Enhanced low bitrate H.264 video coding using decoder-side super-resolution and frame interpolation

Hasan F. Ates
Isik University
Department of Electrical and Electronics Engineering
Sile, Istanbul, Turkey
E-mail: hfates@isikun.edu.tr

Abstract. Advanced inter-prediction modes are introduced recently in literature to improve video coding performances of both H.264 and High Efficiency Video Coding standards. Decoder-side motion analysis and motion vector derivation are proposed to reduce coding costs of motion information. Here, we introduce enhanced skip and direct modes for H.264 coding using decoder-side super-resolution (SR) and frame interpolation. P- and B-frames are downsampled and H.264 encoded at lower resolution (LR). Then reconstructed LR frames are super-resolved using decoder-side motion estimation. Alternatively for B-frames, bidirectional true motion estimation is performed to synthesize a B-frame from its reference frames. For P-frames, bicubic interpolation of the LR frame is used as an alternative to SR reconstruction. A rate-distortion optimal mode selection algorithm is developed to decide for each MB which of the two reconstructions to use as skip/direct mode prediction. Simulations indicate an average of 1.04 dB peak signal-to-noise ratio (PSNR) improvement or 23.0% bitrate reduction at low bitrates when compared with H.264 standard. The PSNR gains reach as high as 3.00 dB for inter-predicted frames and 3.78 dB when only B-frames are considered. Decoded videos exhibit significantly better visual quality as well. [DOI: 10.1117/1.OE.52.7.071505]

Subject terms: H.264 video coding; low bitrate; super-resolution; frame interpolation.

Paper 121587SS received Oct. 31, 2012; revised manuscript received Feb. 15, 2013; accepted for publication Feb. 26, 2013; published online Mar. 21, 2013.

1 Introduction

Last decade has seen a dramatic decrease in bitrates required for video coding at a wide range of quality levels, first through the introduction of H.264 standard and currently with the standardization efforts for High Efficiency Video Coding (HEVC). This improvement in coding efficiency is due to contributions from various advanced coding tools, such as variable blocksize motion estimation (ME), rate-distortion (R-D) optimization, arithmetic coding, etc. With the introduction of each new coding tool, several new coding modes are defined for each macroblock (MB) of a video frame, causing both reduction in coding costs and also a substantial increase in computational complexity of the encoder and possibly the decoder. However, the rapid improvement in hardware speed and costs make these advanced techniques viable even for real-time video coding scenarios.

Despite the sophistication of the coding techniques used in current standards, there is still room for further improvement. For instance, in H.264 video coding standard, the use of several inter-prediction modes based on variable size partitioning of a MB is essential for the overall coding performance. However, these modes are not adequate for efficient coding especially at low bitrate regimes due to the relatively high cost of motion vector (MV) coding. When the quantization parameter (QP) is high and the compressed picture quality is low, MV coding becomes a bottleneck for the performance of H.264 standard. Simulations show that in video sequences with moderate-to-high motion content, an average of 35% to 40% of the total bitrate is spent for MVs at high QP values (i.e., when QP $>$ 40). Therefore, for high QP, skip mode in P-frames and direct mode in B-frames are the most commonly used modes as they do not require any bitrate. However, prediction accuracy of the skip/direct mode is limited by the accuracy of temporal or spatial neighboring MVs for the given MB.

In recent years, several new inter-prediction modes and MV coding techniques are introduced to improve MV encoding efficiency and prediction accuracy. For instance, parametric motion models are used for enhanced skip/direct mode prediction[1] and sprite prediction.[2] There exist several works that try to reduce the cost of differential MV coding by selecting from multiple MV predictors.[3–5] In Ref. 3, index of the best predictor is coded as side information, which is too costly for low bitrates. Therefore, number of predictor candidates is limited to just two. In Ref. 4, the best predictor is estimated at the decoder to reduce the cost of required side information. Similarly, Refs. 5 and 6 use a template matching technique to select the best predictor among multiple candidates. The ME at the decoder is used for improving H.264 coding efficiency as well.[7,8] Decoder-side MV derivation (DMVD)[9,10] uses template matching for decoder-side ME and is also tested in HEVC draft standard.[11] For each MB, the encoder selects whether to use DMVD or explicit MV coding, and this selection is coded as side information. Decoder-side ME (DSME)[7,12] applies bidirectional ME and motion compensated frame interpolation (MCFI) to generate an additional reference frame for improved prediction of a B-frame. At low bitrates B-frame coding could be totally

replaced by this MCFI frame, which does not require any MV or residual coding. However, the reported peak signal-to-noise ratio (PSNR) gains drop fast as the coding bitrate increases. In Ref. 13, DMVD and DSME approaches are combined for better overall coding performance.

In this paper, we introduce new skip and direct modes in H.264 standard based on decoder-side super-resolution (SR) and MCFI. The basic idea is to enhance skip/direct mode prediction using decoder-side motion analysis and estimation. The MCFI requires true ME at the decoder using already decoded reference frames. For SR, a low-resolution (LR) version of the current frame is transmitted to the decoder, and this leads to a coding redundancy because of the additional bitrate required for the LR frame. However, we show that for low-to-moderate bitrates, the improved coding efficiency for the high-resolution (HR) frame more than compensates the redundancy of coding the LR frame.

The SR-based coding techniques have been implemented before, in several mixed (hybrid) resolution coding scenarios. In distributed video coding, decoder-side SR is applied for reducing encoder complexity and improving decoder estimation accuracy.[14,15] The SR estimation could also be used for efficient scalable coding.[16] In similar other works,[17–19] SR-based reconstruction is used to to enhance H.264 coding performance. In the mixed resolution framework, some of the frames are labeled as key-frames and encoded at full resolution while other non key-frames are encoded at reduced resolution. The decoder makes use of decoded key-frames for SR-based reconstruction of non key-frames. In these approaches, SR is treated as a postprocessing step after decoding the LR frame. In other words, SR reconstruction is not part of the encoding/decoding procedure and R-D optimal encoding of the HR frame is not considered. Therefore, the proposed methods are only applicable and useful for very low bitrate and low quality encoding scenarios.

This paper also proposes an SR-based coding extension to H.264 standard for improved coding performance at low-to-moderate bitrates. In this proposed framework, P- and B-frames are downsampled and H.264 encoded at lower resolution. These LR frames are decoded and super-resolved using HR reference frames at the decoder. The SR reconstruction is used as skip/direct mode prediction during encoding of the original HR frame. For B-frames, an alternative reconstruction is also achieved based on MCFI using decoder-side bidirectional true ME and adaptive overlapped block motion compensation (AOBMC). For P-frames, MV extrapolation from references does not provide accurate prediction; therefore, instead of MCFI, bicubic interpolated LR frame is used as an alternative to the SR reconstructed frame. For each P- or B-frame, a R-D optimal mode selection algorithm decides which of the two alternative reconstructions to use as skip/direct mode prediction of a given MB.

The paper differentiates itself from existing work in two major aspects. First, it uses both SR- and MCFI-based reconstruction for B-frames and incorporates both methods into the encoding/decoding framework in a R-D optimal manner. In other words, the encoder also implements SR and MCFI algorithms and decides which prediction and which coding mode to use for the MBs of the HR frame. As a result, HR frame can be efficiently coded at a wider range of quality levels and bitrates. Second, the method does not need HR encoded key-frames as references for SR estimation, and P-frames as well as B-frames are coded using SR-based reconstruction. Therefore, performance improvements are not limited to non key-frames only, as in other mentioned methods. Simulations indicate an average of 1.04 dB PSNR improvement or 23.0% bitrate reduction at low-to-moderate bitrates when compared with H.264 standard. The PSNR gains reach as high as 3.00 dB for inter-predicted frames and 3.78 dB when only B-frames are considered. Decoded videos show significantly better visual quality as well.

Section 2 describes the general encoding/decoding framework. Section 3 explains the SR algorithm. Section 4 details the MCFI for B-frames. Section 5 is about R-D optimal mode selection and coding using these alternative skip/direct modes in a modified H.264 encoder. Section 6 presents simulation results and performance comparisons with H.264 reference software. Section 7 concludes the paper with a discussion of future extensions of the idea.

2 Video Coding Using Super-Resolution and Frame Interpolation

The ME at the decoder has proven to improve coding efficiency by reducing the bitrate required for MV coding and/or by improving the inter-prediction accuracy. This estimation is typically carried out using either bidirectional true ME from neighboring references frames[7] or a template-based MV search in which the template is defined in causal spatial neighborhood of the current MB.[8] However, true ME and subsequent MCFI tend to fail for regions with severe occlusion and when it is not constant speed translational motion. On the other hand, template-based spatial MV estimation suffers from sudden changes in the MV field at the object boundaries, where motion can no longer be assumed to be constant for neighboring blocks.

Hybrid resolution encoding is also proposed in literature,[17,18] in which some of the frames, named as key-frames are encoded at full resolution and others are downsampled and coded at lower resolution. The decoded LR frames are upsampled at the decoder using ME and SR-based reconstruction. Since HR key-frames are already available as references at the decoder, it is in principle possible to find accurate HR matches for the pixels/blocks of the LR frame and make a reasonable estimate of the original HR frame. Coding efficiency is achieved since significantly lower bitrate is required to encode LR frames instead of their HR counterparts. However, SR reconstruction quality is limited by the accuracy of the ME from the decoded LR frames. Hence, coding efficiency is possible only for very low bitrate/low PSNR coding scenarios.

In this paper, we propose a framework that incorporates both MCFI- and SR-based reconstruction techniques into the encoding/decoding processes. The MCFI and SR are not stand-alone modules at the decoder, but are actual parts of the coding flow where any required side information and MB-level mode selection are determined in a R-D optimal setting. The most general form of the coding framework is illustrated in Fig. 1. Figure 2 describes the encoding procedure. As in all predictive coders, the encoder needs to mimic the decoder behavior. Therefore, SR and MCFI algorithms are implemented at the encoder as well as the decoder.

In Fig. 2, P- and B-frames (c_H) are low-pass (LP) filtered, downsampled, and H.264 encoded at lower resolution (c_L).

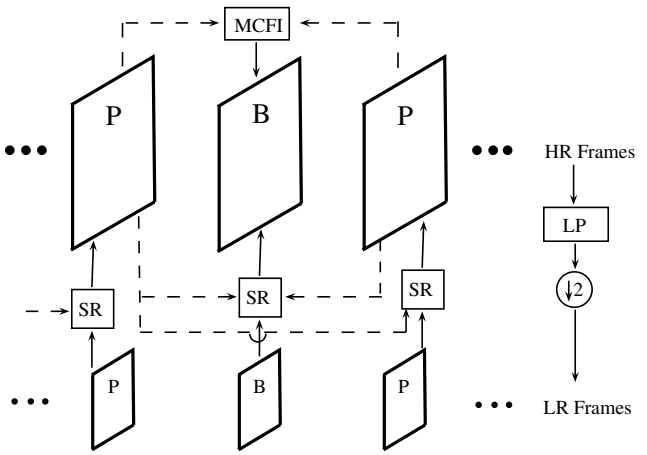


Fig. 1 SR and MCFI-based coding structure.

Previously encoded HR frames ($\hat{r}_0, \hat{r}_1$) are also LP filtered and used as references during LR encoding. Then SR method is applied to decoded LR frame ($\hat{c}_L$) to estimate the original HR frame (c_{SR}). The SR algorithm uses two closest HR reference frames to estimate the current HR frame. For B-frames, previous and next P-type HR reference frames are also used for MCFI (c_{FI}). For P-frames, MV prediction from previous reference frames is not successful; therefore, MCFI is not applicable. Instead, bicubic interpolation of LR frame (c_{BC}) is used as an alternative to SR reconstructed frame. A R-D optimal decision algorithm is applied at the frame and MB-levels to select which of the two alternatives (SR and MCFI reconstructions for B-frames, SR and bicubic interpolations for P-frames) to use as skip mode in P-frames and direct mode in B-frames. When necessary, this selection is signalled as side information to the decoder (see Sec. 5 for details). For B-frames, if the additional bitrate required for LR encoding is not justified in R-D sense, then SR reconstruction is fully discarded and only MCFI reconstruction is used as direct mode and this is signalled to the decoder by a 1-bit flag.

Decoding follows a similar flow as the encoding procedure. If LR version of the current frame is transmitted (which is always for P-frames and optional for B-frames), previously decoded HR frames are LP filtered and used as references in H.264 decoding of the LR frame. The decoded LR frame is bicubic interpolated and goes through the SR module. For B-frames, MCFI is applied to the HR reference frames. Then during decoding of the current HR frame, when a MB is signalled to be in skip/direct mode, mode selection module uses the side information transmitted by the encoder to select one of the two alternatives for predicting the current MB (see Sec. 5 for details).

The coding approach outlined above is significantly different than SR-based coding methods proposed in literature because SR process is treated as an integral part of the coding procedure as opposed to being a separate postprocessing step after decoding. This important novelty brings with it very interesting and original coding tools that have not been fully exploited before. This paper should be seen as only a first step in exploring and optimizing such SR-based coding techniques. However, care must be taken when formulating and designing the correct approach. It makes sense to provide alternative predictions for skip/direct MBs using SR, bicubic and MCFI reconstructions; however, these alternatives come with a cost in the form of bitrate spent for LR encoding and for signalling which alternative is actually chosen by the encoder. If not carefully optimized, this additional bitrate cost might overweigh the gains of using alternative predictions. These issues are discussed in more detail in Secs. 5 and 6.

3 Super-Resolution Based Reconstruction of P- and B-Frames

Successful SR-based reconstruction of current HR frame from its decoded LR counterpart and previously encoded HR reference frames is at the heart of the coding framework introduced in the previous section. When HR frame is well predicted, the additional bitrate required for HR encoding will be significantly reduced. Therefore, it is essential to design a robust and effective SR algorithm.

Within the given coding framework, SR estimation problem could easily benefit from the fact that HR coded versions of reference frames are already available. Therefore, some of the critical problems in SR reconstruction such as subpixel matching and regularized estimation are less of an issue within the given context. In Ref. 17, authors propose an example-based iterative reconstruction approach that uses consistency checking between HR and LR frames and outlier rejection for more robust reconstruction. In our work, we use a simpler one-pass approach for SR estimation in order to limit the additional complexity imposed on the decoder.

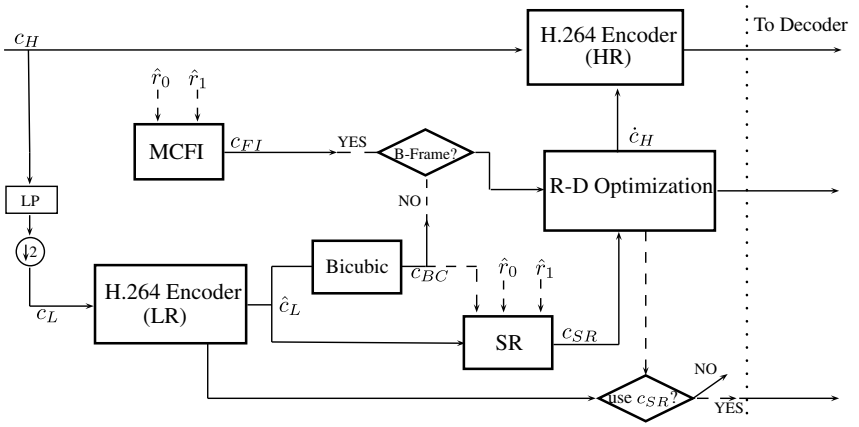


Fig. 2 Modified H.264 encoder.

Simulations show that the approach is effective for successful HR encoding.

Details of the algorithm are provided below. Decoded LR frame is bicubic interpolated first. Fast hexagonal block-based ME is performed between 8×8 blocks of the interpolated LR frame and LP filtered HR reference frames. Then each pixel of the SR frame is reconstructed based on the local matching error within a window centered on that pixel. Suppose that $\mathbf{v}_c^r$ represents MVs from frame c to reference r.

1. Bicubic interpolate decoded LR frame, $\hat{c}_L$, to get c_{BC},
2. LP filter decoded HR reference frames, $\hat{r}_i$, to get $\hat{r}_i^{\mathrm{LP}}$ $(i=0,1)$.
3. Initialize MVs between current and reference frames using the decoded MVs of $\hat{c}_L$; $\mathbf{v}_{c_H}^{r_i}(b_H)=2\mathbf{v}_{c_L}^{r_i}(b_L)$, where b_H and b_L are corresponding 8×8 and 4×4 subblocks of HR and LR frames, respectively. Note that, if $\mathbf{v}_{c_L}^{r_i}(b_L)$ is not available, then it is temporally scaled from the MV of other reference frame [e.g., for B-frame in Fig. 1, $\mathbf{v}_{c_H}^{r_0}(b_H)=-2\mathbf{v}_{c_L}^{r_1}(b_L)$].
4. Refine predicted MVs for each subblock b_H by performing fast hexagonal MV search (see Ref. 20). Repeat hexagonal search twice, using $\mathbf{v}_{c_H}^{r_i}(b_H)$ and (0,0) MVs as starting points, and select the result that gives minimum sum of absolute differences (SAD):

$$\mathrm{SAD}^i(\mathbf{v})=\sum_{(s,t)\in b_H}|c_{\mathrm{BC}}(s,t)-\hat{r}_i^{\mathrm{LP}}(s+v^x,t+v^y)| \tag{1}$$

5. For each pixel $(m,n)\in b_H$:

5.1 Compute $\mathrm{SAD}^i[\mathbf{v}_{c_H}^{r_i}(b_H)]$ using an $M\times M$ window centered on pixel (m,n) $(i=0,1)$.

5.2 If $\mathrm{SAD}^0>\mathrm{Th}_{\mathrm{SR}}$ and $\mathrm{SAD}^1>\mathrm{Th}_{\mathrm{SR}}$

$$c_{\mathrm{SR}}(m,n)=c_{\mathrm{BC}}(m,n) \tag{2}$$

else if $\mathrm{SAD}^0<\mathrm{SAD}^1$

$$c_{\mathrm{SR}}(m,n)=f\cdot\hat{r}_0(m_0,n_0)+(1-f)\cdot c_{\mathrm{BC}}(m,n) \tag{3}$$

else

$$c_{\mathrm{SR}}(m,n)=f\cdot\hat{r}_1(m_1,n_1)+(1-f)\cdot c_{\mathrm{BC}}(m,n), \tag{4}$$

where $m_i=m+v_i^x(b_H)$, $n_i=n+v_i^y(b_H)$ and $\mathbf{v}_{c_H}^{r_i}(b_H)=[v_i^x(b_H),v_i^y(b_H)]$.

In the algorithm, error threshold $\mathrm{Th}_{\mathrm{SR}}$ and weighting coefficient f are used to minimize any SR reconstruction artifacts due to incorrect MV estimation. When ME error is significantly high, bicubic interpolation result is preferable over SR estimate. When the error is below the given threshold, linear weighting between HR estimate and bicubic frame is used to achieve more consistent reconstruction.

Simulations show that this one-pass algorithm provides successful SR estimation results with fairly low complexity for the decoder. More advanced iterative solutions could in principle be designed to improve the accuracy of MV estimates and robustness of the SR reconstruction. However, the most critical issue for the performance seems to be the loss of information due to quantization in decoded LR frame, $\hat{c}_L$, and hence in c_{BC} as well. Especially at low bitrates, the quality of $\hat{c}_L$ determines how well MVs of the HR frame could be predicted. There exist SR algorithms in literature[21] that consider this quantization noise and propose a regularized solution for SR reconstruction. However, within the context of the coding framework given in this paper, the design of a more robust and more accurate SR algorithm is an open problem for future research.

4 Motion Compensated Frame Interpolation for B-Frames

In H.264 standard, MV used in B-frame direct mode is either temporally or spatially predicted from neighboring MBs. In temporal mode, MV of the co-located MB in the future reference frame is temporally scaled and used as the MV for direct mode prediction. However, directly applying scaled MVs of the reference frame to predict the current B-frame creates annoying blocking artifacts and deformations on the object boundaries. A true MV field is needed in occlusion regions, and a consistent and smooth MV field should be provided for blocks on the same motion boundary and/or moving object. In Ref. 22, we present an occlusion adaptive frame interpolation method that uses multiple MV postprocessing steps, which are motivated by the work of Huang et al.[23] This approach is also used in Ref. 24 for decoder-side true ME. In this paper, MCFI is performed based on a simplified version of this multistage MV correction and refinement algorithm together with AOBMC. For more details, see Ref. 22.

The proposed MCFI algorithm uses 8×8 subblocks for ME. Following steps are applied for true MV estimation and interpolation of the B-frame:

1. **MV initialization:** MV of each subblock b in B-frame is initialized by the decoded MV of the co-located block in the corresponding HR P-frame: $\mathbf{v}_c(b)=\mathbf{v}_{r_1}^{r_0}(b)$ and $-\mathbf{v}_c^{r_1}(b)=\mathbf{v}_c^{r_0}(b)=\mathbf{v}_c(b)/2$.
2. **MV reliability classification:** For 8×8 block b, accuracy of the initial MV, $\mathbf{v}_c(b)=(v^x,v^y)$, is evaluated based on total bidirectional prediction difference (BPD) between the two reference frames:

$$\mathrm{BPD}(\mathbf{v}_c(b))=\sum_{(s,t)\in b}\left|\hat{r}_0\left(s+\frac{v^x}{2},t+\frac{v^y}{2}\right)-\hat{r}_1\left(s-\frac{v^x}{2},t-\frac{v^y}{2}\right)\right|. \tag{5}$$

Based on the magnitude of BPD, blocks and their MVs are divided into two different classes: reliable (i.e., $\mathrm{BPD}\le\mathrm{Th}_{\mathrm{FI}}^1$) and unreliable (i.e., $\mathrm{BPD}>\mathrm{Th}_{\mathrm{FI}}^1$). Postprocessing stages of the algorithm are applied only to unreliable MVs.

3. **MV correction:** For this stage, a 16×16 MB B is considered unreliable if at least one of its four 8×8 subblocks is unreliable. For unreliable MBs, a common corrected MV is assigned to all its 8×8

subblocks. This MV is chosen from a subset of MV candidates formed by the original MVs of 4 subblocks and 12 neighboring subblocks of the MB [i.e., at most 16 different candidates, see Fig. 3(a)]. Optimal MV is selected as the one that minimizes BPD for the whole MB B:

$$\mathbf{v}_c^*(B) = \arg\min_{\mathbf{v}\varepsilon S_B}[\mathrm{BPD}(\mathbf{v})]. \tag{6}$$

Here, S_B denotes the set of MV candidates found in the neighborhood of current MB B [see Fig. 3(a)].

4. **MV re-classification:** After MV correction, unreliable blocks are re-classified into reliable and unreliable sets based on the updated BPD values and a new threshold value $\mathrm{Th}_{\mathrm{FI}}^2$ (i.e., for reliable blocks BPD $\leq \mathrm{Th}_{\mathrm{FI}}^2$, and for unreliable blocks BPD $> \mathrm{Th}_{\mathrm{FI}}^2$).
5. **MV refinement:** For still unreliable subblocks and their MVs, a reliability and similarity constrained vector median filter is applied:

$$\mathbf{v}_c^*(b) = \arg\min_{\mathbf{v}\in S_b} \sum_{\mathbf{v}_k\in S_b} w_k\|\mathbf{v}-\mathbf{v}_k\| \tag{7}$$

$$w_k = \begin{cases} 1, & \text{if } \mathbf{v}_k \text{ reliable and } d_k > \mathrm{Th}_{\mathrm{FI}}^3 \\ 0, & \text{else} \end{cases}, \tag{8}$$

where S_b contains MVs of the nine neighboring blocks around and including b [see Fig. 3(b)], and d_k denotes the distance between $\mathbf{v}_k$ and $\mathbf{v}_c(b)$ using the angular distance measure as defined below:

$$d_k = 1 - \frac{\mathbf{v}_k\mathbf{v}_c(b)}{\|\mathbf{v}_k\|\|\mathbf{v}_c(b)\|} = 1-\cos(\theta), \tag{9}$$

where θ is the angle between $\mathbf{v}_k$ and $\mathbf{v}_c(b)$. This metric is used for measuring the similarity of the candidate MVs and the original MV. Only those MVs from neighboring blocks that are reliable and not similar to block's current MV are used in vector median filtering of the unreliable MV.

6. **Adaptive overlapped block motion compensation:** After MV correction and refinement steps, there still exist blocks with high BPD and therefore unreliable MVs. For these blocks, AOBMC is applied to improve the consistency of the interpolated frame. The OBMC is a useful motion compensation (MC) technique that reduces blocking artifacts caused by conventional block based video coders.[25] In AOBMC, each block is synthesized as weighted average of multiple predictions using MVs of both the current block and its immediate horizontal and vertical neighboring blocks, as shown in Fig. 3(b) for $0 \leq k \leq 4$.

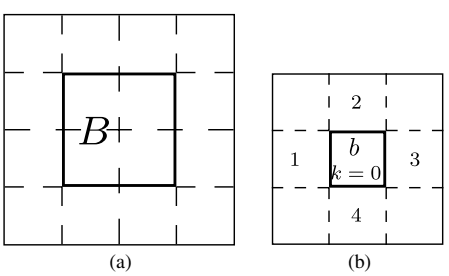


Fig. 3 Neighborhoods for MB and subblocks: (a) S_B; (b) S_b.

Suppose that $\mathbf{v}_k$ is the estimated MV for the k'th adjacent block $[\mathbf{v}_0 = \mathbf{v}_c(b)]$. For each $\mathbf{v}_k$, we generate the average of backward and forward predictions of the current block b $[(m,n)\in b, 0\leq k\leq 4]$:

$$p_k(m,n) = 0.5\hat{r}_0\left(m+\frac{v_k^x}{2}, n+\frac{v_k^y}{2}\right) + 0.5\hat{r}_1\left(m-\frac{v_k^x}{2}, n-\frac{v_k^y}{2}\right). \tag{10}$$

Hence p_k represents MC prediction of the block b for the candidate vector $\mathbf{v}_k$. The AOBMC is applied to unreliable blocks only to avoid oversmoothing the interpolated image. Therefore, the predicted frame is synthesized as follows:

If $\mathrm{BPD}[\mathbf{v}_c(b)] < \mathrm{Th}_{\mathrm{FI}}^4$,

$$c_{\mathrm{FI}}(m,n) = p_0(m,n), \tag{11}$$

else

$$c_{\mathrm{FI}}(m,n) = \sum_{k=0}^{4}\frac{w_k}{W_T}p_k(m,n), \tag{12}$$

where the weights w_k are chosen inversely proportional to BPD of corresponding MV $\mathbf{v}_k$:

$$w_k = \frac{1}{\mathrm{BPD}(\mathbf{v}_k)}, \qquad W_T = \sum_{k=0}^{4} w_k. \tag{13}$$

The proposed true ME algorithm, together with AOBMC, achieves successful interpolation of the B-frame with better PSNR and visual quality than H.264 temporal direct mode coding of the B-frame.[22] During MCFI, wrong MVs generally belong to occluded regions, moving periodic geometric structures (such as a barred gate) and blocks on the motion boundaries. In such cases, correct MV might be found in the neighborhood of the problematic block with high probability. This is the main reason why AOBMC improves the performance of MCFI, since it makes use of neighboring MVs during the synthesis of unreliable blocks.

5 Rate-Distortion Optimal Skip/Direct Mode Selection

Having performed LR frame encoding/decoding, bicubic interpolation, SR reconstruction and MCFI, the encoder is ready for the coding of original HR frame. In this paper, we propose to use c_{SR}, c_{BC} (for P-frames), c_{FI} (for B-frames) as alternative predictions for skip/direct modes in a modified H.264 encoder. At the MB level selecting between the two alternatives require 1-bit side information, which could be too costly for the skip/direct mode especially in the low bitrate regimes. Note that for regions with constant regular motion such as the background, c_{SR} and c_{FI} reconstructions are likely to be very similar and do not require a selection between the two. Likewise c_{SR} and c_{BC} are almost equal for MBs with smooth content or with unreliable MVs

because of Eq. (2). Therefore, such MBs could be detected by the decoder, and the encoder does not need to send any selection information.

Based on the above discussion, we propose the following R-D optimization approach for skip/direct predictor selection at the frame and MB level. Suppose $c_0 = c_{SR}$, $c_1 = c_{BC}$ for P-frames, $c_1 = c_{FI}$ for B-frames and c_H is the final prediction for the HR frame. For each MB, SAD is used as a measure of difference between the two predictions c_0 ands c_1. When this difference is larger than a given threshold Th_{RD}, a 1-bit side information is required for the decoder to decide which prediction to prefer. For all other MBs of the current frame in which the difference is lower than the threshold, a joint decision is made as to which of the two predictions to use. Algorithm 1 performs an exhaustive search to determine for the current frame the optimal threshold value that minimizes the total R-D cost of this predictor selection strategy. In the algorithm, λ_{MD} is the Lagrangian parameter used by H.264 encoder for R-D optimal mode decision. Hence, before H.264 encoding of the HR frame begins, the encoder selects the appropriate skip/direct predictor configuration that minimizes the total R-D cost for the whole frame. The encoder signals this selection to the decoder by coding Th_{RD}, $minFlag_{01}$ as side information for each frame. Note that for MBs that satisfy $SAD(c_0, c_1) > Th_{RD}$, 1-bit side information for predictor selection, $SkipMD(B)$, is included only if MB B is eventually encoded in skip/direct mode. The decoder follows Algorithm 2 to decode skip/direct MBs based on the transmitted parameters.

For B-frames, when compared to c_{FI}, SR reconstruction c_{SR} imposes an additional cost in the form of bitrate spent for encoding the LR frame. This bitrate and the quality of c_{SR} can be controlled through the choice of quantization parameter QP_{LR} for LR encoding, which will be further discussed in the next section. However, for certain B-frames when MCFI results in good quality synthesis c_{FI}, the encoder might choose to abort the LR encoding and hence SR reconstruction completely to improve the coding efficiency. This decision is made by comparing the minimum R-D cost defined in Algorithm 1 plus the cost of LR frame with the distortion cost of using c_{FI} only, as described in Algorithm 3. Here, R_{LR} is the bitrate spent for encoding the LR frame (1 additional bit is for $minFlag_{01}$), and $Flag_{DoSR}$ is a flag indicating whether decoder will receive LR encoded frame or not.

Having decided for each MB which prediction to use in skip/direct mode, the encoder then follows the typical steps of H.264 coding and performs ME and R-D optimal mode decision to select the best inter-prediction mode for each MB of the HR frame. In R-D optimal mode decision, actual distortion and coding rates are calculated for each mode. For skip/direct mode, the coding cost of $SkipMD(B)$ is included in the R-D cost unless $SAD(c_0, c_1) \leq Th_{RD}$ or $Flag_{DoSR} = 0$.

Algorithm 1 Skip/Direct Mode Optimization.

minRD = MAXVALUE

for threshold values Th = 500: 200: 2500 **do**

  $E_T = 0$, $E_0 = 0$, $E_1 = 0$, $co_{mb} = 0$

  for all MBs B **do**

    compute SAD between c_0, c_1

    compute sum of squared differences (SSD) between c_H and the two predictions

      $SSD(c_H, c_i) = \sum_{(s,t)\in B}[c_H(s,t) - c_i(s,t)]^2$

    if $SAD(c_0, c_1) > Th$ **then**

      Increment co_{mb} by 1.

      if $SSD(c_H, c_0) < SSD(c_H, c_1)$ **then**

        $SkipMD(B) = 0$

        $E_T = E_T + SSD(c_H, c_0)$

      else

        $SkipMD(B) = 1$

        $E_T = E_T + SSD(c_H, c_1)$

      end if

    else

      $E_0 = E_0 + SSD(c_H, c_0)$

      $E_1 = E_1 + SSD(c_H, c_1)$

    end if

  end for

  if $E_0 < E_1$ **then**

    $Flag_{01} = 0$

    $E_T = E_T + E_0$

  else

    $Flag_{01} = 1$

    $E_T = E_T + E_1$

  end if

  if $E_T + \lambda_{MD} \cdot co_{mb} < minRD$ **then**

    $minRD = E_T + \lambda_{MD} \cdot co_{mb}$

    $Th_{RD} = Th$

    $minFlag_{01} = Flag_{01}$

  end if

end for

6 Simulation Results

6.1 *Testing Conditions*

The developed coding algorithm, named as H.264 + SR+ FI, is incorporated into H.264 reference software version 16.1. Parameter values used in the tests are as follows: For SR algorithm $M = 7$, $Th_{SR} = 18M^2$, $f = 0.95$, and a separable 3×3 LP filter with 1-D kernel $h = [0.25, 0.5, 0.25]$;

Algorithm 2 Skip/Direct Mode Decoding.

if MB B coded as skip/direct **then**

compute SAD between c_0, c_1

if $\mathrm{SAD}(c_0, c_1) > \mathrm{Th}_{\mathrm{RD}}$ **then**

if $\mathrm{SkipMD}(B) == 0$ **then**

$\dot{c}_H(m,n) = c_0(m,n), \quad \forall\ (m,n) \in B$

else

$\dot{c}_H(m,n) = c_1(m,n), \quad \forall\ (m,n) \in B$

end if

else

if $\mathrm{minFlag}_{01} == 0$ **then**

$\dot{c}_H(m,n) = c_0(m,n), \quad \forall\ (m,n) \in B$

else

$\dot{c}_H(m,n) = c_1(m,n), \quad \forall\ (m,n) \in B$

end if

end if

end if

For MCFI algorithm $\mathrm{Th}_{\mathrm{FI}}^1 = \mathrm{Th}_{\mathrm{FI}}^2 = 500$, $\mathrm{Th}_{\mathrm{FI}}^3 = 0.15$, $\mathrm{Th}_{\mathrm{FI}}^4 = 320$. Simulations show that results are not too sensitive to parameter values as long as they are reasonably chosen. Hence, no exhaustive testing is needed to optimize these set values. A more important parameter for coding efficiency is the quantization parameter used for LR encoding, $\mathrm{QP}_{\mathrm{LR}}$. For the simulations in Sec. 6.2, we set $\mathrm{QP}_{\mathrm{LR}} = \mathrm{QP}_{\mathrm{HR}} - 4$ for P-frames and $\mathrm{QP}_{\mathrm{LR}} = \mathrm{QP}_{\mathrm{HR}} - 3$ for B-frames.

Algorithm 3 LR Encoding Decision for B-frames

compute total distortion cost of c_{FI}

$$D(c_{\mathrm{FI}}) = \sum_B \mathrm{SSD}(c_H, c_{\mathrm{FI}})$$

if $D(c_{\mathrm{FI}}) > \mathrm{minRD} + \lambda_{\mathrm{MD}} \cdot (R_{\mathrm{LR}} + 1)$ **then**

$\mathrm{Flag}_{\mathrm{DoSR}} = 1$

Transmit LR frame bits

else

$\mathrm{Flag}_{\mathrm{DoSR}} = 0$

Do not transmit LR frame bits

end if

In Sec. 6.3, the effect of $\mathrm{QP}_{\mathrm{LR}}$ on overall coding performance of the algorithm will be discussed in more detail.

Simulations are performed for video sequences foreman (CIF, 30 Hz, 300 frames), running girl (part 1,2) (SD, 25 Hz, (150,140) frames), soccer (SD, 60 Hz, 300 frames), crew (SD, 60 Hz, 300 frames), ice (SD, 60 Hz, 240 frames), RaceHorses (832×480, 30 Hz, 200 frames), BasketballDrill (832×480, 50 Hz, 200 frames). In the following, IBPBP... GOP structure is tested, where only first frame is coded as full-resolution I-frame. Intra-modes are disabled in inter-predicted frames. Enhanced Predictive Zonal Search ME algorithm is used during encoding with a search range of $[-32, 32]$. The R-D optimized mode decision and CAVLC encoding are used. In B-frames, subblock sizes smaller than 8×8 are switched off. The P- and B-frames are coded with $\mathrm{QP}_{\mathrm{HR}} = 36, 40, 44, 48$.

6.2 *Comparison with H.264 Standard*

Performance of H.264 + SR + FI algorithm is tested against the H.264 standard. Tables 1 and 2 present two set of simulation results at full frame rates and half frame rates, respectively. The tables give average PSNR gain for Y-component (Luminance) in dB (at equal bitrates) and percentage decrease in bitrate (at equal PSNR) with respect to H.264 standard, as described in Ref. 26.

As seen in Table 1, H.264 + SR + FI achieves an average PSNR gain of 1.04 dB or average bitrate reduction of 23.0% when compared with H.264 reference encoder. One important observation about the results is the variation of performance among tested video sequences. While PSNR gain goes as high as 3 dB for running girl-1, it drops to 0.2 dB for BasketballDrill. The proposed algorithm is especially successful for videos without too much spatial detail and with fast and regular motion in the scene. As a matter of fact, simulations show that some PSNR loss is possible for sequences such as city, where there is significant texture and high-frequency detail which is lost during downsampling and cannot be successfully recovered with SR procedure. There is a more complicated relationship between motion content of the video and performance of the algorithm. When BasketballDrill and running girl-2 are compared, we see that running girl-2 provides much better results even though both sequences have similar levels of spatial detail and motion content. The main difference between the two videos lies in the usage of skip/direct modes; when both sequences are H.264 encoded with QP = 36, 74%, and 45% of all MBs are encoded in skip/direct mode for BasketballDrill and running girl-2, respectively. This means that skip/direct MV prediction in H.264 is already quite accurate for BasketballDrill and there is not much room for improvement by SR and MCFI-based estimation. On the other hand, even though they are two separate scenes of the same video, there is a major difference in PSNR gains for running girl-1 and running girl-2. In fact running girl-1 uses a higher percentage of skip/direct modes than running girl-2. The difference in this case is that running girl-1 has more regular background motion (i.e., camera motion) and almost 2-D translational object motion occluding the background. This constant speed, regular and translational motion content is especially suitable for SR and MCFI-based estimation. To summarize, we can say that H.264 + SR + FI achieves best results for video sequences in which motion

Table 1 Performance evaluation of H.264 + SR + FI (full fps).

	P- and B-frames		B-frames only	
Video	δ PSNR (dB)	δ bitrate (%)	δ PSNR (dB)	δ bitrate (%)
foreman	0.35	−12.0	−0.05	+1.4
run. girl-1	3.00	−42.5	3.78	−51.5
run. girl-2	1.27	−28.6	1.70	−41.5
soccer	0.71	−22.9	0.47	−18.4
crew	1.65	−39.2	2.24	−57.1
ice	0.69	−16.4	0.20	−5.8
BasketballDrill	0.20	−5.9	0.65	−23.7
RaceHorses	0.46	−16.6	0.76	−32.6
Average	**1.04**	**−23.0**	**1.22**	**−28.7**

is fast and complex so that it cannot be predicted well by H.264 skip/direct MVs (e.g., large occluded regions) but not too complex so that SR- or MCFI-based estimation works.

In Table 2 for videos with half number of frames per second (fps), H.264 + SR + FI achieves an average PSNR gain of 1.40 dB or average bitrate reduction of 29.4% when compared with H.264 reference encoder. This substantially improved performance at lower frame rates confirms our observation that the stronger the motion content the better the algorithm performs. When frame rate is reduced, size of MVs between consecutive frames becomes larger, occlusion regions grow bigger and skip/direct MV prediction in H.264 becomes less successful. On the other hand, SR and MCFI algorithms are not that much affected by the increase in temporal distance between the current and reference frames. Also percentage of bit budget spent for MV coding increases with the reduction of frame rate, making it relatively more valuable to improve MV coding efficiency.

Tables 1 and 2 also present coding gains for B-frames only, without considering P-frame PSNR and bitrates. The performance of H.264 + SR + FI is significantly better for B-frames than for P-frames (except for foreman, ice, and soccer at full fps, in which very little bitrate is spent for B-frame coding since H.264 direct mode is mostly sufficient). This performance improvement is partially due to the use of MCFI as an alternative to SR-based estimation. In fact, when MCFI estimation is sufficiently accurate, LR encoding and SR reconstruction are turned off, which avoids the additional cost of bits spent for LR encoded frame. Unfortunately, such "low-cost" estimation for P-frames is not straightforward; prediction of MVs from previous reference frames does not achieve the desired level of accuracy. One solution might be the use of spatial template-based estimation, just like DMVD; however, a spatial template uses causal neighboring decoded blocks, while skip mode selection in Sec. 5 is applied before encoding of the HR frame begins.

In B-frames, contributions of SR and MCFI estimations to the overall coding performance are analyzed by switching off MCFI and using SR reconstruction only. Table 3 compares PSNR gains of H.264 + SR + FI with those of SR-only version of the algorithm. While SR reconstruction is dominantly preferred in some sequences such as running girl-1, we notice comparable contributions from both SR and MCFI in soccer and a higher contribution from MCFI estimate in foreman. Therefore, the use of MCFI as an alternative provides significant boost to the coding performance in some of the tested sequences.

Figure 4 presents PSNR versus bitrate plots for some of the tested video sequences in low-to-moderate bitrate

Table 2 Performance evaluation of H.264 + SR + FI (half fps).

	P- and B-frames		B-frames only	
Video	δ PSNR (dB)	δ bitrate (%)	δ PSNR (dB)	δ bitrate (%)
foreman	0.71	−20.0	1.14	−33.0
run. girl-1	3.95	−50.3	6.19	−64.2
run. girl-2	1.59	−34.0	2.49	−51.6
soccer	1.35	−37.2	1.68	−47.2
crew	1.58	−36.6	2.31	−57.9
ice	0.84	−19.1	0.98	−24.3
BasketballDrill	0.53	−16.2	1.31	−39.1
RaceHorses	0.67	−22.1	1.28	−44.1
Average	**1.40**	**−29.4**	**2.17**	**−45.2**

Table 3 PSNR gains for B-frames for H.264 + SR + FI and SR-only version (full fps).

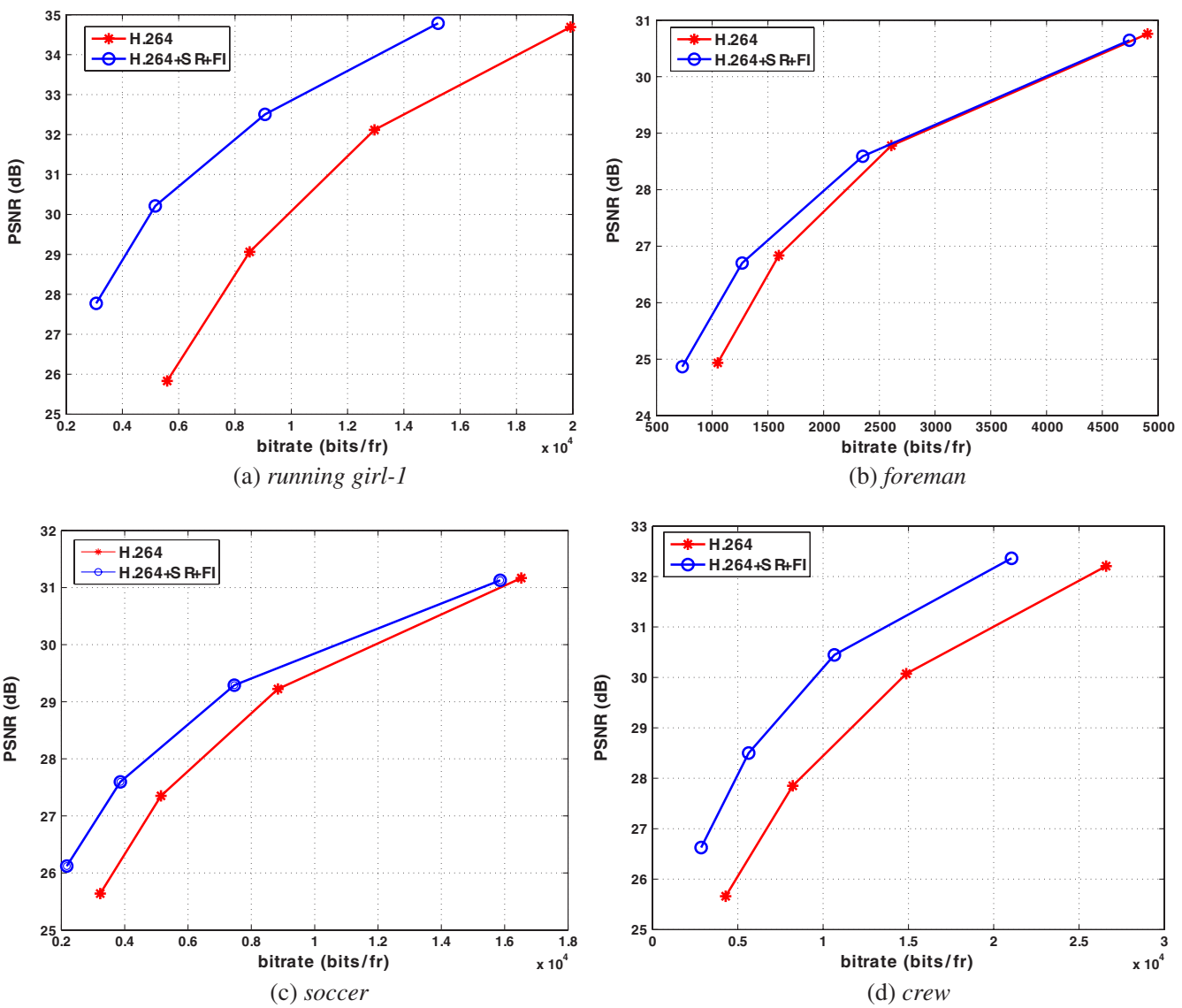
	H.264+SR+FI	SR-only
Video	δ PSNR (dB)	δ PSNR (dB)
foreman	−0.05	−0.77
run. girl-1	3.78	3.72
run. girl-2	1.70	1.58
soccer	0.47	0.21
crew	2.24	1.54
ice	0.20	−0.20
BasketballDrill	0.65	0.52
RaceHorses	0.76	0.64
Average	**1.22**	**0.91**

regimes (i.e., when $QP_{HR} = 36, 40, 44, 48$). The PSNR gains over H.264 standard drop as bitrate increases, even though the reduction in coding gains is typically not too dramatic. However, as bitrate increases further, sooner or later PSNR for H.264 + SR + FI will go below PSNR of H.264 standard, mainly due to the redundant bitrate spent for LR encoding. As bitrate increases, percentage of MBs coded in skip/direct mode and percentage of bit budget spent for MV coding decrease. Hence at higher bitrates, accurate skip/direct mode prediction becomes a much less critical issue for coding efficiency, making SR- and MCFI-based estimation less useful. However, there are several ways in which the performance of H.264 + SR + FI could be further improved, and these are discussed as open research problems in the next section.

Visual comparisons also confirm the superiority of the proposed approach over H.264 reference encoder especially at low bitrates. Figure 5 shows selected frames from running girl-1 and crew. Frames with almost equal bitrate are chosen for comparison. In both figures, H.264 + SR + FI encoded frames have fewer artifacts especially around object boundaries and in occlusion regions. In Fig. 5(b), H.264 + SR + FI provides a more consistent synthesis around running girl's legs, hands, hair, and body. In Fig. 5(d), astronauts' legs are better reconstructed when compared with Fig. 5(c). In Fig. 6, selected regions are zoomed in for better comparison of the reconstruction quality. Overall, H.264 + SR + FI algorithm has significantly better R-D and visual performance in video sequences with fast motion and large occluded areas such as running girl-1.

6.3 *Selection of QP_{LR}*

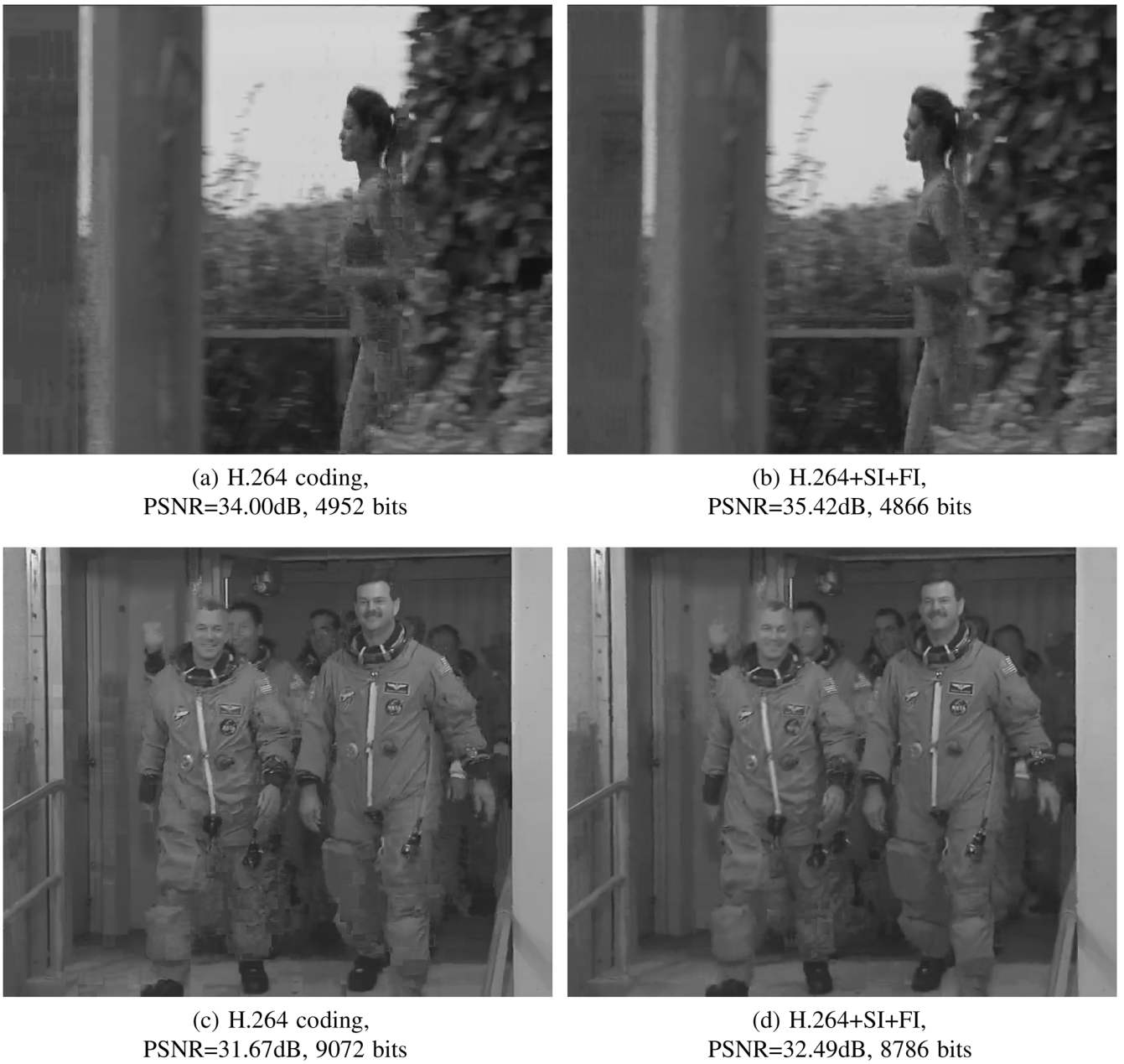
As discussed in previous section, bitrate spent for LR frame is an important factor that effects the overall coding efficiency of H.264 + SR + FI. As QP_{LR} is increased, LR bitrate decreases but so does the quality of SR and bicubic interpolated frames. Hence, it is important to determine an optimal value for QP_{LR} based on the HR coded video quality and overall bitrate. In the above simulations we set $QP_{LR} = QP_{HR} - 4$ for P-frames and $QP_{LR} = QP_{HR} - 3$ for B-frames,



(a) *running girl-1* (b) *foreman*

(c) *soccer* (d) *crew*

Fig. 4 PSNR versus bitrate plots: (a) running girl-1; (b) foreman; (c) soccer; (d) crew.



(a) H.264 coding, PSNR=34.00dB, 4952 bits

(b) H.264+SI+FI, PSNR=35.42dB, 4866 bits

(c) H.264 coding, PSNR=31.67dB, 9072 bits

(d) H.264+SI+FI, PSNR=32.49dB, 8786 bits

Fig. 5 Visual comparisons for running girl-1 [(a), (b)] and crew [(c), (d)].

which provides a good trade-off between LR coding redundancy and HR frame prediction accuracy. Note that QP_{LR} for B-frames is set one level higher than for P-frames since SR frame quality is slightly less critical due to the alternative MCFI estimation.

Simulations show that optimal value of QP_{LR} is dependent on several factors such as video/frame content and overall bitrate. Videos such as running girl-1 can benefit from a high quality coded LR frame and SR reconstruction, while videos such as foreman have to be more conservative in the additional bitrate spent for LR frame. Also using a very high QP_{LR} is not desirable; since, when quantization noise is too high in the LR frame, decoder-side ME during SR fails to find correct MVs. Therefore, as QP_{HR} increases and overall bitrate decreases, the difference between QP_{HR} and QP_{LR} should increase. On the contrary, as QP_{HR} decreases, LR coding redundancy starts to become the bottleneck for performance and therefore QP_{LR} should be set closer or sometimes even higher than QP_{HR}.

In order to understand the affects of QP_{LR} on the coding performance, tested videos are encoded using several different QP_{LR} levels. Then the convex hull of the R-D curve is determined for each video. That means, for each video and for each QP_{HR}, optimal value of QP_{LR} with minimum R-D cost is determined. Table 4 presents coding gains achieved with the use of optimal QP_{LR} over the fixed QP_{LR} case as in Sec. 6.2. Some videos such as ice benefit substantially from the use of optimal QP_{LR} values.

Even though some of the results in Table 4 are promising, it is obviously not practical to encode each video several times and select the best QP_{LR}. In fact, the encoder needs to determine QP_{LR} based on the video content without having

(a) H.264 coding (b) H.264+SI+FI (c) H.264 coding (d) H.264+SI+FI

Fig. 6 Zoomed regions for running girl-1 [(a), (b)] and crew [(c), (d)].

to perform full encoding. This requires a better understanding of the relationship between video properties (such as spatial detail and level of motion) and optimal QP_{LR}. If such a model could be developed, then QP_{LR} could be chosen adaptively for each frame of the video sequence, possibly leading to much better overall performance. However, this adaptive extension for the algorithm is outside the scope of our paper and left as future research.

Selecting an appropriate QP_{LR} is actually not the only concern when dealing with LR coding redundancy. For MBs of the HR frame that are coded in any mode other than skip/direct, bitrate spent for coding the corresponding 8×8 subblocks of the LR frame can be seen as redundant information. However, it is not easy to remove this redundancy because of the intricate dependencies between various steps of the algorithm. Even if the encoder could preemptively determine which MBs will not be coded in skip/direct code, it is not clear how this information could be used during LR coding. Reducing the coding quality of the corresponding LR subblocks might adversely affect the overall LR frame and SR reconstruction quality, leading to even worse HR coding results. A successive refinement of the coding decisions based on iterative LR and HR coding is also not practical because of the substantial increase in encoder and possibly also decoder complexity. Another approach might be trying to make use of LR data and SR estimation in other inter-modes as well. For instance LR data could be used to estimate low-frequency DCT coefficients of HR MB and DCT residual coding could be performed for high-frequency coefficients only. These interactions between LR and HR encoding leave open several interesting coding problems for future research.

Table 4 Performance gains for optimal QP_{LR} (full fps).

	P- and B-frames	
Video	δ PSNR (dB)	δ bitrate (%)
foreman	0.08	−2.4
run. girl-1	0.25	−7.0
run. girl-2	0.15	−5.0
soccer	0.16	−6.4
crew	0.15	−5.2
ice	0.22	−5.4
BasketballDrill	0.09	−3.4
RaceHorses	0.10	−3.5
Average	**0.15**	**−4.8**

6.4 *Computational Complexity*

Simulations are run on a PC with Intel Core 2 Quad CPU at 2.40 GHz. Table 5 provides the percentage increase in execution times for H.264 + SR + FI encoding and decoding when compared with H.264 reference software. Average execution times are 56% and 298% higher for encoding and decoding, respectively. This computational load is expected because of the ME and MV refinement algorithms performed for SR and MCFI both at the encoder and the decoder.

For encoding, about 30% increase in execution time is due to LR encoding of the current frame. The LR encoding also introduces approximately 25% memory overhead. The SR, MCFI, and R-D optimization algorithms contribute roughly 15%, 6%, and 5% additional complexity, respectively. For decoding, since SR procedure is applied independently for each 8×8 subblock, SR estimation is not executed for subblocks that are not encoded in SR-based skip/direct mode. However, MV correction and refinement in MCFI require this algorithm to be applied throughout the whole frame, irrespective of whether MCFI estimate is actually needed in decoding or not. As a result, average contribution of SR and MCFI algorithms to decoding complexity becomes 170% and 105%, respectively.

In this paper, SR and MCFI algorithms are designed for optimal coding performance, without consideration of the

Table 5 Complexity evaluation of H.264 + SR + FI and its decoding-optimized version.

	H.264 + SR + FI		Complexity optimization	
Video	Encoder δT (%)	Decoder δT (%)	Decoder δT (%)	δ PSNR (dB)
foreman	+49	+270	+200	−0.06
run. girl-1	+62	+400	+240	+0.05
run. girl-2	+56	+340	+210	−0.03
Soccer	+59	+325	+230	−0.04
Crew	+54	+210	+160	−0.01
Ice	+59	+300	+220	−0.10
BasketballDrill	+57	+240	+200	−0.01
RaceHorses	+52	+300	+190	−0.04
Average	**+56**	**+298**	**+206**	**−0.03**

computational complexity. However, simulations indicate that these algorithms could be substantially simplified with little loss of coding efficiency. As an example for improving computational performance, we provide a simple R-D and complexity joint optimization approach to reduce decoding complexity in B-frames. The main idea is to skip MCFI or SR during decoding when it is profitable to do so. The encoder makes this decision based on the trade-off between coding loss and complexity reduction of skipping either one of the two algorithms. For that purpose, Algorithm 3 of Sec. 5 is modified as given in Algorithm 4.

In Algorithm 4, T_{SR} stands for the total SR execution time for MBs using SR reconstruction, T_{FI} is the MCFI time for the whole frame, ΔT_{SR-FI} is the additional SR time needed during decoding if MCFI is skipped and SR estimate is used for the whole frame. Flag_{DoFI} is a 1-bit flag transmitted to decoder to signal whether MCFI is skipped or not. The algorithm considers both R-D costs and decoding complexities in selecting either one of the three options for a B-frame: SR-only, MCFI-only, or joint use of SR and MCFI estimates. λ_C is appropriately chosen for each tested video sequence to provide a reasonable trade-off between decoding complexity reduction and loss of coding efficiency. Table 5 shows that decoding complexity is reduced by an average of 92% (relative to H.264 decoder) with marginal average 0.03 dB PSNR loss. In fact, a slight improvement in PSNR is observed for running girl-1, in which SR estimate is preferred for most B-frames and skipping MCFI saves the additional bits used for encoding SkipMD(B) flags. These results illustrate that there is substantial computational redundancy in H.264 + SR + FI. However, the focus of this paper is the optimized coding efficiency. Any further algorithmic simplifications to reduce encoding/decoding complexity is left as future work.

Algorithm 4 SR/MCFI Skip Decision for B-frames.

compute total distortion cost of c_{FI}

compute total distortion cost of c_{SR}

if $D(c_{FI}) < \text{minRD} + \lambda_{MD} \cdot (R_{LR} + 1) + \lambda_C \cdot T_{SR}$ **then**

  $\text{Flag}_{DoSR} = 0$

  Do not transmit LR frame bits

else

  $\text{Flag}_{DoSR} = 1$

  Transmit LR frame bits

  if $D(c_{SR}) + \lambda_C \cdot \Delta T_{SR-FI} < \text{minRD} + \lambda_C \cdot T_{FI}$ **then**

    $\text{Flag}_{DoFI} = 0$

  else

    $\text{Flag}_{DoFI} = 1$

  end if

end if

7 Conclusion

In this paper, we developed decoder-side SR and MCFI algorithms for improving H.264 coding efficiency at low-to-moderate bitrates. The P- and B-frames of the video are LP filtered, downsampled and H.264 encoded at lower resolution. These LR frames are bicubic interpolated and used for SR reconstruction of HR frame. For B-frames a multistage true ME algorithm is developed for MCFI of the HR frame. Bicubic interpolation, SR estimation and MCFI results are used as alternative predictions for skip/direct modes during H.264 encoding of the HR frame. A R-D optimal skip/direct mode selection algorithm is also proposed for choosing between alternative predictions at the frame and MB level. Simulations indicate significant coding gains over H.264 standard especially for videos without too much spatial detail or texture and with strong, regular camera and object motion that causes large occlusion regions.

The use of SR and MCFI based estimation as an integral part of the encoding flow differentiates this paper from existing SR-based hybrid resolution coding methods. In other words, SR algorithm is not treated as a separate post-processing step after decoding, but it becomes part of the codec and therefore could be optimized by the encoder for best possible coding results. However, such an optimization requires understanding of the intricate dependencies between LR coding, SR estimation, and HR coding modules. This paper is a first step in analyzing this new coding framework and developing the appropriate coding tools for it.

There are several directions in which future research could take place. Minimizing LR coding redundancy is essential for improving the algorithm's performance especially at higher bitrates. Optimal adaptation of QP_{LR} at the

frame and possibly MB levels could be a partial solution to this problem. More generally, LR coding decisions should be optimized not for the LR frame but for the HR frame. For instance, a higher or lower amount of bitrate could be spent for LR residual coding and/or LR MV coding depending on how much it helps improve SR estimation accuracy. More advanced SR algorithms that are robust against quantization errors should also be investigated. For P-frames, a lower cost alternative to SR estimation is needed, just like the use of MCFI in B-frames. Spatial template-based ME could be a solution but mode selection algorithm needs to be modified accordingly. Also SR- and MCFI-based estimation could be used to define new inter-modes for H.264, where MB could be jointly estimated using normal MC together with these alternatives. Finally, the developed coding framework could be easily extended to be used in HEVC standard as well.

Acknowledgments

This research was supported by TÜBİTAK Career Grant 108E201.

References

1. A. Glantz et al., "A block-adaptive skip mode for inter prediction based on parametric motion models," in *Proc. 18th IEEE Int. Conf. on Image Process.*, pp. 1201–1204, IEEE, Brussels (2011).
2. A. Krutz et al., "Rate-distortion optimized video coding using automatic sprites," *IEEE J. Sel. Top. Signal Process.* **5**(7), 1309–1321 (2011).
3. G. Laroche, J. Jung, and B. Pesquet-Popescu, "RD optimized coding for motion vector predictor selection," *IEEE Trans. Circ. Syst. Video Technol.* **18**(9), 1247–1257 (2008).
4. J. Dai et al., "Motion vector coding based on predictor selection and boundary-matching estimation," in *Proc. IEEE Int. Workshop on Multimedia Signal Process.*, pp. 1–5, IEEE, Rio De Janeiro (2009).
5. K. Won, J. Yang, and B. Jeon, "Motion vector coding using decoder-side estimation of motion vector," in *Proc. IEEE Int. Symp. on Broadband Multimedia Syst. and Broadcasting*, pp. 1–4, IEEE, Bilbao (2009).
6. W. Yang et al., "An efficient motion vector coding algorithm based on adaptive predictor selection," in *Proc. IEEE Int. Symp. on Circ. and Syst.*, pp. 2175–2178, IEEE, Paris (2010).
7. S. Klomp et al., "Decoder-side block motion estimation for H.264/MPEG-4 AVC based video coding," in *Proc. IEEE Int. Symp. on Circ. and Syst.*, pp. 1641–1644, IEEE, Taipei (2009).
8. S. Kamp, M. Evertz, and M. Wien, "Decoder side motion vector derivation for inter frame video coding," in *Proc. 15th IEEE Int. Conf. on Image Process.*, pp. 1120–1123, IEEE, San Diego, California (2008).
9. S. Kamp, B. Bross, and M. Wien, "Fast decoder side motion vector derivation for inter frame video coding," in *Proc. IEEE Picture Coding Symp.*, pp. 1–9, IEEE, Chicago, Illinois (2009).
10. S. Kamp and M. Wien, "Decoder-side motion vector derivation for hybrid video inter coding," in *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 1277–1280, IEEE, Suntec City (2010).
11. S. Kamp and M. Wien, "Description of video coding technology proposal by RWTH Aachen University," presented at *JVT on Video Coding of ITU-T VCEG and ISO/IEC MPEG 1st Meeting*, JCTVC, Dresden (2010).
12. S. Klomp and J. Ostermann, "Motion estimation at the decoder," in *Effective Video Coding for Multimedia Applications*, InTech, Rijeka, Croatia (2011).
13. S. Klomp, M. Munderloh, and J. Ostermann, "Decoder-side motion estimation assuming temporally or spatially constant motion," *ISRN Signal Process.* **2011**, 1–10 (2011).
14. R. Klepko, D. Wang, and G. Huchet, "Combining distributed video coding with super-resolution to achieve H.264/AVC performance," *J. Electron. Imag.* **21**(1), 013011 (2012).
15. E. M. Hung et al., "Video super-resolution using codebooks derived from key-frames," *IEEE Trans. Circ. Syst. Video Technol.* **22**(9), 1321–1331 (2012).
16. M. Shen, P. Xue, and C. Wang, "A novel scalable video coding scheme using super resolution techniques," in *Proc. IEEE Workshop Multimedia Signal Process.*, pp. 196–199, IEEE, Cairns, Queensland (2008).
17. M. Shen, P. Xue, and C. Wang, "Down-sampling based video coding using super-resolution technique," *IEEE Trans. Circ. Syst. Video Technol.* **21**(6), 755–765 (2011).
18. S. H. Lee and N. I. Cho, "Low bit rates video coding using hybrid frame resolutions," *IEEE Trans. Consum. Electron.* **56**(2), 770–776 (2010).
19. Z. Pan and H. Xiong, "Sparse spatio-temporal representation with adaptive regularized dictionaries for super-resolution based video coding," in *Data Compression Conf.*, pp. 139–148, IEEE, Snowbird, Utah (2012).
20. C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circ. Syst. Video Technol.* **12**(5), 349–355 (2002).
21. C. A. Segall et al., "Bayesian resolution enhancement of compressed video," *IEEE Trans. Image Process.* **13**(7), 898–911 (2004).
22. B. Cizmeci and H. F. Ates, "Occlusion aware motion compensation for video frame rate up-conversion," in *Proc. IASTED Int. Conf. Signal and Image Process.*, ACTA Press, Maui, Hawaii (2010).
23. A. M. Huang and T. Q. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.* **17**(5), 694–708 (2008).
24. H. F. Ates and B. Cizmeci, "Decoder side true motion estimation for very low bitrate B-frame coding," in *Proc. 18th IEEE Int. Conf. Image Process.*, pp. 1673–1676, IEEE, Brussels (2011).
25. M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach," *IEEE Trans. Image Process.* **3**(5), 693–699 (1994).
26. G. Bjøntegaard, "Calculation of average PSNR differences between R-D curves," in *Doc. VCEG-M33*, JVT, Austin (2001).

Hasan F. Ates received his BS degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 1998, and the MA and PhD degrees from the Department of Electrical Engineering, Princeton University, Princeton, New Jersey, in 2000 and 2004, respectively. He was a postdoctoral research associate at Sabanci University, Istanbul, between 2004 and 2005. He is currently an associate professor in the Department of Electrical and Electronics Engineering, Isik University, Istanbul, Turkey, which he joined in August 2005. His research interests include image, video and graphics compression, video enhancement, wavelets and multiresolution representations, and computer vision. He is currently working on industrial- and government-sponsored projects related to video coding, super-resolution, video surveillance and content analysis. He is the author/co-author of more than 30 peer-reviewed publications. He served as technical reviewer for various conferences and journal papers.