techniques during post-processing can be utilized as a mechanism in controlling the bitrate so that the overall quality of the reconstructed video is optimized.

The following scenario is suited for SR for compression. As Segall, Elad, Milanfar, Webb, and Fogg (2004) describe, although the bitrates of current high definition systems ensure fidelity in representing an original video sequence, they preclude widespread availability of high definition programming. For example, satellite and Internet based distribution systems are poorly suited to deliver a number of high rate channels, and video on demand applications must absorb a significant increase in storage costs. Furthermore, pre-recorded DVD-9 stores less than an hour of high definition video. Additionally, with the current proliferation of HDTV sets, there is a growing number of DVD players and DVD recorders that can now upscale standard DVD playback output to match the pixel count of HDTV in order to display the signal on a HDTV set with higher definition. In this context, SR can be applied at the output of the DVD player to increase not only the resolution of the video sequence but, also, to increase its quality.

Within the same high bitrate scenario, DVD recorders cannot record in HDTV standards. This is due to the spatial resolution used in DVD format and the limited space of current DVDs for the storage needs of HDTV signals. An alternative for storing HDTV programs with DVD recorders is to pre-process and downsample the HDTV signal to a resolution supported by the standard DVD recorder prior to the compression and storing process. Then, when playing back the compressed video, a SR process can be applied to recover the signal at the original HDTV resolution. Note that the pre-processing and dowsampling operations can vary spatially and temporally depending on the characteristics of the frame being processed. In this case it will be highly desirable that both the coder and the SR algorithm share the information on how the low resolution DVD video sequence has been obtained from the HDTV signal. Together with these examples, spatial scalability in video represents an area where downsampling, compression, and upsampling play a key role in designing systems that split a single video source (or video object plane) into a base layer (lower spatial resolution) and enhancement layers (higher spatial resolution).

In Molina, Katsaggelos, Alvarez, and Mateos (2006), we analyzed the effect of pre-processing and downsampling a video sequence prior to using the MPEG-4 video compression standard and the posterior use of SR techniques to spatially upsample the compressed sequence. We compared the use of different filtering methods, and also compared the quality of the compressed sequence using a “standard” approach and using pre-processing, downsampling, compression, and SR techniques during decoding, at the same bitrate. As we mentioned in Molina, Katsaggelos, Alvarez, and Mateos (2006), we believe that by addressing a number of questions related, for example, to rate control, and the optimal design of the downsampling and upsampling factors and filters, a new paradigm can emerge in developing new video compression schemes.

In this paper, we explore the combined use of SR techniques and side information on the HR original sequence which is downsampled before compression and transmission to generate a HR video sequence from its downsampled LR decoded version. The side information will consist of a segmentation carried out on an image or a group of HR images. The downsampling process will be performed on each image depending on the obtained segmentation. At the decoder, each downsampled compressed group of images will be upsampled using SR techniques and the additional information (segmentation) included in the bit-stream.

The paper is organized as follows. In Sect. 2, we mathematically formulate two different models for video compression. The first one uses hybrid motion compensation and transform based coding, while the second one segments and downsamples the original HR sequence before using hybrid motion compensation and transform based coding to later upsample and postprocess the compressed video sequence using the HR segmentation and SR techniques. In Sect. 3, we provide a brief high level description of hybrid motion compensation and transform based coding. For the proposed second model, we describe in Sect. 4 how the segmentation is carried out, the use of the segmentation to define different subsampling patterns and finally the combination at the decoder of the LR compressed sequence, the segmentation and SR techniques to obtain an estimate of the original HR sequence. Section 5 shows a particular implementation of the proposed method on currently available compression standards and the benefit of its use to obtain HR video sequences from LR compressed observations. Finally, Sect. 6 concludes the paper.

2 Problem formulation

Let us denote by $f(x, y, t)$ the continuous in time and space dynamic scene which is being imaged. If sampling according to the Nyquist criterion in time and space, such scene is represented by the sequence $f_l(m, n)$, where $l = 1, \ldots, L$, $m = 1, \ldots, M$, and $n = 1, \ldots, N$ represent, respectively, the discrete temporal and two spatial coordinates. We use $\mathbf{f}_l$ to denote the vector corresponding to the lth image frame which is lexicographically ordered by rows. In order to represent images as vectors, all images will be lexicographically ordered by rows.

To compress the original sequence $\mathbf{f} = \{\mathbf{f}_1, \ldots, \mathbf{f}_L\}$ we can use hybrid motion compensated and transform based coding at a given bitrate R^I, to obtain $\mathbf{f}^I = \{\mathbf{f}_1^I, \ldots, \mathbf{f}_L^I\}$, an estimate of the original sequence $\mathbf{f}$, as

$$\mathbf{f}^I = C_{R^I}[\mathbf{f}], \tag{1}$$

where C_{R^I} denotes a compression–decompression method at bitrate R^I. This model is henceforth referred to as Model I.

We can control (reduce) the bitrate needed to compress the sequence $\mathbf{f}$ by downsampling in addition to pre-processing of the original sequence before compression and by post-processing with upsampling the reconstructed downsampled sequence. This model will be referred to as Model II.

Our pre-processing for Model II includes a segmentation of the images in the original HR sequence. The segmentation classes will be defined later. The video compression method to be used will have to allow for the transmission of this segmentation as side information (see, for instance, MPEG-2 (Information technology, 2004), MPEG-4 (Information technology, 2005) and H.264 (Information technology, 2005) standards). The side information (segmentation) denoted by $\boldsymbol{\zeta}$ is the result of the application of the function SI on the HR sequence, that is,

$$\boldsymbol{\zeta} = \mathrm{SI}(\mathbf{f}). \tag{2}$$

Once the segmentation (side information) has been obtained, each original frame $\mathbf{f}_l$ in the sequence $\mathbf{f}$ is pre-processed and downsampled by a factor δ using the $(M/\delta \times N/\delta) \times (M \times N)$ matrix $H_{D\,l}^{\zeta}$ to produce the LR frame

$$\mathbf{g}_l = H_{D\,l}^{\zeta}\mathbf{f}_l, \quad l = 1, \ldots, L. \tag{3}$$

Note that the downsampling of each image depends on the segmentation of the original sequence. Matrix $H_{D\,l}^{\zeta}$ could also include temporal processing. In this case, $H_{D\,l}^{\zeta}$ is a function whose argument is the sequence $\mathbf{f}$.

The LR sequence $\mathbf{g} = \{\mathbf{g}_1, \ldots, \mathbf{g}_L\}$ is lossy compressed producing the LR compressed sequence $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$ while the side information $\boldsymbol{\zeta}$ is uncompressed. $\mathbf{g}$ and $\boldsymbol{\zeta}$ are then transmitted by the encoder at a bitrate R^{II}, that is,

$$\mathbf{y}, \boldsymbol{\zeta} = C_{R^{II}}[\mathbf{g}, \boldsymbol{\zeta}]. \tag{4}$$

Finally, the reconstructed LR sequence $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$ is post-processed using U^{ζ} to obtain the second estimate of the original HR sequence $\mathbf{f}^{II}$, that is,

$$\mathbf{f}^{II} = U^{\zeta}[\mathbf{y}]. \tag{5}$$

Note that this post-processing depends on the side information $\boldsymbol{\zeta}$, as well as the specific SR technique, and will produce the HR sequence $\mathbf{f}^{II} = \{\mathbf{f}_1^{II}, \ldots, \mathbf{f}_L^{II}\}$.

3 Model I: Hybrid motion compensated and transform based coding

We provide a brief and high level description of a hybrid motion-compensated video compression system. The sequence $\mathbf{f}$ is compressed with a video compression system resulting in $\mathbf{f}^I = \{\mathbf{f}_1^I, \ldots, \mathbf{f}_L^I\}$, according to Eq. 1.

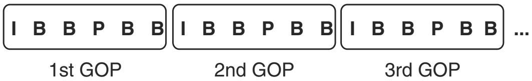


Fig. 1 Example of a particular Group of Pictures (GOP)

During compression, frames are divided into groups of pictures (GOP) and each image in a GOP is encoded with one of two available methods, intracoding or intercoding. Three types of images are usually present in each GOP: intra-coded frames (I-frames), predictive-coded frames (P-frames) and bidirectionally predictive-coded frames (B-frames). Intra-codes frames are coded independently of previous or future frames. Predictive-coded frames are predicted from temporally preceding I or P-frames, and bidirectionally predictive-coded frames are predicted from the previous and the subsequent I or P-frames for bidirectional pictures. For an example of a GOP see Fig. 1, where both the number of B-frames between P-frames and the number of P-frames between I-frames have been previously established for a GOP period equal to six.

For intra-coded images (I-frames), a linear transform such as the DCT (Discrete Cosine Transform) is applied to each block (usually of size 8 × 8 pixels). The operator decorrelates the intensity data and the resulting transform coefficients are independently quantized and transmitted to the decoder. For inter-coded images (P- and B- frames), predictions for the blocks are first generated by motion compensating previously transmitted image frames. The compensation is controlled by motion vectors that define the spatial and temporal offset between the current block and its prediction. Computing the prediction error, transforming it with a linear transform, quantizing the transform coefficients, and transmitting the quantized information refine the prediction.

Using all this information, the relationship between compressed and uncompressed frames becomes

$$\mathbf{f}_l^I = T^{-1}Q\left[T\left(\mathbf{f}_l - \mathrm{MC}_l(\mathbf{f}_l^{IP}, \mathbf{v}_l^I)\right)\right] + \mathrm{MC}_l(\mathbf{f}_l^{IP}, \mathbf{v}_l^I), \quad l = 1, \ldots, L, \tag{6}$$

where $Q[.]$ represents the quantization procedure, $T[.]$ and $T^{-1}[.]$ are the forward and inverse-transform operations, respectively, $\mathrm{MC}_l(\mathbf{f}_l^{IP}, \mathbf{v}_l^I)$ is the motion compensated prediction of $\mathbf{f}_l$ formed by motion compensating previously decoded frame(s) as defined by the encoding method, and $\mathbf{f}_l^{IP}$ and $\mathbf{v}_l^I$ denote the set of decoded frames and motion vectors that predict $\mathbf{f}_l$, respectively. We want to make clear here that MC_l depends only on a subset of $\mathbf{v}_l^I$ and $\mathbf{f}^I$. For example, typically $\mathbf{f}_l^{IP} = \mathbf{f}_{l-1}^I$ and $\mathbf{v}_l^I = \mathbf{v}_{l,l-1}^I$. However, as there is a trend toward increased complexity and non-causal predictions within the motion compensation procedure, a larger subset of previously decoded frames can be used, as is for example the case with H.264.

As already mentioned, during compression images are divided into blocks. If an image of size $M \times N$ is divided into blocks of size $p \times q$ a generic block in the lth image will be denoted by $f_l^B[bm, bn]$, that is,

$$\begin{gathered} f_l^B[bm, bn] = \left(f_l\left((bm-1)*p+i, (bn-1)*q+j\right) \mid i = 1, \ldots, p,\ j = 1, \ldots, q\right), \\ bm = 1, \ldots, \frac{M}{p}, \quad bn = 1, \ldots, \frac{N}{q}, \end{gathered} \tag{7}$$

where the components of the block have been ordered lexicographically by rows. The superscript B denotes that block notation is used.

We will denote by $v^{BI}(bm, bn, l, k)$ the (2×1) vector used to predict the block $f_l^B[bm, bn]$ from some previously coded frame $\mathbf{f}_k^I$. The motion vectors that predict block (bm, bn) in image $\mathbf{f}_l$ from $\mathbf{f}_k^I$ are represented by the $(2 \times \frac{M}{p} \times \frac{N}{q}) \times 1$ vector $\mathbf{v}_{l,k}^{BI}$ that is formed by stacking the transmitted horizontal and vertical offsets.

4 Model II: Region based super-resolution with the use of side information

Model II aims at reducing the bit-rate needed to transmit a video sequence, by performing the following three steps: (a) segmenting and pre-processing with downsampling the images in the original HR sequence, (b) compressing the obtained LR sequence, and (c) post-processing and upsampling the compressed LR sequence.

Let us assume that the blocks used for the compression of the HR images are of size $p \times q$. In this paper, each $\frac{M}{\delta} \times \frac{N}{\delta}$ LR image $\mathbf{g}_l$ will be divided into blocks of size $\frac{p}{\delta} \times \frac{q}{\delta}$, so that we have the same number of blocks in the HR and LR images.

A generic block in $\mathbf{g}_l$ will be denoted by $g_l^B[bm, bn]$, that is,

$$g_l^B[bm, bn] = \left(g_l((bm-1) * \frac{p}{\delta} + i, (bn-1) * \frac{q}{\delta} + j) \mid i = 1, \ldots, \frac{p}{\delta}\ j = 1, \ldots, \frac{q}{\delta}\right), \qquad (8)$$
$$bm = 1, \ldots, \frac{M}{p}, \quad bn = 1, \ldots, \frac{N}{q},$$

where the components of the block are lexicographically ordered by rows, and the superscript B denotes again that block notation being used.

From Sect. 3, we know that the compression system also provides us with the motion vectors $v^{BII}(bm, bn, l, k)$ that predict block $g_l^B[bm, bn]$ from some previously coded frame $\mathbf{y}_k$. These motion vectors are represented by the $(2 \times \frac{M}{p} \times \frac{N}{q}) \times 1$ vector $\mathbf{v}_{l,k}^{BII}$ that is formed by stacking the transmitted horizontal and vertical offsets.

Using Eq. 6 to obtain $\mathbf{y}$ from $\mathbf{g}$ in Eq. 4 we have

$$\mathbf{y}_l = T^{-1}Q\left[T\left(\mathbf{g}_l - \mathrm{MC}_l(\mathbf{y}_l^P, \mathbf{v}_l^{II})\right)\right] + \mathrm{MC}_l(\mathbf{y}_l^P, \mathbf{v}_l^{II}), \quad l = 1, \ldots, L \qquad (9)$$

and from Eq. 3 we have

$$\mathbf{y}_l = T^{-1}Q\left[T\left(H_{D\,l}^{\zeta}\mathbf{f}_l - \mathrm{MC}_l(\mathbf{y}_l^P, \mathbf{v}_l^{II})\right)\right] + \mathrm{MC}_l(\mathbf{y}_l^P, \mathbf{v}_l^{II}), \quad l = 1, \ldots, L. \qquad (10)$$

Equation 5 is then used to obtain $\mathbf{f}^{II}$, which depends on the side information ζ.

We now divide the description of our model into three parts: segmentation, preprocessing and downsampling, and post-processing and upsampling. Note that the segmentation, and pre-processing and downsampling tasks are carried out at the coder while post-processing and upsampling is carried out at the decoder. A pictorial description of Model II is provided by Fig. 2.

4.1 Motion and texture segmentation

The segmentation algorithm will identify regions in an image according to their amount of motion and texture, producing a final map with up to three labels: "motion (M)", "no motion and flat (F)" and "no motion and textured (T)". The segmentation will be added to the compressed bit-stream as user data information. Our motivation

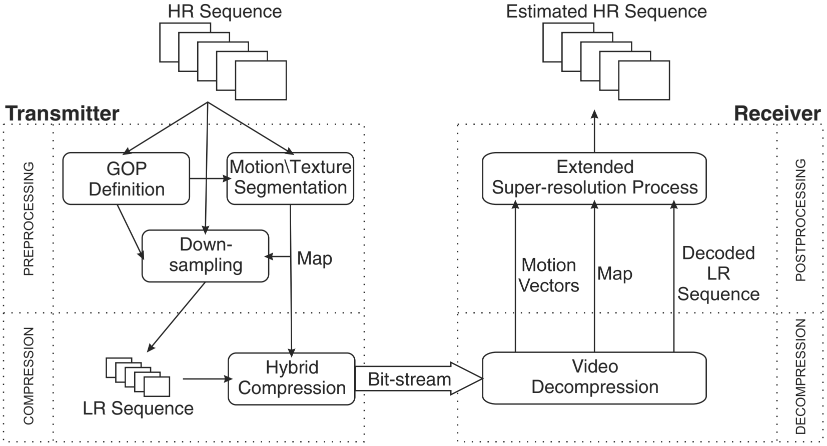


Fig. 2 Pictorial description of Model II. At the transmitter the pre-processing and downsampling stages are performed before compression. The bit-stream is decompressed at the receiver and an extended SR method is applied for post-processing and upsampling the LR sequence

to perform this segmentation is provided below and also explained as a conclusion of Barreto, Alvarez, & Abad (2006).

The motion estimates are usually poorer in flat areas than in areas with texture. When using SR techniques to obtain HR video sequences from LR compressed video sequences, a distinction between these two types of regions can help to improve the estimated HR sequence as well as to reduce the computational load. The estimated HR sequence improves if the texture classification is combined with motion segmentation. HR textured areas with motion can usually be well recovered from the LR images and the motion estimates using SR techniques. Flat areas, particularly when there is no motion, can be recovered using simple interpolation techniques. Finally, areas with no motion and texture are the ones most affected by the downsampling and compression process. Many details will be removed from these areas that are impossible to recover using motion based SR techniques. For such areas we either use motionless super resolution techniques (see, for instance, Chaudhuri & Manjunath, 2005; Freeman, Jones, & Pasztor, 2002) or we modify the downsampling process to attenuate the loss of high frequencies during downsampling (care to avoid the excessive loss of high frequencies for such regions should also be given by an effective rate controller).

Instead of using a segmentation map for each HR image, we will produce a segmentation map for groups of images (GOIs). This will reduce the amount of side information to be transmitted. In the following description, a segmentation mask is produced for each set of images corresponding to the I-frame up to its nearest P-picture in the current GOP and down to and excluding the nearest P-picture in the previous GOP (see Fig. 3). Note that although the description will be carried out for this group of images, the method can also be applied to other partitions of the original HR sequence.

We now proceed with the description of the segmentation procedure. Let us denote by $\boldsymbol{\gamma}$ the set of K HR images defining a given GOI, that is, $\boldsymbol{\gamma}$ is of the form,

$$\boldsymbol{\gamma} = \{\mathbf{f}_{u+1}, \ldots, \mathbf{f}_{u+K}\} \tag{11}$$

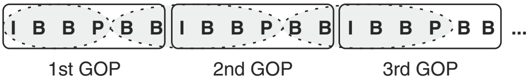


Fig. 3 Each ellipse in shade denotes a Group Of Images (GOI) used for motion and texture segmentation

for a given u. Note that $\boldsymbol{\gamma}$ depends on the set of images used but we will not explicitly write this dependency to simplify the notation. Since Principal Component Analysis (PCA) (Mardia, Kent, & Bibby, 1979) is a powerful tool for analyzing data of high-dimension, we will use for each GOI, the first Principal Component as the basis to build the segmentation map. Note that PCA has already been used in SR problems as a way to summarize prior information on images (see Chaudhuri & Manjunath, 2005; Capel & Zisserman, 2001; Gunturk, Batur, Altunbasak, Hayes, & Mersereau, 2003).

We start by calculating the mean image vector of size $(M \times N) \times 1$ defined by

$$\boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{f}_{u+k}. \tag{12}$$

For each (i,j) HR image position we consider

$$\mathbf{x}_{i,j} = \{f_{u+1}(i,j) - \mu(i,j), \ldots, f_{u+K}(i,j) - \mu(i,j)\} \;\; i = 1, \ldots, N, \quad j = 1, \ldots, M \tag{13}$$

and assume that all $\mathbf{x}_{i,j}$ are realizations of a K-dimensional zero mean random vector $\mathbf{x}$. Then we calculate the sample covariance matrix

$$S = \frac{1}{MN}\sum_{i,j} \mathbf{x}_{i,j}^T \mathbf{x}_{i,j} \tag{14}$$

and obtain its K-dimensional column eigenvectors of size $K \times 1$, $\mathbf{e}_1, \ldots, \mathbf{e}_K$. These eigenvectors are associated, respectively, to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$.

Let

$$\mathbf{e}_1 = (e_1(i), \ldots, e_1(K))^T. \tag{15}$$

Then we use as summary of the whole GOI the image defined as

$$\mathbf{pc} = \sum_{k=1}^{K} e_1(k)(\mathbf{f}_{u+k} - \boldsymbol{\mu}). \tag{16}$$

To perform the three class classification in each GOI we will use the images **pc** and $\boldsymbol{\mu}$. The classification will be done for each block of size $p \times q$ in an image of size $M \times N$ denoted by $B[bm, bn]$ for $bm = 1, \ldots, p$ and $bn = 1, \ldots, q$.

To detect motion we calculate for each block in the **pc** image the quantity

$$M_{\mathrm{SAD}}[bm, bn] = \frac{1}{pq} \sum_{(i,j)\in B[bm,bn]} |pc(i,j) - \overline{pc}^B[bm, bn]|, \tag{17}$$

where $\overline{pc}^B[bm, bn]$ represents the mean value of **pc** in the (bm, bn) block. Following the ideas in Schultz, Meng, and Stevenson (1998) the quantity

$$\tau_M = \mathrm{mean}(M_{\mathrm{SAD}}) - \frac{\mathrm{std}(M_{\mathrm{SAD}})}{2} \tag{18}$$

is used to detect motion, as explained later.

To detect motionless textured and non-textured images we calculate for each block in the $\boldsymbol{\mu}$ image the quantity

$$T_{\mathrm{SAD}}[bm,bn] = \frac{1}{pq} \sum_{(i,j)\in B[bm,bn]} |\mu(i,j) - \overline{\mu}^{B}[bm,bn]|, \tag{19}$$

where $\overline{\mu}^{B}[bm,bn]$ represents the mean value of $\boldsymbol{\mu}$ in the (bm,bn) block. Following, again, the ideas in Schultz, Meng, and Stevenson (1998) the quantity

$$\tau_T = \mathrm{mean}(T_{\mathrm{SAD}}) - \frac{\mathrm{std}(T_{\mathrm{SAD}})}{2} \tag{20}$$

is used to separate textured from non-textured regions, as explained later.

The quantities in Eqs. 17–20 are used for the classification of each block of pixels $B(bm,bn)$ into motion, M, textured, T, and non-textured or flat, F, regions according to

$$\mathcal{C}(B(bm,bn)) = \begin{cases} M & \text{if } M_{\mathrm{SAD}}[bm,bn] > \tau_M, \\ T & \text{if } M_{\mathrm{SAD}}[bm,bn] \leq \tau_M \text{ and } T_{\mathrm{SAD}}[bm,bn] > \tau_T, \\ F & \text{if } M_{SAD}[bm,bn] \leq \tau_M \text{ and } T_{\mathrm{SAD}}[bm,bn] \leq \tau_T. \end{cases} \tag{21}$$

Note that for a given GOI, our segmentation procedure assigns a label to each block of pixels of size $p \times q$ in an image of size $M \times N$. This segmentation obtained utilizing the summary image is propagated to all images in a given GOI.

4.2 Downsampling process

The downsampling process depends on the type of block and the picture prediction type. Without loss of generality in this paper we will assume that $\delta = 2$. As we mentioned earlier, we are considering two types of GOIs, namely BBIBBP and IBBP. We address the first type first, followed by the second one. Clearing, the proposed downsampling approach can be applied to other types of GOIs.

For blocks belonging to M and F classes the downsampling is carried out according to

$$\begin{aligned} \mathbf{g}_l^B[bm,bn] = {} & (\mathbf{f}_l((bm-1)p + 2i - 1, (bn-1)q + 2j - 1) \\ & \mid i = 1, \ldots, \frac{p}{\delta} j = 1, \ldots, \frac{q}{\delta}\Big), \\ & bm = 1, \ldots, \frac{M}{p}, \quad bn = 1, \ldots, \frac{N}{q}, \end{aligned} \tag{22}$$

where $\mathbf{g}_l^B$ is the lth LR block (bm,bn). Note that this downsampling works under the hypothesis that when motion is present, SR techniques can be used to increase the resolution of images, and for flat regions without motion, simple interpolation techniques can be used to increase the size of an image without loss of details.

For blocks in the T class the downsampling pattern depends on the prediction type of the block.

For I-frames, the downsampling coincides with the one applied to blocks in classes M or F, according to Eq. 22.

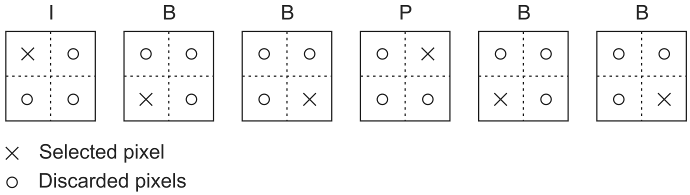


Fig. 4 Example of the downsampling procedure applied to a 2 × 2 block classified as textured (T). Samples labelled as "Selected pixels" are used to create the LR version of every block. The downsampling pattern depends on the prediction image type. For I-frames and P-Frames, the selection of the pixel is performed according to Eqs. 22 and 23, respectively. For B-frames, the downsampling pattern is mathematically described in Eqs. 24 and 25

For blocks in P-pictures belonging to the T class, the downsampling pattern changes to:

$$\mathbf{g}_l^B[bm,bn] = \left(\mathbf{f}_l((bm-1)p+2i-1,(bn-1)q+2j) \mid i=1,\ldots,\frac{p}{\delta}, j=1,\ldots,\frac{q}{\delta}\right). \tag{23}$$

For blocks in B-pictures belonging to the T class, the downsampling pattern is :

$$\mathbf{g}_l^B[bm,bn] = \left(\mathbf{f}_l((bm-1)p+2i,(bn-1)q+2j-1) \mid i=1,\ldots,\frac{p}{\delta}, j=1,\ldots,\frac{q}{\delta}\right) \tag{24}$$

for the first and third B pictures in the GOI and

$$\mathbf{g}_l^B[bm,bn] = \left(\mathbf{f}_l((bm-1)p+2i,(bn-1)q+2j) \mid i=1,\ldots,\frac{p}{\delta}, j=1,\ldots,\frac{q}{\delta}\right) \tag{25}$$

for the second and fourth B pictures in the GOI. Equations 22–25 are graphically shown in Fig. 4.

For the first GOI in Fig. 3, which is of IBBP type, the only difference with the above described downsampling procedure is that we only have two B-frames which are downsampled like the first and second B-frames using Eqs. 24 and 25. Note that in a IBBP sequence we recover, for a T block, the whole HR block. Note also that this sampling introduces artificial motion between blocks.

4.3 Combining HR segmentation, LR compressed images, and SR techniques at the decoder

As already described (see Eq. 5), the image upsampling at the decoder depends on the block class (M, T, or F) on segmentation map ζ and the type of image (I, P, or B) the block belongs to. The upsample operation U^{ζ} over one block of pixels $y_l^B[bm,bn]$ then becomes:

$$U^{\zeta}(y_l^B[bm,bn]) = \begin{cases} \mathrm{BLI}(y_l^B[bm,bn]) & \text{if } C(y_l^B[bm,bn]) = F, \\ \mathrm{SR}(y_l^B[bm,bn]) & \text{if } C(y_l^B[bm,bn]) = M, \\ \mathrm{AMR}(y_l^B[bm,bn]) & \text{if } C(y_l^B[bm,bn]) = T, \end{cases} \tag{26}$$

where BLI stands for BiLinear Interpolation, AMR stands for Artificial Motion Reconstruction, and SR stands for the application of Super Resolution techniques. The SR and AMR procedures are now described.

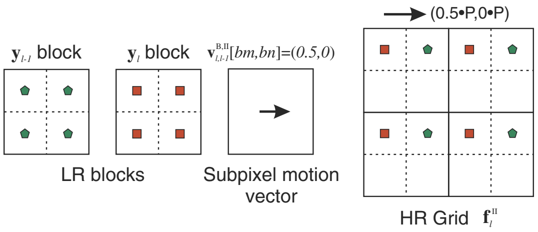


Fig. 5 An example of the applied SR technique to a M-type block of a P-frame. Projection of LR samples onto a HR grid according to the refined transmitted motion vectors

The SR upsampling procedure for the M-labeled blocks is based on non-uniform interpolation techniques. Ur and Gross (1992) used the generalized multichannel sampling theorem of Papoulis (1977) and Brown (1981) to perform non-uniform interpolation. Alam, Bognar, Hardie, and Yasuda (2000) utilized a weighted nearest neighbor interpolation technique. Nguyen and Milanfar (2000) proposed a wavelet interpolation for interlaced two-dimensional data. Callicó, Nú nez, Llopis, and Sethuraman (2003) modified an existing hybrid video encoder platform to add SR by placing new data onto a high resolution grid according to the sub-pixel shifts between frames (for a description of the shift-and-add principle see Farsiu, Robinson, Elad, & Milanfar, 2004).

Our SR procedure adapts the method proposed in Callicó, Nú nez, Llopis, and Sethuraman (2003) to video compression. An HR grid, associated to the lth decompressed image in the sequence $\mathbf{y}_l$, is filled not only with values from $\mathbf{y}_l$ but also from adjacent frames for P- and B-pictures when there is sub-pixel motion. The motion vectors $\mathbf{v}^{B\,II}_{l\,k}$ (see Sect. 4) are provided by the compression standard and are used as initial estimates for a reduced search block matching algorithm, see Erickson & Schultz, 2000 for details and Mateos, Katsaggelos, & Molina, 2000 for an alternative approach. A description of the steps the SR algorithm carries out is the following (an example is shown in Fig. 5):

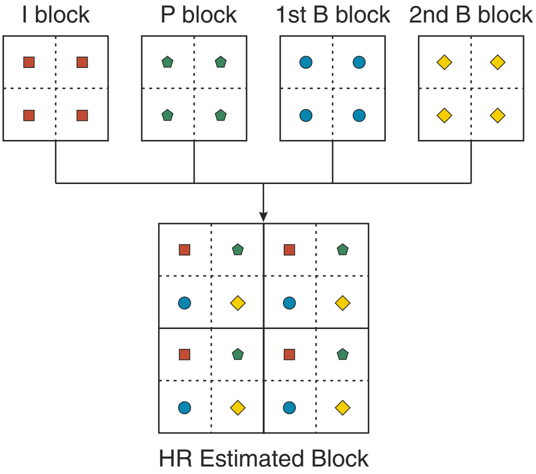
(1) Current LR image pixel values of $\mathbf{y}_l$ are placed in their corresponding positions in the HR image $\mathbf{f}^{II}_l$;
(2) Pixels from adjacent frames (for P and B prediction types) are placed in "empty" positions of the HR grid by utilizing the refined transmitted motion vectors $\mathbf{v}^{B\,II}_{l\,k}$;
(3) After filling all possible positions, "empty" HR pixel cells are interpolated to achieve the final SR image.

For the AMR process the upsampling operation is applied according to the downsampling process which has been performed according to the prediction type and is known to the decoder. The final HR estimated block, for a downsampling factor $\delta = 2$, will be the combination of four upsampled LR blocks, as shown in the example of Fig. 6.

5 Experimental results

The hybrid motion compensated and transform based coder (Model I), described in Sect. 3, and the proposed region based SR with the use of side information (Model II), described in Sect. 4, are now compared on the "Children2", "Deadline" and "Foreman" sequences in CIF format (352 × 288 pixels) at a frame rate of 30 fps stored

Fig. 6 Reconstruction of a block classified as no motion and textured (T) for a block size 2×2 and a downsampling factor of 2



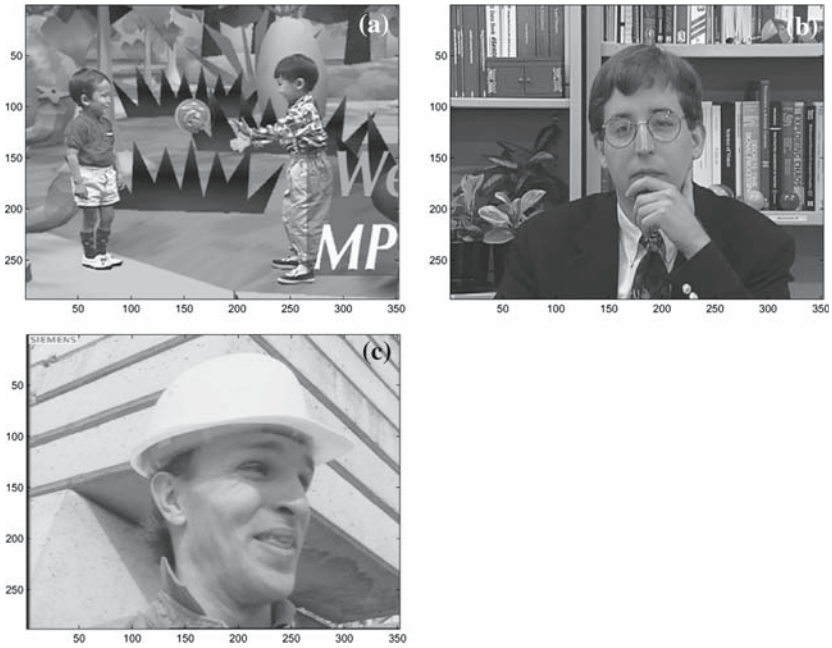


Fig. 7 Original HR image frames: (**a**) frame 12 from the "Children2" sequence, (**b**) frame 3 from the "Deadline" sequence, and (**c**) frame 9 from the "Foreman" sequence

in progressive format. Frame 12 from the "Children2" sequence, frame 3 from the "Deadline" sequence and frame 9 from the "Foreman" sequence are shown in Fig. 7.

MPEG-4 is used as Model I to compress 16 frames of the image sequence (from frames 3 to 18). The GOP size is set equal to 6, with one P-picture between I-frames and two B-pictures between P-frames. The 16 images have therefore the sequential form "IBBPBBIBBPBBIBBP". The MPEG-2 TM5 (Test Model 5) rate control procedure is used in the MPEG-4 configuration file in order to control the bitrate by

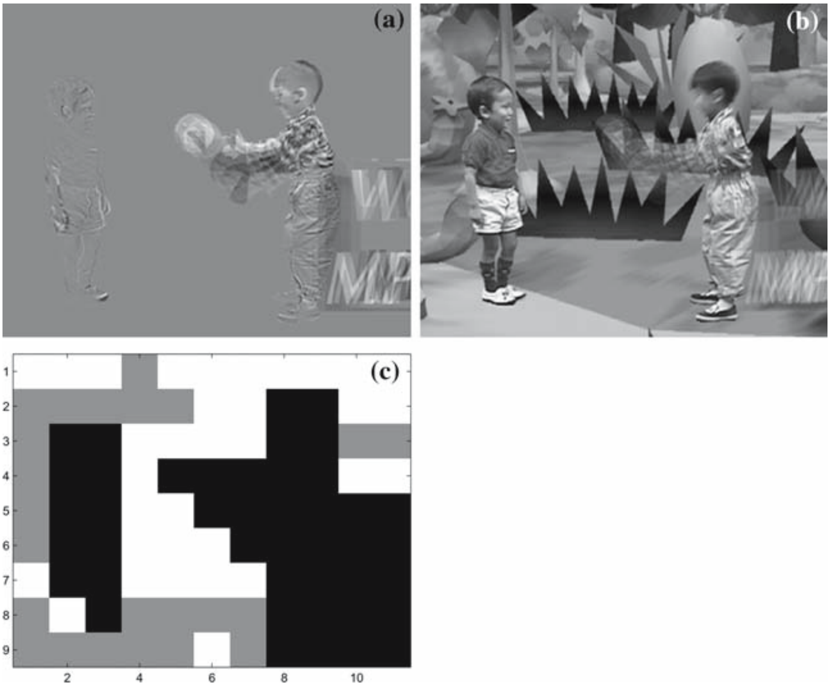


Fig. 8 Segmentation of the second GOI of the “Children2” sequence: (**a**) first principal component for motion analysis, (**b**) mean PCA image used for classification into motionless textured and flat regions, and (**c**) labeling of the macroblocks in the second GOI of the “Children2” sequence. *Black* represents areas with motion (*M*), *gray* motionless flat (*F*) areas and *white* regions classified as no motion and textured (*T*)

adapting the macroblock quantization parameter (frame-based rate control is used as opposed to object-based rate control, also supported by MPEG-4).

For $l = 1, \ldots, 16$, the quality of the HR image reconstructions $\mathbf{f}_l^I$ by Model I (see Eq. 1), and $\mathbf{f}_l^{II}$ by Model II (see Eq. 5), is measured using the Peak Signal to Noise Ratio (PSNR). The mean PSNR is used to asses the quality of the reconstructed video sequences $\mathbf{f}^I$ and $\mathbf{f}^{II}$. Decoded video sequences are compared for bitrates ranging from 32 Kbps to 4 Mbps.

An algorithmic description of Model II, which includes implementation details used in the experiments, is the following:

(1) Each HR macroblock (32 × 32 pixels, corresponding to a 16 × 16 macroblock in LR) of the original sequence is classified into motion (M), texture (T), and flat (F) classes using Eq. 21 to generate the side information file. This side information is produced for every GOI in the HR sequence (see Fig. 3) and consists of a 2-bit label per macroblock.

(2) Each macroblock in every image of the original HR image sequence is downsampled by a factor of 2. The downsampling depends on the MPEG-4 image prediction type and its above classification (see Sect. 4.2).

(3) Model I (MPEG-4) is applied to the downsampled image sequence and the side information (which is not compressed). The side information is included in the

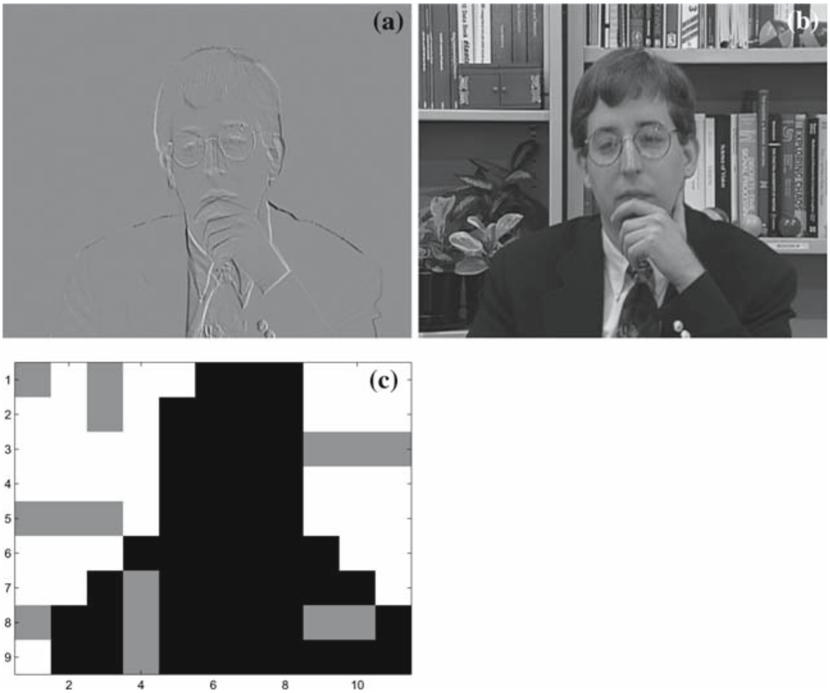


Fig. 9 Segmentation of the first GOI of the "Deadline" sequence: (**a**) first principal component for motion analysis, (**b**) mean PCA image used for classification into motionless textured and flat regions, and (**c**) labeling of the macroblocks in the first GOI of the "Deadline" sequence. *Black* represents areas with motion (M), *gray* motionless flat (F) areas and *white* regions classified as no motion and textured (T)

bitstream as part of the GOV (Group of VideoObjectPlane) header in the User Data field.

(4) The LR compressed sequence is upsampled using the proposed extended SR procedure (see Sect. 4.3).

Model II is computationally more complex than Model I due to the pre-processing and post-processing stages, although the encoding/decoding time is reduced. These experiments were run using Matlab v7.1.0.246 R14 under Windows XP Professional on a Pentium 4 processor at 2.00 GHz with 768 MB of RAM. Computational time was calculated as the mean time for all the sequences and the bitrates considered here, normalized by the number of frames in each sequence. For Model I, the encoding/decoding time was equal to 3.19 s (note that, in this case, the image size is 352×288). For Model II the computational time is divided in three blocks: pre-processing, encoding/decoding, and post-processing. The pre-processing time, including the segmentation and downsampling procedures, was equal to 1.27 s. The encoding/decoding time (note that, in this case, the image size is 176×144), was equal to 0.74 s. Finally, the post-processing time was equal to 15.3 s.

Figures 8, 9, and 10, show an example of the classification procedure based on the PCA of the "Children2", "Deadline" and "Foreman" sequences. The first principal component (see Eq. 16), used for motion detection in Eqs. 17 and 18, is displayed in

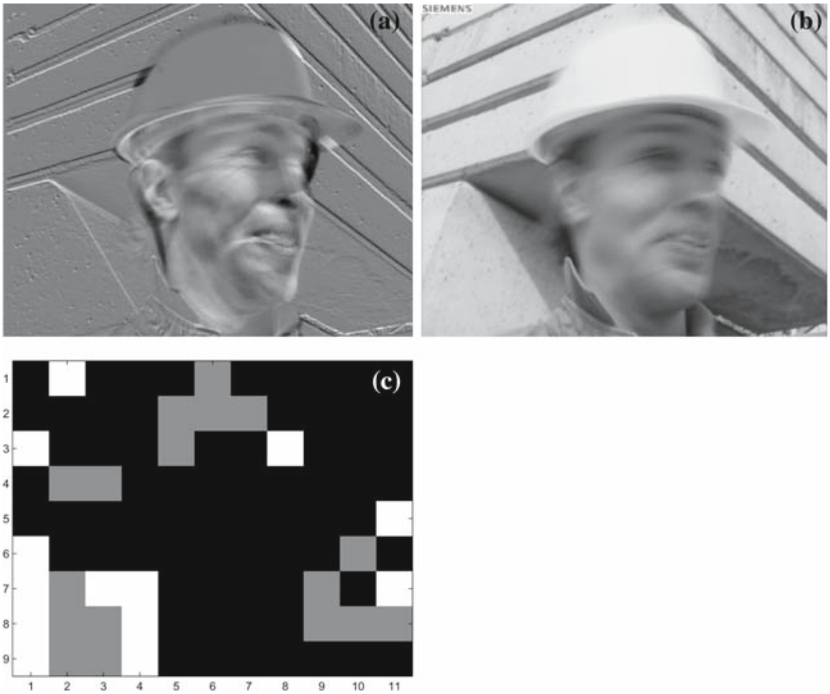


Fig. 10 Segmentation of the second GOI of the "Foreman" sequence: (**a**) first principal component for motion analysis, (**b**) mean PCA image used for classification into motionless textured and flat regions, and (**c**) labeling of the macroblocks in the second GOI of the "Foreman" sequence. *Black* represents areas with motion (M), *gray* motionless flat (F) areas and *white* regions classified as no motion and textured (T)

Figs. 8a, 9a, and 10a. Once a macroblock has been classified as motionless, segmentation into flat and textured regions is performed utilizing the mean PCA (see Eq. 12) image displayed in Figs. 8b, 9b, and 10b using, respectively, Eqs. 19 and 20. The final classification (see Eq. 21) is depicted in Figs. 8c, 9c, and 10c. Every GOI is thus divided into three different regions: "motion" (black), "no motion and flat" (gray) and "no motion and textured" (white).

Examples of the QCIF (176 × 144 pixels) LR images obtained after the downsampling process (which depends on the obtained segmentation) are shown in Fig. 11. The images have been enlarged to CIF size by pixel replication for visualization purposes.

On top of Fig. 12 we show the rate-distortion (RD) curves of the "Children2" image sequence for the 32 Kbps to 4 Mbps bitrate range. The mean PSNR for the HR compressed image sequence using Model I ranges from 27.5 to 40.8 dB while the mean PSNR for the HR image sequence using Model II ranges from 31.3 to 37.2 dB. Model II outperforms Model I by about 5 dB at 512 Kbps. Also note that Model II outperforms Model I at low and medium bitrates.

The RD curve for the "Deadline" image sequence is also depicted in Fig. 12. The mean PSNR for the HR compressed image sequence ranges from 27.8 to 41 dB while using Model II the mean PSNR ranges from 31.8 to 37 dB. Note that Model II outperforms Model I by about 3 dB at 512 Kbps. Again, Model II produces better results than Model I at low and medium bitrates.

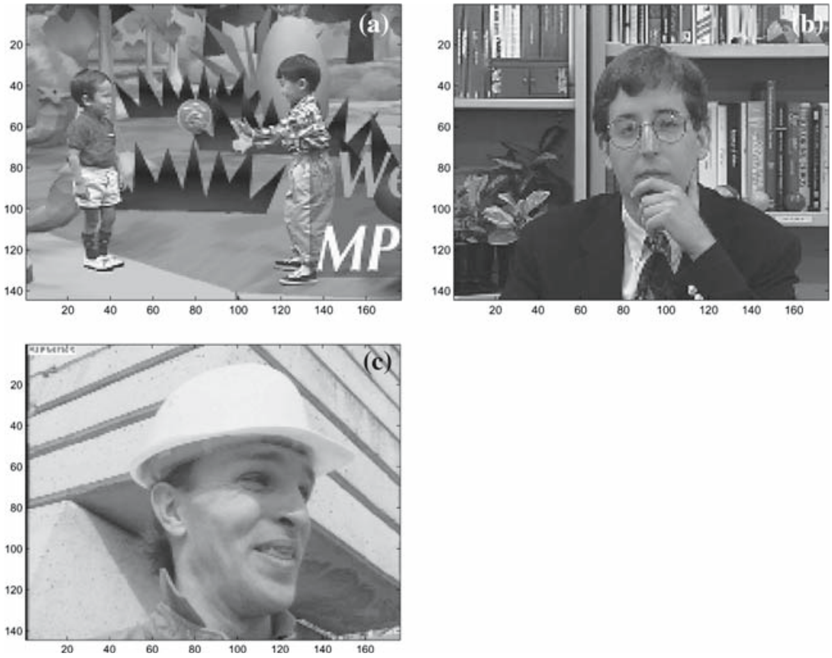


Fig. 11 Examples of LR images after the segmentation and downsampling processes: (**a**) frame 12 of the “Children2” sequence, (**b**) frame 3 of the “Deadline” sequence, and (**c**) frame 9 of the “Foreman” sequence. They have been enlarged by pixel replication for visualization purposes

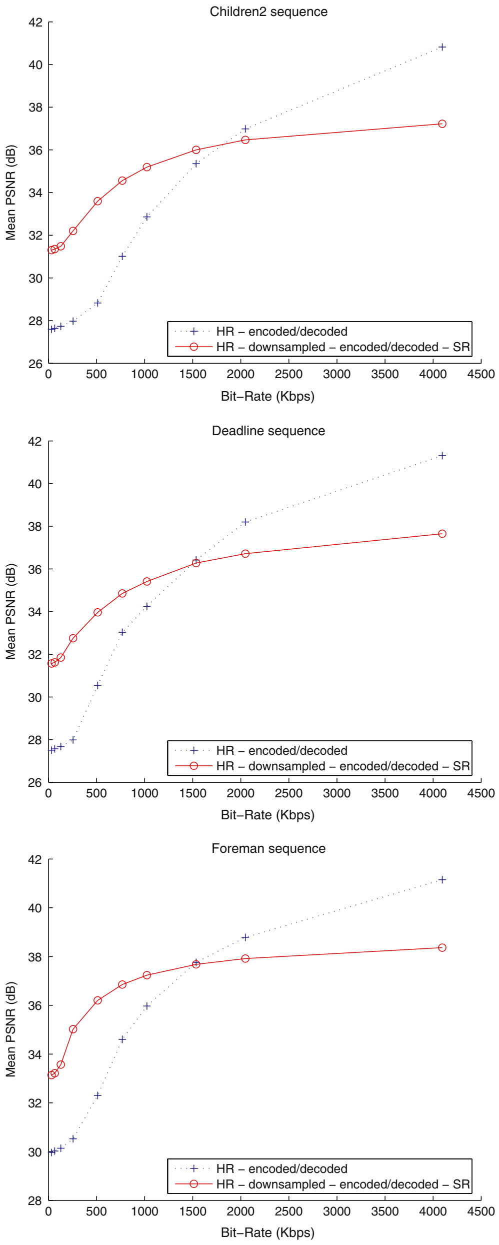
Finally, in the bottom plot of Fig. 12 we show the RD curves of the “Foreman” image sequence for the 32 Kbps to 4 Mbps bitrate range. The mean PSNR for the HR compressed image sequence using Model I ranges from 30 to 41.3 dB, while the mean PSNR for the HR image sequence using Model II ranges from 33.2 to 38.1 dB. Note that Model II outperforms Model I at low and medium bitrates (for example at 512 Kbps) by about 3.5 dB.

We also observe from these figures that for low and medium bitrates in order to achieve specific PSNR values the bitrate needed by Model I is almost double the bitrate needed by Model II. For instance, for a mean PSNR of 34 dB for the “Children2” sequence, Model I requires approximately 1,200 Kbps while the proposed SR approach (Model II) needs only 600 Kbps.

Figure 12 also shows that at high bitrates (above 1.5 Mbps) Model I outperforms Model II. At these bitrates little information is discarded by the coder for both Models I and II. The information discarded, however, during the downsampling process by Model II cannot be recovered by the model. We note here that the side information file size remains constant for every bitrate ($\approx$ 200 bits per GOI). We believe that unless the side information is further compressed (by a lossless scheme) or additional side information is calculated and transmitted, Model II will not outperform Model I at high bitrates.

The PSNR values for every frame for these sequences are depicted in Fig. 13. As can be observed, the Model II performance for each image is relatively less dependent on the frame prediction type, specially at low and medium bitrates. At higher

Fig. 12 Rate distortion characteristics for the considered sequences



bitrates, although the PSNRs obtained by Model I are improved, Model II continues demonstrating a rather frame independent performance.

Reconstructed frame 12 of the "Children2" sequence, frame 3 of the "Deadline" sequence, and frame 9 of the "Foreman" sequence at a bitrate of 512 Kbps by Model I are shown in Fig. 14a, c, and e, respectively. Corresponding reconstructions by Model II are shown in Fig. 14b, d, and f. The shown "Children2" and "Foreman" frames are

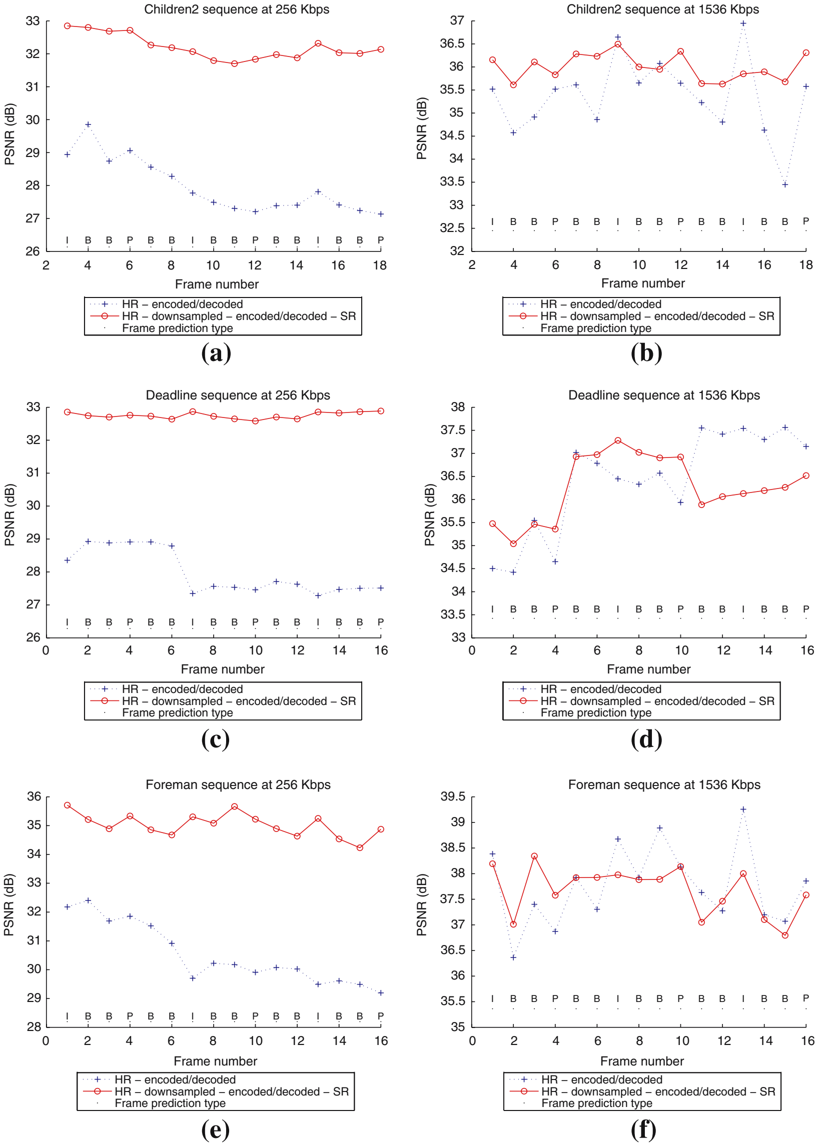


Fig. 13 PSNR values for every frame for the sequences: (**a**) “Children2” at 256 Kbps, (**b**) “Children2” at 1,536 Kbps, (**c**) “Deadline” at 256 Kbps, (**d**) “Deadline” at 1,536 Kbps, (**e**) “Foreman” at 256 Kbps, and (**f**) “Foreman” at 1,536 Kbps. The prediction type of every frame is also depicted

B-pictures and the “Deadline” frame is an I-picture. We observe that the blocking artifacts that appear in the images obtained by Model I are not present in the images obtained by Model II due to the interpolations performed over flat areas. In areas with motion, the quality of the SR reconstruction highly depends on the correctness of the motion vectors provided by the encoder and their posterior refinement. Motion areas are better reconstructed in the bidirectional mode, since it is possible to group information from three different frames (actual frame and two adjacent frames). Note

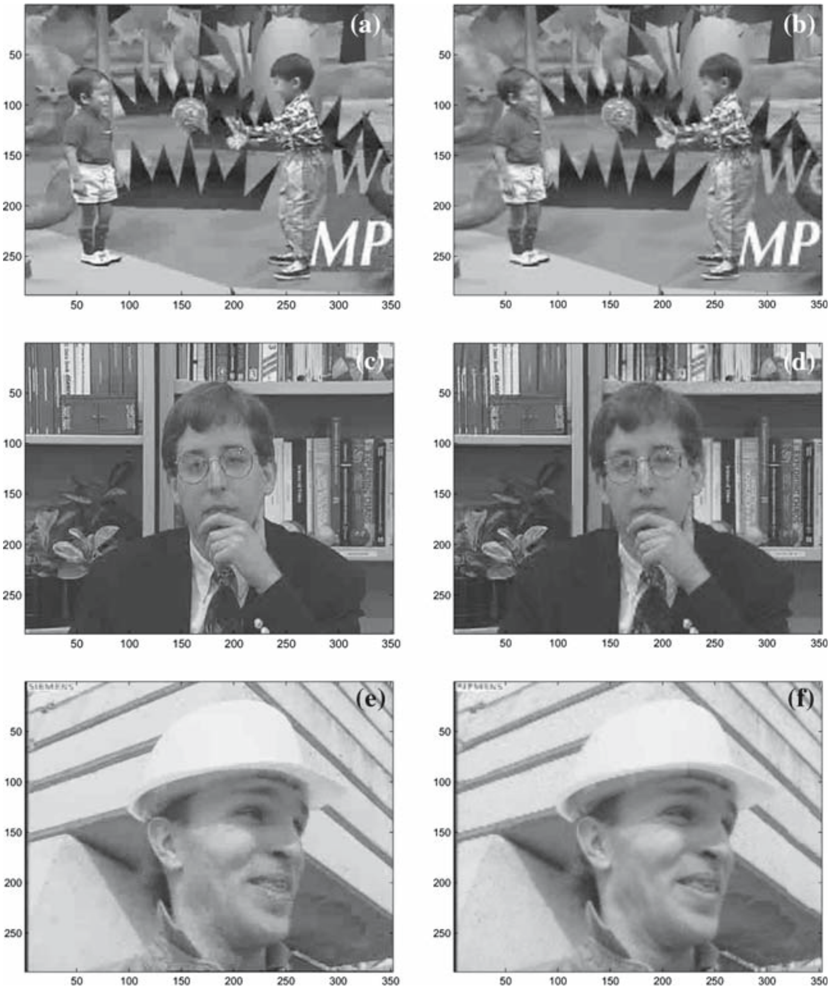


Fig. 14 Frame 12 of the “Children2” sequence, frame 3 of the “Deadline” sequence and frame 9 of the “Foreman” sequence. (**a**), (**c**) and (**e**) are the results of Model I. (**b**), (**d**) and (**f**) are the results of Model II

that with the current implementation of the proposed approach M-blocks do not fair well in an I-frame, since no motion is utilized and therefore spatial interpolation is only applied. This is why the region with motion displayed in Fig. 14d appears to be smoother than its HR compressed version. Clearly this can be improved by using alternative SR techniques. As an example of this type of reconstruction, Fig. 15a and b show details of the man’s face in the “Foreman” sequence for Models I and II, respectively. It can be clearly seen that, in this case, the shown area is better recovered using Model II. Finally, textured areas with no motion can be well recovered. An example is shown in Fig. 15c and d, where the title of one of the books that appears in the picture cannot be figured out in the HR compressed version while it can be read “Exploring Chaos” in the image reconstructed by Model II. Nevertheless, certain artifacts can also appear in these regions due to the errors introduced by the compres-

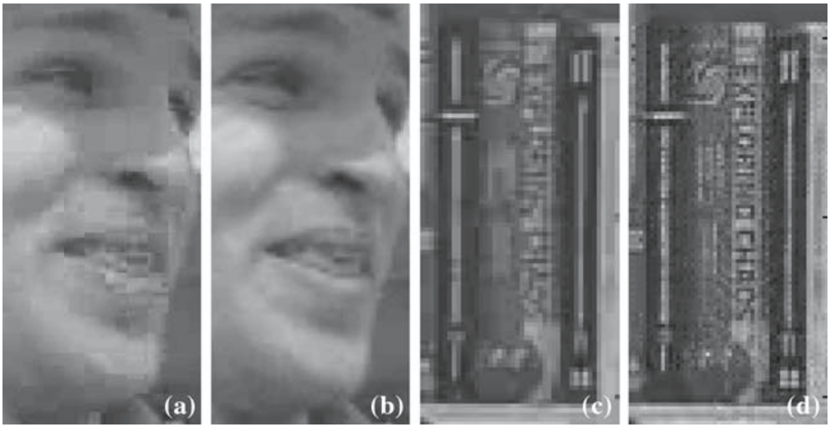


Fig. 15 Detail of a "Foreman" region labeled as motion (M) and a "Deadline" region labeled as no motion and texture (T). (**a**) and (**c**) are images obtained by Model I, (**b**) and (**d**) are images obtained by Model II. The mouth in "Foreman" and the title of the book in "Deadline" have been better reconstructed using Model II

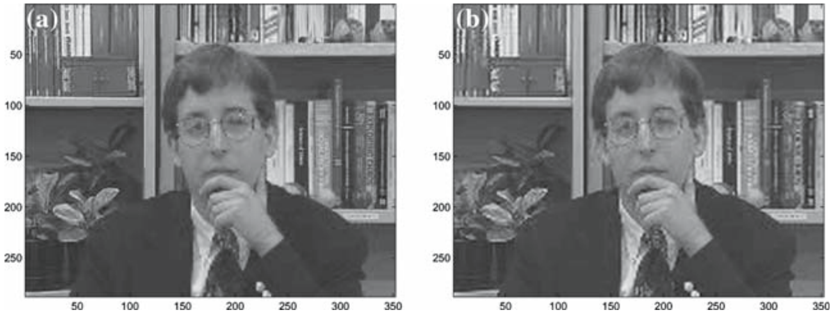


Fig. 16 Frame 3 of the "Deadline" sequence for: (**a**) Model II reconstruction, and (**b**) cubic spline interpolation

sion model. Future work will consider alternative SR motion estimation procedures and quantization noise modelling to further reduce these visually annoying effects.

Finally, Fig. 16 visually compares a frame of the "Deadline" sequence obtained by Model II and the bicubic interpolation of the corresponding downsampled image. As can be seen in this figure, the interpolation produces a blurred image and therefore does not recover many of the details that Model II does.

6 Conclusions

In this paper, we have explored the combined use of super resolution techniques and side information provided by a HR original sequence to generate a HR video sequence from its downsampled LR decoded version. The side information consisted of a segmentation carried out on groups of HR images and the downsampling process has been performed on each image depending on the obtained segmentation. The proposed model has been compared with standard hybrid motion compensated and transform based coding applied to the original HR sequence. Experimental results have validated the utility of the proposed method.

Acknowledgements This work is supported by the Spanish Ministry of Education and Science under the projects TIC2003-09687-C02-02 and TIC2003-00880 and by the Greece-Spain Integrated Action HG2004-0014.

References

Alam, M. S., Bognar, J. G., Hardie, R. C., & Yasuda, B. J. (2000). Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames. *IEEE Transactions Instrumentation Measurement, 49*, 915–923.

Barreto, D., Alvarez, L., & Abad, J. (2006). Motion estimation techniques in super-resolution image reconstruction. A performance evaluation. In: *Virtual observatory. Plate content digitalization, archive mining and image sequence processing*, Sofia, Bulgary, Vol. I, pp.254–268.

Borman, S. (2004). Topics in multiframe superresolution restoration. Ph.D. dissertation, University of Notre Dame, Notre Dame, IN.

Brown, J. (1981). Multi-channel sampling of low pass signals. *IEEE Transactions on Circuits and Systems, 28*, 101–106.

Callicó, G. M., Núnez, A., Llopis, R. P., & Sethuraman, R. (2003). Low-cost and real-time super-resolution over a video encoder ip. In: *Fourth international symposium on quality electronic design (ISQED'03)*, Los Alamitos, CA, USA, pp.79–84.

Capel, D., & Zisserman, A. (2001). Super-resolution from multiple views using learnt image models. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, Kauai, Hawaii USA, Vol. 2, pp.627–634.

Chaudhuri, S., & Manjunath, J. (2005). *Motion-free super-resolution*. Berlin: Springer.

Erickson, K. J., & Schultz, R. R. (2000). Mpeg-1 super-resolution decoding for the analysis of video stills. In: *Proceedings 4th IEEE southwest symposium image analysis*, Austin, TX, pp.13–20.

Farsiu, S., Robinson, D., Elad, M., & Milanfar, P. (2004). Fast and robust multi-frame super-resolution. *IEEE Transactions on Image Processing, 13*, 1327–1344.

Freeman, W. T., Jones, T. R., & Pasztor, E. C. (2002). Example based super-resolution. *IEEE Computer Graphics and Applications, 22*, 56–65.

Gunturk, B., Batur, A., Altunbasak, Y., Hayes, M., & Mersereau, R. (2003). Eigenface-domain super-resolution for face recognition. *IEEE Transactions on Image Processing, 12*, 597–606.

"Information technology—generic coding of moving pictures and associated audio information: Video," ISO/IEC 13818-2:2000, Tech. Rep., 2000.

"Information technology—coding of audio-visual objects—part 2: Visual," ISO/IEC 14496-2:2004, Tech. Rep., 2004.

"Information technology—coding of audio-visual objects—part 10: Advanced video coding," ISO/IEC 14496-10:2005, Tech. Rep., 2005.

Karunaratne, P. V., Segall, C., & Katsaggelos, A. (2001). Rate distortion optimal video pre-processing algorithm. In: *Proceedings of the IEEE international conference on image processing* Thessaloniki, Greece, Vol. 1, pp. 481–484.

Mardia, K., Kent, J., & Bibby, J. (1979). *Multivariate analysis*. New York: Academic Press.

Mateos, J., Katsaggelos, A. K., & Molina, R. (2000). Simultaneous motion estimation and resolution enhancement of compressed low-resolution video. In: *Proceedings IEEE international conference on image processing*, Vancouver, B.C. Canada, Vol. 2, pp.653–656.

Molina, R., Katsaggelos, A., Alvarez, L., & Mateos, J. (2006). Towards a new video compression scheme using super-resolution. In: *Proceedings of the SPIE conference on visual communications and image processing*, San Jose, CA, USA, Vol. 6077, pp.607706/1–607706/13.

Molina, R., Mateos, J., & Katsaggelos, A. K. (2006). Super resolution reconstruction of multispectral images. In: *Virtual observatory. Plate content digitalization, archive mining and image sequence processing*, Sofia, Bulgary, Vol. I, pp.211–220.

Nguyen, N., & Milanfar, P. (2000). Efficient wavelet-based algorithm for image superresolution. In: *Proceedings of the IEEE international conference on image processing*, Vancouver, B.C. Canada, Vol. 2, pp.351–354.

Papoulis, A. (1977). Generalized sampling theorem. *IEEE Transactions on Circuits and Systems, 24*, 652–654.

Schultz, R. R., Meng, L., & Stevenson, R. L. (1998). Subpixel motion estimation for super-resolution image sequence enhancement. *Journal of Visual Communication and Image Representation, 9*, 38–50.

Segall, C. A., Elad, M., Milanfar, P., Webb, R., & Fogg, C. (2004). Improved high-definition video by encoding at an intermediate resolution. In: *Proceedings of the SPIE conference on visual communications and image processing*, San Jose, CA, USA, Vol. 5308, pp.1007–1018.

Ur, H., & Gross, D. (1992). Improved resolution from sub-pixel shifted pictures. *CVGIP: Graphical Models and Image Processing, 54*, 181–186.

Biographical sketches

Dacil Barreto received the M.Sc degree in telecommunications engineering from the University of Las Palmas de Gran Canaria (ULPGC), Spain, in 2003. She is currently pursuing the Ph.D. degree in advanced telecommunications in the Research Institute for Applied Microelectronics (IUMA) at the ULPGC. Her technical interests include image processing, motion estimation and video compression.

Luis D. Alvarez-Corral was born in Granada, Spain, in 1977. He obtained the BS degree in Computer Science from the University of Granada in 2001 and the Advanced Studies Diploma (MS degree) in 2003, now he is pursuing his Ph. D. degree. His research interests include image and video resolution enhancement, particulary superresolution of compressed video within a Bayesian framework. Currently he is working for Indra Sistemas on image and video processing in Madrid.

Rafael Molina was born in 1957. He received the degree in mathematics (statistics) in 1979 and the Ph.D. degree in optimal design in linear models in 1983. He became Professor of computer science and artificial intelligence at the University of Granada, Granada, Spain, in 2000. His areas of research interest are image restoration (applications to astronomy and medicine), parameter estimation in image restoration, super resolution of images and video, and blind deconvolution. He is currently the Head of the computer science and Artificial Intelligence Department at the University of Granada.

Aggelos K. Katsaggelos received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979 and the M.S. and Ph.D. degrees both in electrical engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively. In 1985 he joined the Department of Electrical Engineering and Computer Science at Northwestern University, where he is currently professor. He is also the Director of the Motorola Center for Seamless Communications and a member of the Academic Affiliate Staff, Department of Medicine, at Evanston Hospital.

Dr. Katsaggelos is a member of the Publication Board of the IEEE Proceedings, the IEEE Technical Committees on Visual Signal Processing and Communications, and Multimedia Signal Processing, the Editorial Board of Academic Press, Marcel Dekker: Signal Processing Series, Applied Signal Processing, and Computer Journal. He has served as editor-in-chief of the IEEE Signal Processing Magazine (1997-2002), a member of the Publication Boards of the IEEE Signal Processing Society, the IEEE TAB Magazine Committee, an Associate editor for the IEEE Transactions on Signal Processing (1990-1992), an area editor for the journal Graphical Models and Image Processing (1992-1995), a member of the Steering Committees of the IEEE Transactions on Image Processing (1992-1997) and the IEEE Transactions on Medical Imaging (1990-1999), a member of the IEEE Technical Committee on Image and Multi-Dimensional Signal Processing (1992-1998), and a member of the Board of Governors of the IEEE Signal Processing Society (1999-2001). He is the editor of Digital Image Restoration (Springer-Verlag 1991), co-author of Rate-Distortion Based Video Compression (Kluwer 1997), and co-editor of Recovery Techniques for Image and Video Compression and Transmission, (Kluwer 1998). He was the holder of the Ameritech Chair of Information Technology (1997-2003), and he is the co-inventor of twelve international patents, a Fellow of the IEEE, and the recipient of the IEEE Third Millennium Medal (2000), the IEEE Signal Processing Society Meritorious Service Award (2001), and an IEEE Signal Processing Society Best Paper Award (2001).