Dyadic Spatial Resolution Reduction Transcoding for H.264/AVC

Jan De Cock, Stijn Notebaert, Kenneth Vermeirsch, Peter Lambert, and Rik Van de Walle

In: Multimedia Systems, 16 (2), pp. 139—149, March 2010

Link: http://www.springerlink.com/content/u26h5xj051773731/

Noname manuscript No.
(will be inserted by the editor)

Dyadic Spatial Resolution Reduction Transcoding for H.264/AVC

Jan De Cock · Stijn Notebaert · Kenneth Vermeirsch · Peter Lambert · Rik Van de Walle

the date of receipt and acceptance should be inserted later

Abstract In this paper, we examine spatial resolution downscaling transcoding for H.264/AVC video coding. A number of advanced coding tools limit the applicability of techniques which were developed for previous video coding standards. We present a spatial resolution reduction transcoding architecture for H.264/AVC, which extends open-loop transcoding with a low-complexity compensation technique in the reduced-resolution domain. The proposed architecture tackles the problems in H.264/AVC and avoids visual artifacts in the transcoded sequence, while keeping complexity significantly lower than more traditional cascaded decoder-encoder architectures. The refinement step of the proposed architecture can be used to further improve rate-distortion performance, at the cost of additional complexity. In this way, a dynamic-complexity transcoder is rendered possible. We present a thorough investigation of the problems related to motion and residual data mapping, leading to a transcoding solution resulting in fully compliant reduced-size H.264/AVC bitstreams.

Keywords Video adaptation · Transcoding · Spatial resolution reduction · H.264/AVC

1 Introduction

In heterogeneous multimedia environments, it is beneficial to adjust the resolution of video streams to the display capabilities of the receiving devices. While content is created often in a single high-resolution format, the video streams are displayed on a large number of devices with widely varying characteristics such as display resolution, processing power, or battery life. Reducing the spatial resolution of the video stream can tailor the video to the needs and capabilities of these devices. Additionally, a reduction of spatial resolution induces a reduction of the bit rate of the video stream, hereby limiting network bandwidth or storage requirements.

While pixel-based resolution reduction can be used, i.e., decoding, pixel-domain downscaling, and subsequently re-encoding, other solutions are desired which limit the computational complexity. Transcoding is used to speed up the conversion process, by efficiently reusing information contained in the original bitstream, such as motion vectors, prediction modes, and residual data. Efficient techniques have been examined for previous video coding standards, such as MPEG-1 or MPEG-2 Video. In this paper, we discuss efficient spatial resolution reduction transcoding for H.264/AVC. When compared to previous video coding standards, a number of issues arise that require closer attention in order to obtain high-quality transcoded video sequences.

In MPEG-2, low-complexity frequency-domain spatial resolution reduction could be accomplished. To achieve this, low-resolution blocks could be synthesized using low-order frequency components of the original blocks. Frequency synthesis, and the resulting drift was, for example, analyzed in [5] for MPEG-2 video coding.

Different reduced spatial resolution transcoding architectures were examined in [17], where the authors examined the problem of transcoding from MPEG-2 to MPEG-4 Visual, with a focus on temporal drift com-

J. De Cock · S. Notebaert · K. Vermeirsch · P. Lambert · R. Van de Walle
Ghent University – IBBT
Department of Electronics and Information Systems – Multimedia Lab
Gaston Crommenlaan 8 b 201, B-9050 Ledeberg-Ghent
Tel.: +3293314957, Fax: +3293314896, E-mail: jan.decock@ugent.be

pensation. Drift due to motion vector misalignment during downscaling has been dicussed for example in [8] for MPEG-2 video transcoding.

More recently, spatial resolution transcoding was examined for H.264/AVC. In [18,9,11], the problem of mode decision was examined, to provide a speed-up in cascaded decoder-encoder architectures (i.e., re-encoding). In previous publications, these cascaded architectures have been the primary focus of spatial resolution transcoding for H.264/AVC. Due to their double-loop nature, however, complexity and buffer requirements remain high when compared to fast open-loop transcoding architectures such as those developed for previous video coding standards [12].

So far, little has been written on reduced-complexity architectures for downscaling in H.264/AVC. A number of improvements of H.264/AVC video coding, such as sub-macroblock partitioning (with partition sizes down to 4×4 pixels) and the use of multiple reference pictures, introduce new challenges when developing fast techniques for H.264/AVC video stream downsizing and prohibit straightforward application of previously existing techniques. In this paper, we highlight and tackle these new problems, with a focus on the case of dyadic downscaling. Open-loop techniques for arbitrary downscaling ratios have been discussed in [6]. As we will see in Sect. 2, however, the applicability of these open-loop techniques is restrained by a number of features of H.264/AVC. Without proper measures, significant artifacts and drift arise in the video stream. After a discussion of the advanced coding tools that cause these limitations, we introduce a fast architecture that tackles the issues, which is based on initial work discussed in [3].

The remainder of this paper is organized as follows. In Sect. 2, we describe mapping-related problems in H.264/AVC. In Sect. 3, we lay out our architecture for spatial resolution reduction transcoding. Sect. 4 and Sect. 5 give more details on the motion and mode data mapping process, and the residual data conversion process, respectively. In Sect. 6, we provide implementation results for our architecture. Finally, conclusions are given in Sect. 7.

2 Issues in H.264/AVC spatial resolution mapping

In this section, we focus on mapping problems related to spatial resolution reduction of motion-compensated pictures in H.264/AVC. Four major issues arise when downscaling H.264/AVC video streams without fully decoding and re-encoding. If not taken care of properly, transcoding would result in artifacts and drift in the output video stream.

These four problems are illustrated in Fig. 1, for a possible mapping; shaded areas correspond to blocks subject to misprediction.

2.1 Sub-macroblock partitions

Firstly, the H.264/AVC tree-structured motion compensation design allows block sizes smaller than 8×8 pixels, i.e., sub-macroblock partitions down to 8×4, 4×8, or 4×4 pixels. For these sub-macroblock partitions, there are no respective counterparts in the reduced-resolution domain. When downsizing H.264/-AVC-coded sequences, this means that a straightforward mapping of macroblock partitions and motion vectors from four macroblocks to one is not possible if sub-macroblock partitions are used in the original stream. This is illustrated in Fig. 1(a), where the bottom-right macroblock in the original resolution contains 4×8 and 4×4 sub-macroblock partitions.

2.2 Multiple reference pictures

A second problem is related to the multiple reference picture motion-compensated prediction design. In H.264/AVC, up to 16 reference pictures can be used. The reference picture lists can contain both short-term and long-term reference pictures. Reference picture indices can vary for every macroblock partition; the same reference picture, however, will be shared by all *sub*-macroblock partitions in the same macroblock partition. This corresponds to a reference picture granularity down to 8×8 pixels. A misprediction occurs after downscaling H.264/AVC streams if different reference pictures were used in the original stream to predict a single macroblock. At most four different reference picture indices need to be mapped to a single reference picture index. In Fig. 1(b), an example is shown where macroblocks two and four (in raster scan order) are predicted based on multiple reference pictures.

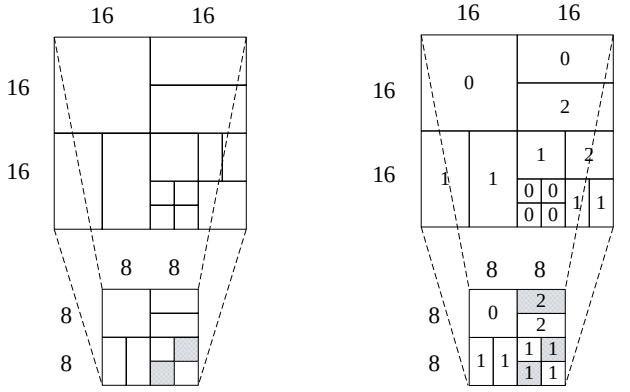
2.3 Variable prediction direction in B pictures

A further complication occurs in B pictures, where in a single macroblock the choice can be made between motion-compensated prediction based on forward prediction (reference pictures in reference picture list 0), backward prediction (reference picture list 1), or bidirectional prediction (prediction from both lists). This selection can vary with the same granularity as for the

reference indices, i.e., a different choice can be made for every 8×8 block of pixels. This is illustrated in Fig. 1(c) for a possible mapping of four macroblocks containing macroblock partitions using forward prediction (list 0, abbreviated as 'L0'), backward prediction (list 1, 'L1'), and bidirectional prediction (both lists, 'Bi').

2.4 Intra-coded macroblocks

Fourthly, the availability of intra-coded macroblocks in P and B pictures requires an update of techniques available in previous video coding standards. Intra-coded macroblocks form their prediction based on the pixels of surrounding, reconstructed blocks, and can be inserted at any location in P and B pictures. When no reconstruction is available of the surrounding (macro)blocks (due to the absence of a pixel-domain reconstruction loop), it is not possible to map these macroblocks to macroblock partitions in the downsized stream.



(a) Sub-macroblock partitions

(b) Multiple reference indices

(c) Variable prediction direction in B pictures

(d) Intra macroblocks in P and B pictures

Fig. 1 Coding tools in H.264/AVC leading to mapping-related problems; areas subject to mapping-related errors are indicated in gray.

From here on, macroblocks that do not suffer from one of these complications will be referred to as 'direct-mappable macroblocks' (DM macroblocks), whereas the 'non-direct-mappable macroblocks' (NDM macroblocks) are restricted by one of the above reasons. The NDM macroblocks require closer attention for spatial resolution reduction transcoding.

3 Transcoding Architecture

3.1 Cascaded decoder-encoder architecture

A cascaded decoder-encoder approach will completely decode the original sequence, downscale the decoded frames in the pixel domain, and subsequently re-encode the downsized frames [12]. This approach can be regarded as a reference for rate-distortion performance, but has low computational efficiency due to highly complex operations at the encoder side. In order to speed up the re-encoding process, the time-consuming motion estimation process can be avoided by using motion mapping, i.e., finding suitable macroblock partitions, reference indices, and motion vectors based on the information available in the original bitstream. A number of mode mapping algorithms have been presented in literature that limit the loss when compared to rate-distortion optimal mode and motion estimation, e.g. in [18,9]. The resulting architecture is shown in Fig. 2. In H.264/AVC, both motion-compensated prediction ('MC') and intra prediction ('IP') can be used to form the most appropriate prediction signal. In the following sections, we investigate further simplifications of this transcoder architecture, leading to a dynamic-complexity spatial resolution reduction transcoder.

3.2 Open-loop architecture for DM macroblocks

If the input video stream contains only DM macroblocks, a straightforward mapping can be used from the incoming macroblock partitions, reference indices, and motion vectors to the output bitstream. Downsizing techniques in the compressed domain as in [6] can be used in this case, based on an open-loop transcoder architecture, as shown in Fig. 3. The downsampling can be performed in the pixel domain as well as in the compressed domain. In the latter case, frequency synthesis techniques can be used, analogous to techniques for MPEG-2 bitstreams [5]. A change is required to reflect the integer transform used for H.264/AVC, as opposed to the DCT which was used for prior video coding standards. As was shown in [6], a conversion in the DCT do-

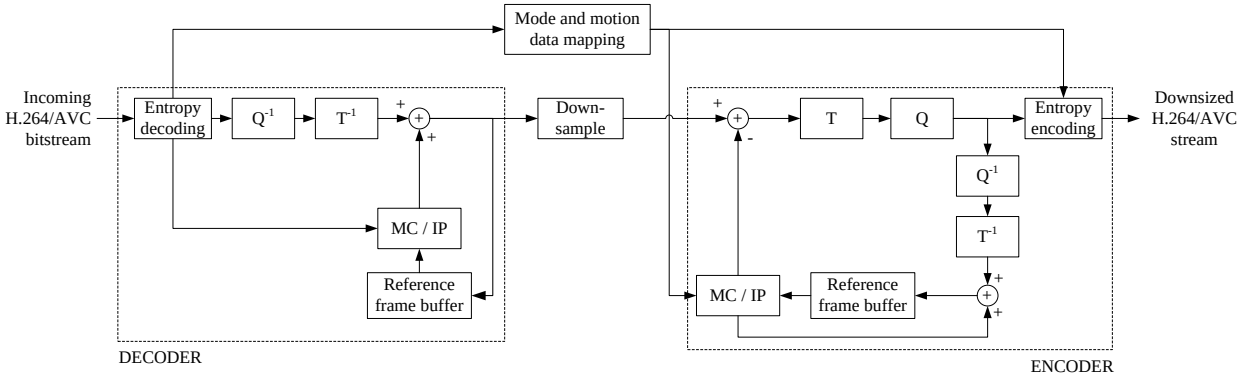


Fig. 2 Cascaded spatial resolution transcoder with mode and motion data mapping.

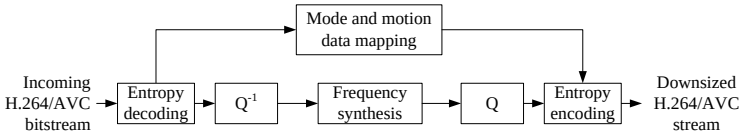


Fig. 3 Open-loop spatial resolution transcoder (frequency synthesis).

main is able to outperform pixel-domain bicubic spline and bilinear filters.

The open-loop architecture can be considered as the most efficient transcoding approach, but has no provisions to update residual data when a change is required to the motion parameters when NDM macroblocks are encountered in the incoming bitstream.

3.3 Proposed architecture for DM and NDM macroblocks

When NDM macroblocks are present, open-loop mapping of transform coefficients results in serious errors due to wrongfully used reference pictures or motion vectors for the downsized stream. For these macroblocks, a correction of the motion compensation reference block is required. For this reason, we introduce the architecture as shown in Fig. 4. In this architecture, correction of the errors (represented by the second motion compensation step) is only required for NDM macroblocks. To allow this correction operation, a *reduced-resolution* reference picture is created. This is represented in Fig. 4 by the first motion compensation loop.

The reduced-resolution pixel-domain reconstruction allows us to solve the shortcomings of transform-domain solutions. In particular, it allows us to correct the residual data blocks in case an update in mode or motion information is required. The signals in bold in Fig. 4 refer to the residual data blocks, before and after the successive operations. The upper-case signals indicate transform-domain blocks, while the lower-case signals correspond to pixel-domain residual data blocks. More detail on the residual data mapping process is given in Sect. 5.

A number of differences with the cascaded decoder-encoder architecture can be identified that reduce computational complexity. Firstly, for DM macroblocks, open-loop downsampling or frequency synthesis can be applied. Note that for these blocks, the first motion-compensation step is also performed, because each block can be used itself as reference for future motion-compensated prediction. Secondly, complexity is reduced by only applying motion compensation at the reduced resolution. In the case of dyadic downscaling, complexity is reduced by a factor four for this operation. Also, this results in reference pictures that need only be stored at the reduced resolution. The use of the high-resolution buffers from the first loop in the cascaded architecture is avoided. Finally, the second motion compensation step is only required for correction of NDM macroblocks. This additional motion compensation operation uses the updated motion vector, or adjusted reference picture, and compensates for the mismatches mentioned in Sect. 2. As will be explained later, this second motion compensation loop can also be used for refinement of DM macroblocks.

Due to the relatively low computational complexity of intra prediction, re-encoding of intra-coded pictures is applied. Research in the context of SNR transcoding has also pointed out that re-encoding of intra-coded pictures results in improved visual results [2]. The complexity of the overall architecture is influenced only to a minor extent. This strategy is also applied for our proposed spatial resolution reduction transcoder. Downsampling of the intra frames is executed according to

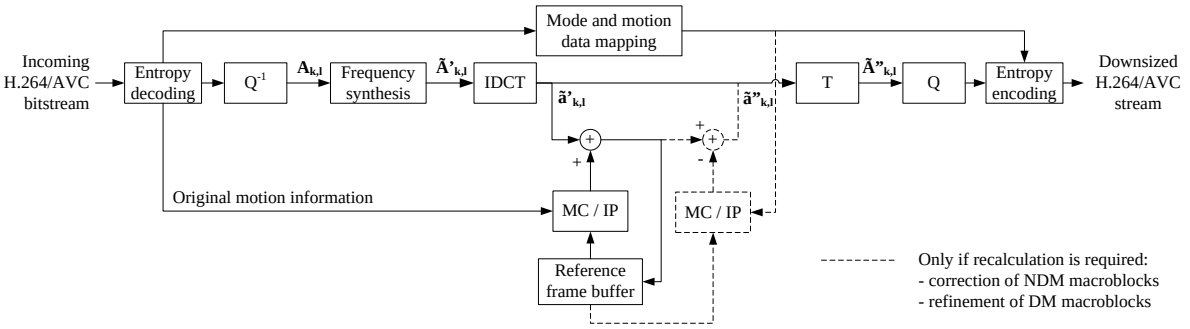


Fig. 4 Proposed spatial resolution transcoder.

the sine-windowed Sinc-function, which is also recommended as downsampling filter for spatial scalability in the scalable extension of the H.264/AVC video coding standard (SVC). This filter can be defined as follows:

$$f(x) = \begin{cases} \frac{\sin(\pi \frac{x}{D})}{\pi \frac{x}{D}} \sin\left(\frac{\pi}{2}(1 + \frac{x}{ND})\right) & |x| < ND \\ 0 & \text{otherwise} \end{cases}$$

with a decimation parameter $D = 2.5$ and $N = 3$ lobes for the Sinc-function on each side. For practical applications and complexity reasons, the filter size is limited to 12 taps [10].

3.4 Dynamic complexity refinement

So far, our reduced-complexity architecture only used the second motion compensation step for correction of NDM macroblocks, while DM macroblocks could be transcoded open-loop. It is, however, possible to use the second motion compensation step to further refine the motion vectors, or in order to select a more appropriate choice of reference picture. The amount in which this functionality is used, can be dynamically varied, resulting in a trade-off between rate-distortion efficiency and computational complexity. From the results in Sect. 6, the impact of this refinement step becomes clear. Rate-distortion performance is improved in a significant way by refining motion vectors and macroblock partitioning. In order to achieve this, techniques can be used as in Sect. 4.

4 Mode and motion data mapping

4.1 Motion mapping for DM macroblocks

For DM macroblocks, the original partitions can be mapped to their respective downscaled counterparts. In this way, a single macroblock is mapped to an 8×8 partition, and incoming macroblock partitions are mapped to 8×8, 8×4, 4×8, or 4×4 sub-macroblock partitions. This results in the use of 8×8 partitions for every downscaled macroblock. The motion vectors are mapped accordingly, while the reference indices and prediction directions remain identical. If desired, the motion parameters that are obtained in this way can be further refined, as will be discussed in Sect. 4.3.

4.2 Motion mapping for NDM macroblocks

Particular care needs to be taken for the case of NDM macroblocks. For NDM macroblocks, the appropriate output motion parameters need to be determined, i.e., motion vectors, reference indices, and prediction directions. In a first step, an initial mapping is performed, leading to an H.264/AVC-compliant bitstream. As is the case for DM macroblocks, further refinement can be applied if desired.

4.2.1 Motion vector mapping

When sub-macroblock partitions were used in the original bitstream, no downscaled counterpart exists for these partitions. A new motion vector needs to be derived, based on the motion information from the corresponding (up to four) incoming motion vectors. After derivation of the new output motion vector, correction of the corresponding residual values is required using the second motion compensation loop in our proposed architecture. The pixels for which correction is required depends on the choice of motion vector for the sub-macroblock partition. We treat the motion vectors from the incoming bitstream as candidate output motion vectors, and evaluate them in a rate-distortion optimized way, as will be further explained in Sect. 4.4.

4.2.2 Reference index and prediction direction mapping

An additional problem in H.264/AVC downscaling occurs when multiple reference indices were used to code

a single macroblock. In this case, a selection needs to be made of the best-suited reference index for the composed macroblock partition. In B pictures, this is further complicated when multiple prediction directions were used within macroblocks, i.e., a combination of forward, backward, and/or bidirectional prediction.

Selection of the appropriate motion parameters is performed similar to the motion vector selection, by rate-distortion optimized evaluation of the candidate reference indices and prediction directions.

4.2.3 Intra macroblocks in P and B pictures

Since intra-coded macroblocks are allowed in P and B pictures, provisions need to be made for spatial resolution reduction of typical H.264/AVC video streams. Intra macroblocks contribute to the rate-distortion performance improvement over previous video coding standards, and are often inserted in P and B pictures, e.g., when little correlation is found when compared to the available reference pictures. Since the low-resolution reconstructed pictures are available in the proposed transcoder, it becomes possible to convert the intra macroblocks to intra or inter macroblocks in the downsized stream. Although transform-domain intra prediction has been mentioned as an alternative approach to removing intra macroblocks from P and B pictures, non-linear operations result in highly degraded quality and highly complex computations, requiring an extensive amount of floating-point multiplications [1]. In our approach, we benefit from the availability of pixel-domain reconstructed pictures (in the reduced-size domain) to convert intra macroblocks.

Given the high amount of dependencies in intra macroblocks, particular care has to be taken to avoid spatial drift. Since we have the reduced-size reconstructed picture at our disposal, it is possible to reconstruct intra-coded macroblocks. For optimum quality results, we perform the intra prediction in the *original* resolution. In a first step, we upscale the surrounding prediction pixels that are used to form the 4×4 or 16×16 prediction, using the 4-tap separable filter used for upscaling in SVC [7]. After performing standard H.264/AVC intra prediction, the result is downscaled as is done for intra-coded pictures. Given the relatively low complexity of intra prediction (in particular when compared to motion-compensated prediction), the impact on the overall complexity is kept low using this approach.

When intra macroblocks are encountered, a proper macroblock conversion needs to be performed. When the four input macroblocks are intra-coded, the corresponding output macroblock is likely to be intra-coded also. Otherwise, a *mixed* block is found. In this case, we investigate the possibility of using an intra-coded macroblock in the output stream. As an alternative, MCP is evaluated using motion vectors derived in the same way as described above for the case of updated reference indices.

4.3 Motion refinement

Once an H.264/AVC-compliant bitstream is obtained by mapping DM and NDM macroblocks, refinement of the motion parameters can be applied, depending on the requirements of the scenario in which the transcoder is used. Further refinement somewhat increases complexity, yet improves rate-distortion performance. In our architecture, we evaluated motion refinement by using a bottom-up limitation mode decision process, as is described for example in [9]. Overall complexity is determined by the amount of macroblocks to which this process is applied, and by the amount of refinement steps that are applied to each macroblock. Note that an increase in number of refined macroblocks also induces a refinement of residual data for these macroblocks, i.e., the activation of the second motion compensation loop.

4.4 Rate-distortion optimized mode selection

An important advantage of the proposed architecture is that pixel-domain (reduced-size) reference pictures are created which can be used as a reference for future prediction. These pictures allow the architecture to correct artifacts that would otherwise arise in NDM macroblocks. The availability of these reference pictures also enables us to calculate the displacement difference for every set of motion parameters, and perform rate-distortion optimized motion selection. Given the evaluated partition type, motion vector, and reference index, the output rate and distortion can be determined. Using this calculation, a choice can be made based on Lagrangian minimization:

$$\arg\min\{D + \lambda \cdot R\} \ .$$

Here, distortion D is expressed as the SAD of the displacement difference in the second motion compensation loop, and R is calculated as the rate cost of the motion information. λ is chosen according to the relationship that was determined empirically in [15], i.e., $\lambda = 0.85 \cdot Q^2$ (this leads to the same λ factors as used in the Joint Model H.264/AVC reference encoder software).

5 Residual data mapping

5.1 Residual data mapping

Several approaches have been proposed in literature for downconversion of texture information. In [13,14], frequency synthesis was used for downconversion. In frequency synthesis, four 4×4 blocks of a macroblock are subject to a global transformation. In this way, a single frequency domain block is realized using information in the entire macroblock. Using frequency synthesis, more of the block energy is captured, and the frequency content of the block is represented more accurately. From the synthesized block, the low-order frequency components are cut out.

A similar approach for arbitrary resizing was discussed in [6], which allows the target DCT frame to be constructed as a whole from the 4×4 integer transform blocks in the original frame. The resizing operation was represented as a multiplication using fixed matrices. We obtain the residual values by converting the H.264/AVC integer transform blocks to DCT blocks, subsequently discarding high frequency DCT coefficients, and applying an inverse DCT to the resulting blocks.

This is expressed as follows. At first, an inverse integer transform is applied to the 4×4 blocks $A_{k,l}$:

$$a_{k,l} = C_i^T \times A_{k,l} \times C_i \ ,$$

where C_i represents the inverse H.264/AVC integer transform [4]. A traditional forward 8×8 DCT transform is applied to a group of four 4×4 blocks

$$A'_{k,l} = D_8 \times \begin{bmatrix} a_{2k,2l} & a_{2k,2l+1} \\ a_{2k+1,2l} & a_{2k+1,2l+1} \end{bmatrix} \times D_8^T \ .$$

From this 8×8 block, the high-frequency components are discarded, resulting in the 4×4 DCT blocks

$$\tilde{A}'_{k,l} = L_{4\times8} \times A'_{k,l} \times L_{4\times8}^T \ .$$

with $L_{4\times8} = \begin{bmatrix} I_4 \ 0 \end{bmatrix}$. The result is inverse DCT transformed:

$$\tilde{a}'_{k,l} = D_4^T \times \tilde{A}'_{k,l} \times D_4 \ .$$

These resulting values are used in the reduced-resolution reconstruction loop, as is shown in Fig. 4. We use this approach given its improved performance over spatial filters and its computational efficiency [6]. By combining the successive steps in fixed multiplication matrices, computational complexity remains limited. The output 4×4 matrices $\tilde{a}'_{k,l}$ are used to construct the reduced-size reference pictures in the first motion compensation loop.

5.2 Residual data refinement

The approach using matrix multiplication results in the matrices $\tilde{a}'_{k,l}$ which can be used for reconstruction in the first motion compensation loop. Although this technique, as introduced in [6], works well for residual data of DM macroblocks, downsizing of motion parameters is not taken into account. Further adjustments are required in case NDM macroblocks are present, as is the case for typical H.264/AVC bitstreams, or if motion data refinement is applied. The use of sub-macroblock partitions, multiple reference indices, or intra macroblocks in the original stream will lead to significant artifacts in the transcoded video stream. If a valid H.264/AVC bitstream needs to be output, changes in the motion information are required, which in turn necessitates an update of the residual data.

In this case, the approach as discussed in Sect. 4 implies a refinement of the residual data using the second motion compensation loop in our proposed architecture. In order to obtain the output residual data, the motion compensation loop uses the newly derived prediction direction, reference index (or indices) and motion vector(s), leading to residual data blocks $\tilde{a}''_{k,l} \neq \tilde{a}'_{k,l}$.

6 Implementation results and discussion

The architecture as described in the previous section was implemented in software, and tested using sequences with varying statistics, namely Akiyo, Paris, and Foreman (original: CIF resolution, 25 fps), and Harbour and Soccer (original: 4CIF resolution, 25 fps). The sequences were encoded using the Joint Model (JM) reference software, version 13.0. All sequences were encoded with an IBBP GOP structure and using CABAC entropy coding, with rate-distortion optimization enabled. We used quantization parameter (QP) values of 22, 27, 32, and 37. The re-encode curves were obtained by successively decoding, pixel-domain downscaling, and re-encoding. For downscaling, the SVC reference software down converter tool was used, which makes use of the 12-tap sine-windowed Sinc-function. The downscaled sequences were re-encoded using the same encoder options in the reduced resolution (JM 13.0). In the tests, a GOP length of 12 frames was used, i.e., an intra-coded picture is inserted every 480 ms at 25 Hz. This GOP length is typically used for broadcast and entertainment-type applications, hereby allowing fast random access [16]. In the remainder of this section, we show the bit rates of the original sequences in the caption of the rate-distortion plots, corresponding to QP values {22, 27, 32, 37}.

6.1 Results for sequences containing DM macroblocks only

Firstly, it is worthwhile to look at results for sequences containing only DM macroblocks, i.e., when no sub-macroblock partitions, multiple reference pictures, or intra macroblocks are used in the original bitstreams. Fig. 5 (Foreman) shows that in this case, the open-loop solution with frequency synthesis performs reasonably well, leading to rate-distortion losses limited to less than 2 dB when compared to full decoding and re-encoding, with significant computational complexity savings. Similar results are obtained for Paris (Fig. 6). Here, however, the gap increases towards higher bit rates.

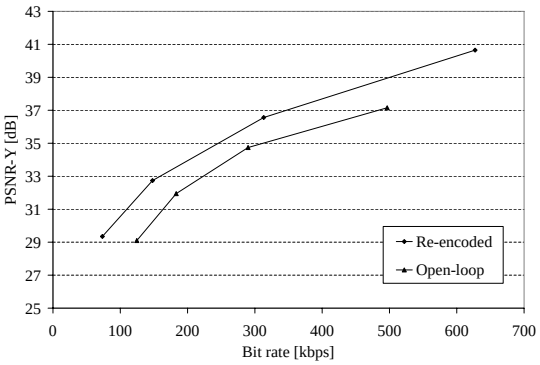


Fig. 5 Rate-distortion performance for sequences containing only DM macroblocks (Foreman, CIF to QCIF). Corresponding rate points in original CIF resolution: {1908, 833, 358, 174} kbps.

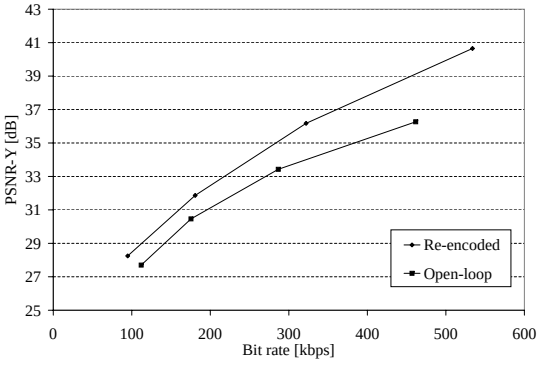


Fig. 6 Rate-distortion performance for sequences containing only DM macroblocks (Paris, CIF to QCIF). Corresponding rate points in original CIF resolution: {2032, 1218, 699, 376} kbps.

6.2 Results for sequences containing NDM macroblocks

6.2.1 Percentage of NDM macroblocks

To illustrate the importance of NDM macroblocks, the percentage of NDM macroblocks per sequence is shown in Table 1. The presence of NDM macroblocks depends not only on the used quantization parameter, but also on the picture type (P or B[1]). In general, the percentage of NDM macroblocks decreases for higher QP values. This could be expected, since a finer partitioning (hence, more submacroblock partitions) is used for higher-quality video streams (lower QP values). For B pictures, larger macroblock types are typically selected, given the more accurate prediction and lower temporal distance to the reference pictures.

6.2.2 Rate-distortion results

In Fig. 7, rate-distortion results are shown for the Paris sequence containing NDM macroblocks in the input sequence. From these curves, it is clear that open-loop processing of residual data results in highly distorted results, and unacceptable quality loss. The presented transcoding architecture with correction of NDM macroblocks (indicated as 'no refinement') is able to significantly improve the R-D results. By additionally refining DM macroblocks, the output quality can further be improved. By refining all macroblocks, the curve indicated as 'with refinement' is achieved. Hence, this indicates the best achievable rate-distortion performance. For the generated results, the refinement step resulted in R-D points with slightly lower PSNR values; bit rate, however, decreased significantly after refinement. After the operation, results are able to approach the decoder-encoder cascade within 1-2 dB.

Results for the Akiyo sequence under the same conditions (i.e., containing NDM macroblocks) are shown in Fig. 8. Here, the loss of open-loop transcoding is less significant, due to the reduced amount of motion in the sequence. Quality is improved only to a minor extent due to correction of NDM macroblocks. Here, due to the low motion content, only a small number of macroblocks uses sub-macroblock partitioning, multiple reference indices, or intra macroblocks. In fact, more than 90% of the macroblocks are DM macroblocks for this sequence, even for low bit rates. This limits the achievable R-D gain of NDM macroblock correction. Refining DM macroblocks, however, can further improve the quality of the output bitstream by more than 1 dB.

[1] Note that I frames contain only NDM (intra-coded) macroblocks, and are re-encoded to maximize quality.

Table 1 Amount of NDM macroblocks [%].

	Akiyo (CIF) QP				**Foreman (CIF)** QP			
	22	27	32	37	22	27	32	37
P frames	7.7	6.2	3.6	2.0	40.4	26.4	16.0	8.2
B frames	0.6	0.4	0.2	0.2	10.5	7.7	4.4	2.8

	Paris (CIF) QP			
	22	27	32	37
P frames	24.6	21.2	17.8	13.9
B frames	9.8	8.3	6.0	4.4

	Soccer (4CIF) QP				**Harbour (4CIF)** QP			
	22	27	32	37	22	27	32	37
P frames	26.6	18.8	13.1	6.6	36.8	34.2	28.3	18.1
B frames	5.7	5.6	4.2	2.7	4.2	5.3	5.7	4.7

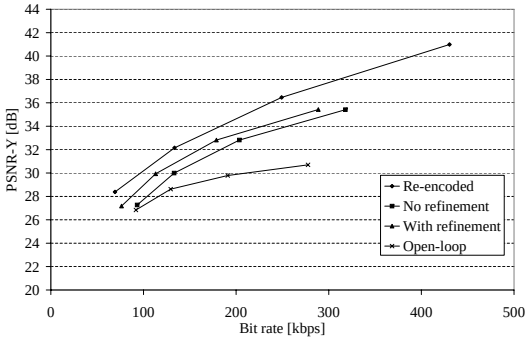


Fig. 7 Rate-distortion performance for sequences containing NDM macroblocks (Paris, CIF to QCIF). Corresponding rate points in original CIF resolution: {1668, 962, 526, 272} kbps.

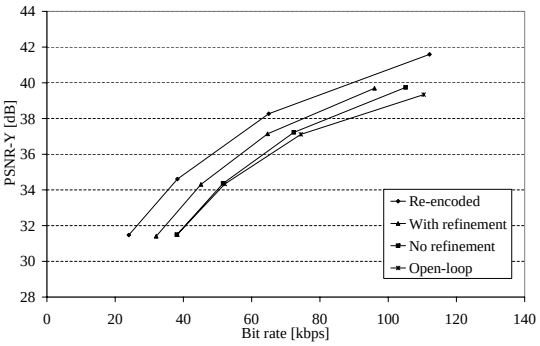


Fig. 8 Rate-distortion performance for sequences containing NDM macroblocks (Akiyo, CIF to QCIF). Corresponding rate points in original CIF resolution: {379, 195, 103, 62} kbps.

Results for the Soccer and Harbour sequences, transcoded from 4CIF to CIF, are shown in Fig. 9 and Fig. 10. From these results, it can be seen that our algorithm results in larger reductions of the bit rate than re-encoding for the same QP values. When compared to re-encoding, more high-frequency information is removed from the bitstream. Here also, the loss in rate-distortion performance remains limited to 1-2 dB, and significant gains are obtained when compared to open-loop transcoding. This is particularly so for high-motion sequences, such as Soccer (Fig. 9). For the Harbour sequence, the loss when compared to re-encoding is limited to 0.5-1 dB in the lower bit rate range, which is the typical use range in transcoding scenarios.

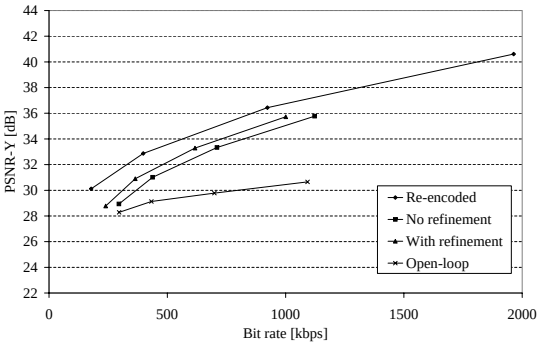


Fig. 9 Rate-distortion performance for sequences containing NDM macroblocks (Soccer, 4CIF to CIF). Corresponding rate points in original 4CIF resolution: {8170, 3724, 1514, 681} kbps.

6.2.3 Visual results

Visual results after transcoding are shown in Fig. 11. The screenshots shown on the left hand side demonstrate the visual artifacts introduced by open-loop transcoding, i.e., without appropriate correction of NDM macroblocks. The screenshots on the right hand side demonstrate the results by using the proposed architecture (before refinement). In Fig. 11(a), many artifacts can be seen around the players due to incorrectly

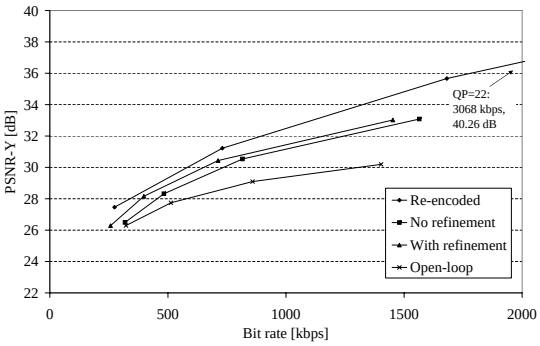


Fig. 10 Rate-distortion performance for sequences containing NDM macroblocks (Harbour, 4CIF to CIF). Corresponding rate points in original 4CIF resolution: {12299, 6258, 2760, 1142} kbps.

predicted blocks after merging. By using the proposed architecture, these artifacts are removed, as shown in Fig. 11(b). For the *Foreman* sequence in Fig. 11(c), artifacts are present around the hat and in the face of the foreman. Visual quality is highly improved after correction using the presented transcoder (see Fig. 11(d)).

6.3 Complexity results

Computational complexity increases when mode and motion vector refinement is used, given that more macroblocks will be using the second motion compensation step. In this way, a trade-off is made between computational complexity and bit rate reduction. As an indication of complexity, timing results are given in Table 2, for re-encoding, open-loop transcoding, and transcoding without and with refinement. The results are shown relative to the time needed for re-encoding. Note that the Joint Model reference software was used for re-encoding. Although the JM software is non-optimized, these results give an indication of the timing savings that can be achieved using the proposed architecture. Both the JM software and the used transcoding software can be further optimized.

From Table 2, we see that significant computational complexity gains are made when compared to re-encoding. When open-loop transcoding is applicable, a reduction of more than 97% is obtained. Depending on the amount of refinement applied to the downscaled bitstream, transcoding achieves timing gains of 88% to 96% relative to re-encoding. This amount can be varied dynamically, depending on the available computational resources in the transcoding system.

7 Conclusions

In this paper, we highlighted a number of problems as found in H.264/AVC spatial resolution reduction transcoding. A number of restrictions inhibit the use of straightforward, open-loop, residual data synthesis techniques. We provided a low-complexity transcoder architecture, which is able to handle both direct-mappable as well as non-direct-mappable macroblocks. Complexity is reduced by performing motion compensation only in the reduced-resolution domain. The refinement step of the proposed architecture can be used to further improve rate-distortion performance, at the cost of additional complexity. In this way, a dynamic-complexity transcoder is made possible. Complexity, however, remains many times lower than that of re-encoding, with reductions of 88 to 96% depending on the amount of refinement applied.

Acknowledgements The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Flanders), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

1. C. Chen, P.-H. Wu, and H. Chen. Transform-domain intra prediction for H.264. In *Proceedings of the 2005 IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.
2. J. De Cock, S. Notebaert, and R. Van de Walle. A novel hybrid requantization transcoding scheme for H.264/AVC. In *Proc. Int. Symp. Signal Process. Appl. (ISSPA)*, February 2007.
3. Jan De Cock, Stijn Notebaert, Kenneth Vermeirsch, Peter Lambert, and Rik Van de Walle. Efficient spatial resolution reduction transcoding for H.264/AVC. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, October 2008.
4. Henrique Malvar, Antti Hallapuro, Marta Karczewicz, and Louis Kerofsky. Low-complexity transform and quantization in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):598–603, July 2003.
5. R. Mokry and D. Anastassiou. Minimal error drift in frequency scalability for motion-compensated DCT coding. *IEEE Trans. Circuits Syst. Video Technol.*, 4(4):392–406, August 1994.
6. V. Patil and R. Kumar. A fast arbitrary factor H.264/AVC video re-sizing algorithm. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, September 2007.
7. A. C. Segall and G. J. Sullivan. Spatial scalability within the H.264/AVC scalable video coding extension. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9):1121–1135, September 2007.
8. Bo Shen. Submacroblock motion compensation for fast down-scale transcoding of compressed video. *IEEE Trans. Circuits Syst. Video Technol.*, 15(10):1291–1302, October 2005.

(a) Open-loop transcoding (detail of *Soccer*, QP 22, 4CIF to CIF, 9th frame).

(b) Transcoding without refinement (detail of *Soccer*, QP 22, 4CIF to CIF, 9th frame).

(c) Open-loop transcoding (detail of *Foreman*, QP 22, CIF to QCIF, 15th frame).

(d) Transcoding without refinement (detail of *Foreman*, QP 22, CIF to QCIF, 15th frame).

Fig. 11 Visual results for open-loop and proposed architecture (without refinement) for *Soccer* and *Foreman* sequences.

9. H. Shen, X. Sun, F. Wu, H. Li, and S. Li. A fast downsizing video transcoder for H.264/AVC with rate-distortion optimal mode decision. In *Proc. IEEE Int. Conf. Multimedia & Expo (ICME)*, pages 2017–2020, July 2006.
10. S. Sun and J. Reichel. *AHG Report on Spatial Scalability Resampling*. Joint Video Team, Doc. JVT-R006, Bangkok, Thailand, January 2006.
11. Yap-Peng Tan and Haiwei Sun. Fast motion re-estimation for arbitrary downsizing video transcoding using H.264/AVC standard. *IEEE Trans. Consum. Electron.*, 50(3):887–894, August 2004.
12. A. Vetro, C. Christopoulos, and H. Sun. Video transcoding architectures and techniques: an overview. *IEEE Signal Processing Magazine*, pages 18–29, March 2003.
13. A. Vetro, H. Sun, P. DaGraca, and T. Poon. Minimum drift architectures for 3-layer scalable DTV decoding. *IEEE Trans. Consum. Electron.*, 44(3):527–536, August 1998.
14. Anthony Vetro and Huifang Sun. Frequency domain down-conversion of HDTV using an optimal motion compensation scheme. *Journal of Imaging Science and Technology*, 1998.
15. Thomas Wiegand and Bernd Girod. Lagrange multiplier selection in hybrid video coder control. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, September 2001.
16. Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini, and Gary J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):688–703, July 2003.

Table 2 Timing results relative to re-encoding [%].

	Akiyo (CIF to QCIF) QP				**Foreman (CIF to QCIF)** QP			
	22	27	32	37	22	27	32	37
Open-loop	2.6	2.5	2.3	2.3	2.6	2.5	2.4	2.4
No refinement	4.4	4.2	3.9	3.8	4.9	4.7	4.6	4.1
With refinement	10.5	10.0	9.4	8.8	11.2	10.9	10.6	9.3

	Paris (CIF to QCIF) QP			
	22	27	32	37
Open-loop	2.7	2.6	2.5	2.4
No refinement	5.0	4.7	4.6	4.2
With refinement	12.0	10.8	10.7	9.6

	Soccer (4CIF to CIF) QP				**Harbour (4CIF to CIF)** QP			
	22	27	32	37	22	27	32	37
Open-loop	2.7	2.4	2.2	2.2	2.8	2.6	2.5	2.5
No refinement	4.9	4.6	4.2	4.1	5.3	5.1	5.0	4.8
With refinement	11.5	10.7	9.7	9.3	11.9	11.6	11.1	11.0

17. P. Yin, A. Vetro, B. Liu, and H. Sun. Drift compensation for reduced spatial resolution transcoding. *IEEE Trans. Circuits Syst. Video Technol.*, 12(1):1009–1020, November 2002.
18. P. Zhang, Y. Lu, Q. Huang, and W. Gao. Mode mapping method for H.264/AVC spatial downscaling transcoding. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, October 2004.