Corrales-García *et al. EURASIP Journal on Advances in Signal Processing* 2012, **2012**:135
http://asp.eurasipjournals.com/content/2012/1/135

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Wyner–Ziv to Baseline H.264 Video Transcoder

Alberto Corrales-García[1*], Jose Luis Martínez[1], Gerardo Fernández-Escribano[1], Jose Miguel Villalon[1], Hari Kalva[2] and Pedro Cuenca[1]

Abstract

Mobile-to-mobile video communications is one of the most requested services which operator networks can offer. However, in a framework where one mobile device sends video information to another, both transmitter and receptor should employ video encoders and decoders with low complexity. On the one hand, traditional video codecs, such as H.264, are based on architectures which have encoders with higher complexity than decoders. On the other hand, Wyner–Ziv (WZ) video coding (a particular case of distributed video coding) is an innovative paradigm, which provides encoders with less complexity than decoders. Taking advantage of both paradigms, in terms of low complexity algorithms, a suitable solution consists in transcoding from WZ to H.264. Nevertheless, the transcoding process should be carried out in an efficient way so as to avoid major delays in communication; in other words, the transcoding process should perform the conversion without requiring the complete process of decoding and re-encoding. Based on all the algorithms and techniques we have proposed before, a low complexity WZ to H.264 Transcoder for the Baseline Profile is proposed in this article. Firstly, the proposed transcoder can efficiently turn every WZ group of pictures into the common H.264 I11P pattern and, secondly, the proposed transcoder is based on the hypothesis that macroblock coding mode decisions in H.264 video have a high correlation with the distribution of the side information residual in WZ video. The proposed algorithm selects one sub-set of the several coding modes in H.264. Moreover, a dynamic motion estimation technique is proposed in this article for use in combination with the above algorithm. Simulation results show that the proposed transcoder reduces the inter prediction complexity in H.264 by up to 93%, while maintaining coding efficiency.

Keywords: Distributed video coding, Wyner–Ziv, H.264, Advanced video coding, Machine learning, Motion vectors, Transcoding

Introduction

Nowadays, the newest networks for mobile devices (such as 4 G) are supporting more and more services for users. Multimedia communications between these mobile devices are becoming an important area of interest in telecommunications. In particular, mobile-to-mobile video teleconferencing is one of the most requested services that these networks can support. However, in a framework where one mobile device sends video information to another, both transmitter and receptor should employ video encoders and decoders of low complexity. On the one hand, traditional video codecs, such as H.264 advanced video coding (AVC) [1] are based on architectures which have encoders of higher complexity than decoders. On the other hand, distributed video coding [2] is an innovative paradigm which does not exploit the temporal correlation on the encoder side. In this way, it provides encoders of less complexity than decoders. Therefore, the requirements for low complexity on both encoder and decoder sides have not been met by using traditional video codecs. Low-cost video communications employing traditional video codecs lead to an inefficient configuration because the encoders sacrifice rate–distortion (RD) performance in order to reduce the encoding complexity by using only lower complexity encoding tools. To support low-cost video coding applications, H.264 defines the baseline profile which defines some restriction tools in order to achieve low complexity constraints which are desirable for videoconferencing and mobile applications. The B frames treatment is beyond the scope of this article since we focus on the I11P (which means one I frame followed by

* Correspondence: albertocorrales@dsi.uclm.es
[1]Instituto de Investigación en Informática de Albacete (I3A), University of Castilla-La Mancha, Campus Universitario, Albacete 02071, Spain
Full list of author information is available at the end of the article

11 P frames) H.264 pattern, which is the most suitable Group of Pictures (GOP) pattern for low-cost communications where small buffer is needed in the devices and the encoding complexity is kept to a minimum. Wyner–Ziv (WZ) [2] video coding is a particular case of DVC. WZ video coding departs from the WZ theorem [3] on source coding with Side Information (SI) available only at the decoder, which is known to be a technique used to reduce the processing complexity of the encoder, leading to a low-cost implementation, while the majority of the computations are taken over by the decoder. Taking advantage of both paradigms, in terms of low complexity algorithms, a suitable solution could consist in transcoding from WZ to H.264 as Figure 1 depicts. In the scenario depicted in Figure 1, the end-user devices will employ the lowest complexity algorithm of traditional and WZ video coding while the majority of computations will be taken over by the transcoder.

Nevertheless, the transcoding process should be carried out in an efficient way so as to avoid major delays in communication. Generally, transcoding can be improved by using information gathered from the first stage (WZ decoding algorithm) to accelerate the second one (H.264 encoding algorithm). In addition, video communications could be established over networks with different bandwidth requirements. Therefore, flexibility is a desired feature in the transcoder, in order to control the bitrate generated and encoder complexity constraints. This could be achieved by using different WZ GOP formats which generate low bitrates and/or complexity (the fact is that WZ encoding algorithm complexity decreases as that of GOP increases [4]), or by controlling the bitrate generated in the output of the transcoder by changing the Quantization Parameter (QP) (in the second half of the transcoder).

The fact is that between WZ and H.264 there are many differences, such as type of frames (I, P frames as opposed to Key and WZ frames), GOP patterns (IPPP as opposed to K-WZ-K), GOP sizes (2, 4, 8 as opposed to 12), that need to be resolved in the transcoder. This article proposes an improved transcoder, with respect to the reference cascade one, which not only efficiently converts the bitstream, but also reduces the time to perform this task. The proposed transcoder includes two different mechanics which can operate together and which: (1) reduce the MacroBlock (MB) mode decision and; (2) reduce the Motion Estimation (ME) process. The first algorithm reuses part of the information collected in the SI generation process (this algorithm can be seen as the ME performed in the WZ decoding algorithm) to reduce the overall MB partition checked into a sub-set of them; the algorithm is based on a Data Mining (DM) process which generates a decision tree from this statistical information. The second algorithm presented in this article reuses the Motion Vectors (MVs) generated in the SI generation process to dynamically reduce the search window in H.264. This is because the MVs generated in the WZ decoding algorithm are correlated with those that will be generated in the H.264 encoding one in such a way that they can be used to focalize the search. The results show a transcoder that performs this process efficiently, with a time saving of up to 93% with a negligible RD penalty drop.

The rest of the article is organized as follows: Section 2 briefly reviews the principles of operation of the WZ video coding as well as the H.264 video coding standard; in Section 3, we make an in-depth study of the most relevant proposals in the framework of WZ-based video transcoders; Section 4 introduces our approaches for inter-frame prediction, based on machine learning (ML) techniques and dynamic search range areas for MV refinement, specifically designed for WZ to H.264 transcoders; in Section 5, we carry out a strong performance evaluation of the proposed algorithm in terms of its computational complexity RD results, which will be introduced here; finally, Section 6 draws our conclusions and outlines our future research plans.

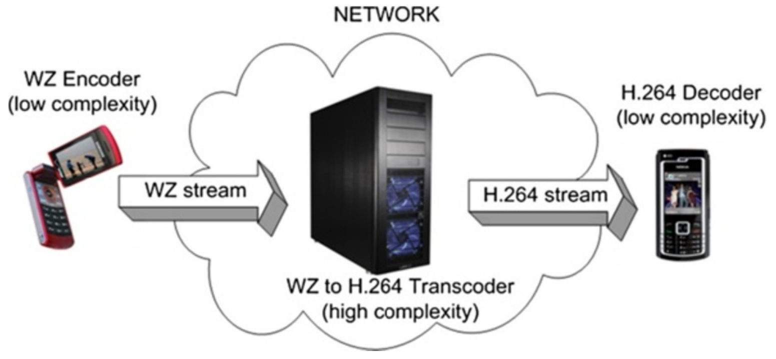


Figure 1 Scheme of DVC to H.264 transcoding framework to support low-cost mobile-to-mobile video communications.

Methods

WZ video codec architectures

Most DVC research is carried out on the architecture first proposed by Stanford [2]; this architecture is mainly characterized by turbo codes as Slepian-Wolf coding and a feedback channel to perform rate control at the decoder. Departing from this architecture, many improvements have been introduced in the literature [5-8] but, recently, a WZ video coding architecture was established by the VISNET-II project [9,10]. The VISNET-II WZ video codec [10] is the most referenced and advanced codec available in the literature, and is based on the architecture depicted in Figure 2.

In a nutshell, the video sequence is divided into Key (*K*) frames and WZ frames in the splitting module (1). At the encoder, the *K* frames are encoded using Intra H.264 video codec [1] (2). WZ frames will follow the WZ encoding algorithm which, firstly, transforms pixel values into coefficients by means of the integer Discrete Cosine Transform (DCT) (3a). In addition, coefficients of WZ frames (already in DCT domain) are organized into bands, and then quantized (3b). Over the resulting quantized symbol stream, bitplane extraction is performed per band (3c). A new bitstream is created per band which contains all the band bitplanes ordered bitplane by bitplane, and then this is independently channel encoded, starting with the stream which contains the DC band bitplanes (3d). The parity bits produced by the source encoder (turbo encoder) are stored in the buffer and transmitted in small chunks upon decoder request via the feedback channel; the systematic bits are discarded (3f).

On the decoder side, the *K* frames are firstly decoded using Intra H.264 video codec [1] (4) and theses frames are used in the SI generation process (5). The frame interpolation module is used to generate the SI frame, an estimate of the WZ frame, based on previously decoded frames [4]. These SI pixel values are also transformed into coefficients and then are used as soft values for the information bits, taking into account the statistical modelling of the virtual noise (7b). The SI can be seen as a corrupted version of the original information, the difference between the original WZ frame and its corresponding SI is considered as correlation noise in a virtual channel. A Laplacian model is used to obtain an approximation of the residual distribution (6) [7]. The SI is used by an iterative decoding algorithm to obtain the decoded quantized symbol. The parity bits and statistical modelling of the correlation noise are the input of the channel decoders. The success of the channel decoding algorithm is determined by module 8b; if the decoding algorithm does not converge, more parity bits are requested using the feedback channel. This iterative procedure is repeated until successful decoding is obtained (normally without errors) and another band starts being decoded. After that, the quantized reconstructed coefficients are obtained using the correlation noise model estimated in (6) and the quantized SI coefficients (8c). After that, module (8d) inverts the transform coefficients to recover pixel values.

H.264 inter prediction

The main purpose of H.264 is to offer a good quality standard able to considerably reduce the output bit rate of the encoded sequences, compared with previous standards, while exhibiting a substantially improved definition of quality and image. H.264 promises a significant advance compared with the commercial standards currently most in use (MPEG-2 [11] and MPEG-4 [12]). For this reason H.264 contains a large amount of compression techniques and innovations compared with the previous standards; it allows more compressed video sequences to be obtained and provides greater flexibility for implementing the encoder. Although H.264 can achieve higher coding efficiency than any other previous coding standard, the computation complexity also increases significantly.

Inter prediction is the most time-consuming task in the H.264 encoder. It is a process which removes the temporal redundancy between images, comparing the current one with previous or later images in terms of time (reference images), looking for a pattern that indicates how the movement is produced inside the sequence. Therefore, this is the most suitable part of the

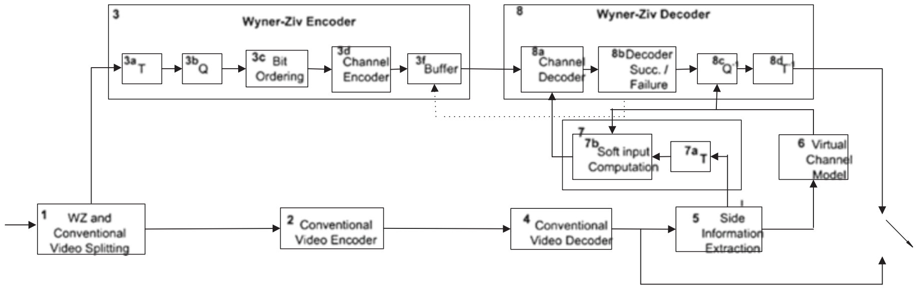

3
Wyner-Ziv Encoder
3a T
3b Q
3c Bit Ordering
3d Channel Encoder
3f Buffer
8
Wyner-Ziv Decoder
8a Channel Decoder
8b Decoder Succ. / Failure
8c Q-1
8d T-1
7
7b Soft input Computation
7a T
6 Virtual Channel Model
1 WZ and Conventional Video Splitting
2 Conventional Video Encoder
4 Conventional Video Decoder
5 Side Information Extraction

Figure 2 Block diagram of the reference WZ video coding architecture.

H.264 encoder to be accelerated. Inter prediction in H.264 allows not only the use of the MBs in which the images are decomposed for the ME and Motion Compensation (MC) processes, but also the use of partitions resulting from dividing the MB in different ways. Greater flexibility for the ME and MC processes, and greater MV precision, gives greater reliability to the H.264 encoding process. This feature is known as variable block size for the ME; Figure 3 shows the different block sizes in which an MB can be divided and Figure 4 shows the MB sub-partitions in which the 8 × 8 partitions can be further divided.

For each of these different MB sizes, the ME procedure is performed and a separate vector is required for each partition or sub-partition. The encoder encodes an MB using all possible modes (inter, intra, or skip) depending on the length of the MV and the amount of residual (cost). Encoding an MV for each partition can take a significant number of bits, especially if small partition sizes are chosen. MVs neighbouring partitions are often highly correlated and so each MV is predicted from vectors of nearby, previously coded partitions. The MVs prediction-forming method depends on the MC partition size and on the availability of nearby vectors. In order to evaluate the cost of the MB mode decision, the H.264 JM 17.0 reference software [13] implements a set of different evaluation methods, including a high-complexity mode, among others [14].

State-of-the-art

The main objective that a transcoding process should have is to try to work out what calculations and processes that have been carried out in the first stage could be re-used in the second half. All the information that has to be generated, but which could have been approximated by the data gathered in the first stage, is wasted computing time in the transcoding process. In fact, transcoding algorithms between traditional video coding standards are easier to accelerate due to the fact that the input and output video formats are based on more comparable paradigms. Many different transcoding approaches, based on traditional standards, have been proposed in the literature, yet there are only a few approaches based on DVC which have been proposed recently for transcoding WZ to H.263 [15] or WZ to H.264 [16-18]. Although the idea of applying the DVC paradigm in a transcoder framework was introduced in [19], it was not until 2008 that Peixoto et al. [20] proposed the first architecture to support mobile-to-mobile communications by a WZ to H.263 transcoder. In this approach, a reutilization of the MVs to accelerate the time spent on the H.263 ME was also made. However, H.264 offers better performance than H.263 in RD terms [21] and its inter prediction algorithm is also more complex than the one implemented in H.263. Furthermore, the authors of [15,20] failed to fully exploit the correlation between the WZ MV and the traditional ME and only used them to determine the starting centre of the H.263 ME process. Moreover, the acceleration of the ME process was not measured numerically. Instead the authors showed a formula to approximate the time saved [15] which is useless for other researchers.

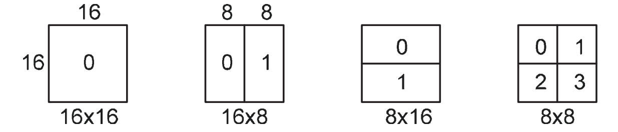


Figure 3 MB partitions in H.264.

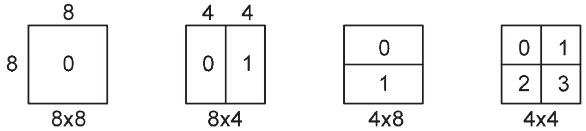


Figure 4 8 × 8 sub-partitions in H.264.

However, these drawbacks were tackled in our previous work, in particular in [18]. In 2009, the authors of this article proposed an improved WZ to H.264 video transcoder that reuses the incoming MV of the SI generation in order to reduce the ME process performed in H.264. Moreover, in [16], we propose a decision tree algorithm that replaces the MB-coded mode decision algorithm used at the H.264 encoder while the ME itself is left untouched. The different MB coding mode partitions are reduced into a sub-set based on the cited algorithm. Finally, in [17], the authors combine both approaches.

On the one hand, both previous approaches are focalized and adapted for ME [18] and MB-partitions [16] but they are not optimized to work together. In this study, we have improved and adapted each one of them in order to they can operate together. As it will be shown later, it is not necessary to develop a deep decision tree for the MB-partition because when combining with reduced ME, the tree could be simplified to a more simple decision where only a sub-set of partitions are checked.

On the other hand, all approaches previous to this study [16-18] can only convert from the incoming WZ GOP length 2 to only the I-P-I-P H.264 GOP output transcoder. However, both DVC and H.264 GOP sizes and format used in this transcoder were not realistic because (1) DVC supports higher GOPs than 2 and, (2) the bitrate generated by the I-P-I-P H.264 pattern is too high as well as being unpractical. This article proposes a transcoding algorithm from any incoming WZ GOP size and format to baseline profile IPPP pattern in

H.264. In addition, in [16-18] a functional WZ implementation approximation was used, which was far from realistic. In this article, the WZ architecture has been changed for a new one (depicted in Figure 2), based on the VISNET-II [4,9] project and using the VISNET2-WZ-IST software, which implements lossy key frame coding and on-line correlation noisy modelling, and uses a more realistic procedure for the stopping criterion implemented on the decoder side. Moreover, compared to previous works [16,18], some statistical information extracted from the SI residual frames, such as mean and variance, was determined to be very correlated with the H.264 MB mode decision. Nevertheless, the results were not optimized: neither the ML process nor the MV from WZ. In this study, an exhaustive analysis of the statistical information together with a better DM process (both data collect and ML algorithm) has been carried out. At present, the goal is to offer a WZ to H.264 transcoder that will be able to exhibit better performance over all kinds of video content and video formats than previous ones. Furthermore, in this study more sequences and more results will be shown. In other words, this article shows that the new approach is able to reduce the encoding time by up to 93%, compared with the cascade reference transcoder. Moreover, this has no PSNR or bitrate impact; in fact, these values are also optimized, since a better process has been carried out. And we shall describe the proposed transcoding algorithm in greater depth and with more details than in previous works, which is another reason for this article.

Proposed transcoder

To provide a mobile video communication framework with low complexity at both ends, this article proposes a WZ to H.264 video transcoder which keep the complexity of the end-user devices as low as possible. In this framework, it is necessary to convert from a source format with low complexity at the encoder (WZ bitstream), to another one with low complexity at the decoder (H.264 bitstream). The architecture of the proposed transcoder is depicted in Figure 5, where the first stage is composed of a WZ decoder based on the VISNET-II architecture [4,9]. Specifically, we have chosen the transform domain (TD) because of the better RD results obtained [4]. The reference transcoder is composed of the full WZ decoding and full H.264 encoding algorithms, but both processes working sequentially implies higher time consumption and thus higher delays. As mentioned in Section 2.2, the inter prediction part of the WZ/H.264 transcoder takes a long time to search all inter modes and intra modes for inter-frame coding exhaustively, and it is the most suitable part to be accelerated. Therefore, the improved transcoder presented in this article reuses some of the operations performed in the first half (WZ decoding) to prevent unnecessary operations from being performed in the second half (H.264 encoding). The data which are passed to the second half of the transcoder are depicted in Figure 5 by dotted lines.

The following sections will specifically describe the improvement developed in the proposed transcoder. First, Section 4.1 outlines the motivations behind the proposed approach and the information which is suitable for correlation between the two paradigms. Then, Sections 4.2. and 4.3 describe two different techniques (which can in fact operate together) that can be applied to speed up the H.264 encoding algorithm of the transcoder by using information gathered in the WZ decoding part.

Motivations and observations

As has been explained in Section 2.2, the inter prediction procedure performed by the H.264 encoding algorithm can be tackled in two ways: the procedure called MB mode decision (different partitions of each MB) and ME itself. The homologous procedure to the inter prediction carried out on the WZ side is the SI creation process. This process tries to determine the movement of the called WZ frames by means of motion compensated interpolation techniques (as well as extrapolation) between adjacent key frames. As has been said before, this procedure is very similar to the ME and MC processes performed on the H.264 encoder side. It thus seems that most of the operations and data generated in the SI generation process can be useful in the inter prediction algorithm, or at least this information can be used to reduce the time taken.

The SI generation is a crucial task for any WZ codec due to the fact that WZ frames are decoded and reconstructed on the basis of the SI. The SI can be seen as the starting point for the WZ decoded frame, and the better of the quality of the SI, the better the final quality of the decoded WZ is. There are many studies on this topic but there are basically two main approaches: hash-based ME [22,23], and Motion Compensated Temporal Interpolation [4]. In particular, VISNET-II codec employs the latter one. For this purpose, in this article the approach adopted for SI generation is the one proposed in [24]. In Figure 6, the first step of SI generation is shown, which consists in matching each forward frame MB with a backward frame MB inside the search area. Frame k means that this is a Key frame coding and $k+n$ is the next key frame which delimits the GOP, thus n is the GOP length. This matching takes all the possibilities into account and chooses the lowest residual MB. Through this process, for each MB an MV is obtained which quantifies the displacement between the two MBs, and the middle of this MV represents the displacement for the interpolated MB. As has been mentioned above,

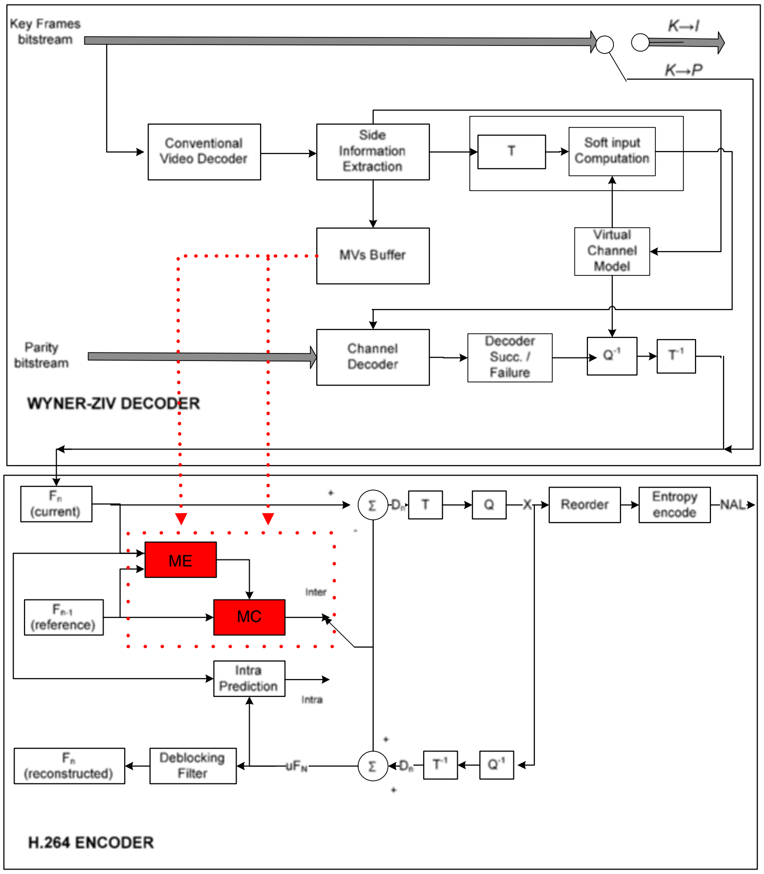


Figure 5 Proposed WZ to H.264 video transcoder.

this procedure is closely correlated with the traditional ME performed in traditional video codecs such as H.264 for a P-frame; furthermore, as in traditional video codecs, in this step an MV as well as a residual block is obtained. The complete SI estimation procedure is detailed in [4]. Although in the SI generation process a bidirectional refinement is carried out and this could be seen as an approximation of a B-frame H.264 ME, this article focuses on the H.264 Baseline profile where B frames are not allowed; but, the present approach can easily be extended for the B frames treatment in H.264.

Therefore, the information suitable for use in the H.264 encoding algorithm (which should be stored) is (1) the MVs, because they can be correlated with those that will be calculated at H.264 ME (as will be shown in Sections 4.3); as well as (2) the lowest residual that determines the current MB and the MB displaced by the MVs, because this residual can be correlated with the procedure called MB mode coded partitions (as will be shown in Section 4.2).

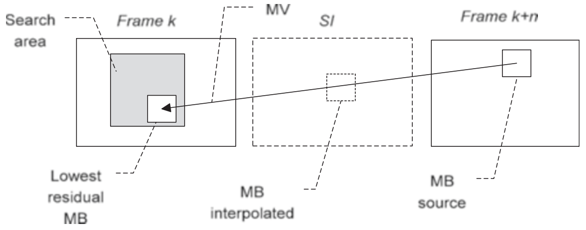


Figure 6 First step of SI generation process.

The present approach tries to generalize for every WZ GOP size, and therefore the data collected must be collected properly. Figure 7 shows the procedure when the distance between K frames (or source frames) is two, but in WZ video coding any GOP size is allowed, although in the literature the most commonly used GOP sizes are 2,4 and 8. The first K frame is passed to an I-frame without any conversion, because it was encoded (at the sender user device) using H.264 Intra. On the other hand, for every WZ frame one SI is estimated, as well as one MV for each MB (those which achieve lower residual). This is shown in the top row, where V_{0-2} represents the MVs calculated between K_0 and K_2 in order to estimate SI_1 for WZ_1 and so on. In other words, V_{0-2} in Figure 7 corresponds with MV in Figure 6.

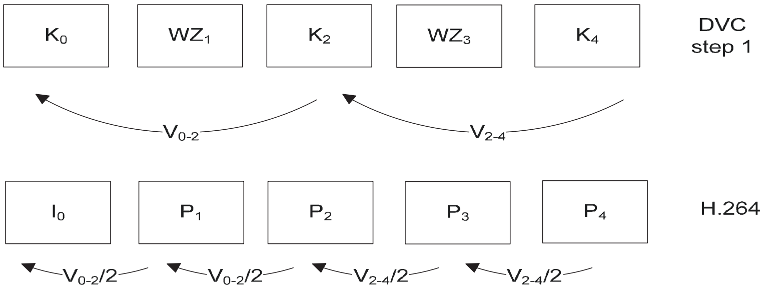


Figure 7 Mapping from DVC GOP 2 to H.264 GOP I11P.

This idea can be extended for higher WZ GOPs but some considerations must be taken into account. Figure 8 shows the transcoding process for a WZ GOP 4. As is shown in the two top rows, the WZ decoding algorithm divides the decoding process into two steps. In the first step, WZ_2 is decoded by calculating the SI between K_0 and K_4. The information collected in this step (MVs V_{0-4} as well as the residual) is ignored as it has low accuracy. In the second step, there is a reconstruction (using the parity bit sent by the WZ encoder) of the WZ_2 frame (labelled WZ'_2), and now this case is similar to the previous one shown in Figure 7 (when the distance is 2). These vectors, which were generated in step 2, will be used in our proposed transcoder to accelerate the H.264 encoding algorithm. However, the vectors generated in step 1 are discarded because long distances contain less accurate information. For longer GOPs, the procedure is the same because the last DVC step (that with distance 2) is always contained. In the case of IPPP, in the H.264 stage P frames have references with distance 1 whereas MVs were calculated for distance 2. For this reason, the useful vectors (V_{0-2} and V_{2-4}) are divided into two halves.

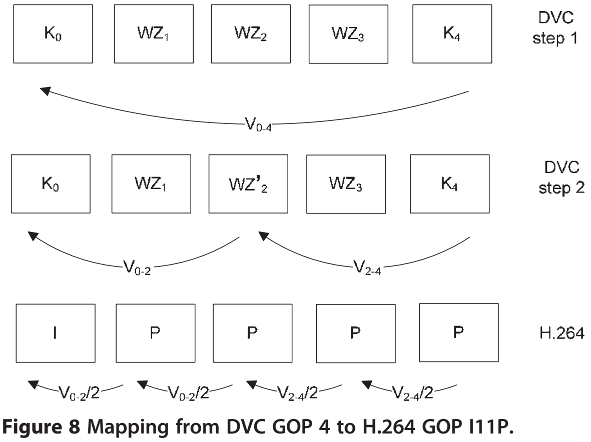


Figure 8 Mapping from DVC GOP 4 to H.264 GOP I11P.

Decision tree algorithm for MB mode coded partition

The starting point for this idea was the “look-and-feel” of the mathematical statistics applied to the WZ SI motion residual information. For example, the Flower Garden ME is represented against the mean and the variance in Figure 9. In this way, we show the steps followed to give shape to the idea presented (all the pictures refer to the second frame in the sequence). Figure 9a shows the original YUV second frame, Figure 9b shows the WZ SI motion residual information (|WZ − SI|), Figure 9c,d shows the mean and the variance statistic applied to this WZ motion residual information in 4 × 4 blocks, respectively. On the other hand, Figure 9e,f shows the different types of encoded MB supported by the H.264 standard, and the MB mode selection made by the H.264 reference software for coding each MB within the frame, after decoding the sequence using the WZ decoder. As the coding mode changes from low complexity modes (skipped, 16 × 16, 16 × 8, etc.) to high complexity modes (8 × 8 and Intra modes), the mean and the variance value increases for the corresponding MB, as can be clearly seen. Comparing the images shown in Figure 9, we can see that there are some relationships between the processed WZ motion residual information (generated in the SI generation process) and the H.264 mode partition selection.

DM is the process of finding correlations or patterns among dozens of fields in large relational databases. These datasets are made up of any facts, numbers, or text that can be processed by a computer. Then, the patterns, associations, or relationships that could exist among all this data can provide very valuable information for an extensive range of applications, including, among others, video coding. The information gathered in this way can be converted into knowledge, for use in applications that rely on knowledge. ML is concerned with the design and development of algorithms and techniques that allow computers to “learn”. The way in which DM algorithms obtain knowledge can vary depending on the ML paradigm they are based on, such as artificial neural networks (models that learn and look like biological neural networks in structure), rule

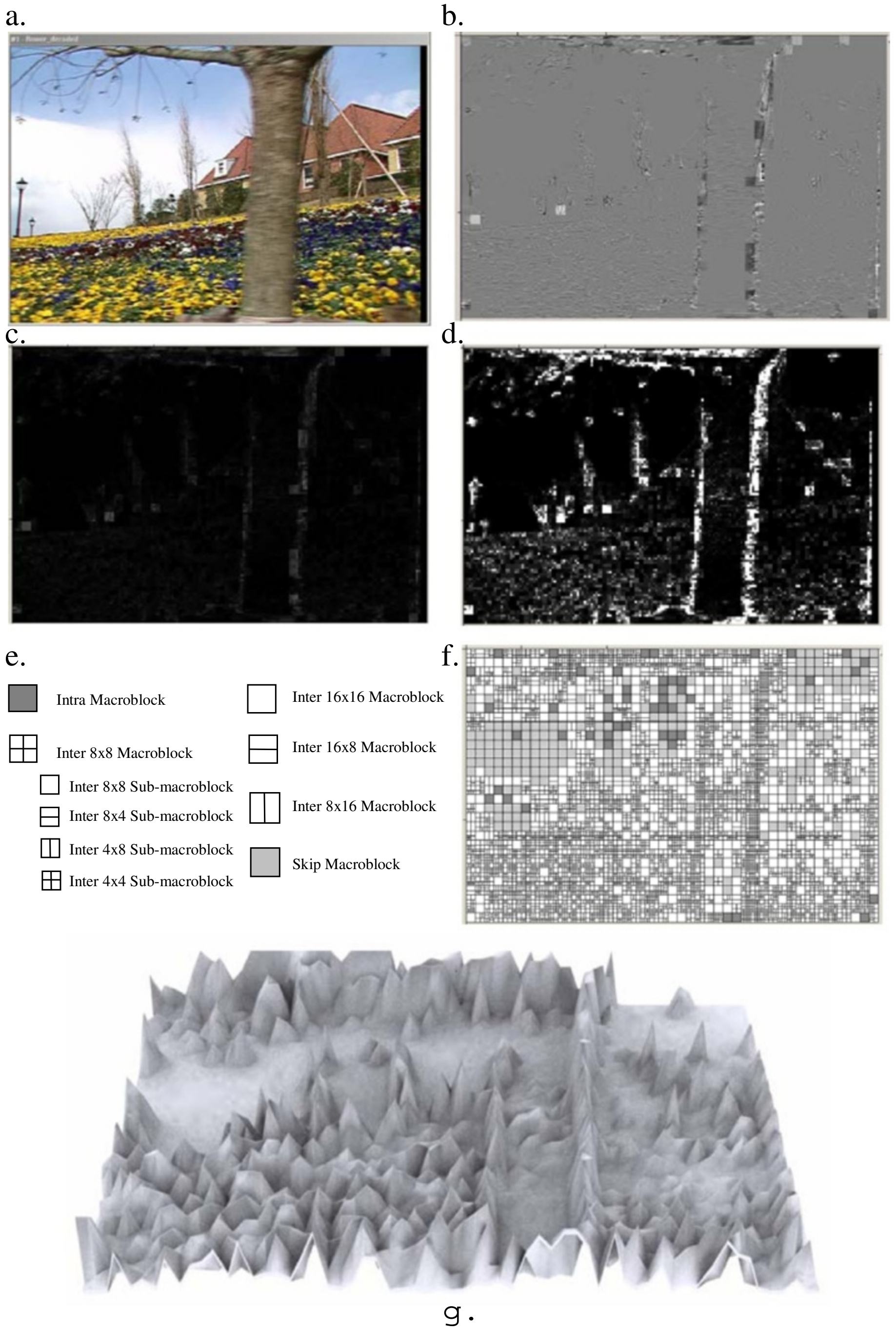


Figure 9 Correlation between residual and MB mode decision.

induction (generating if-then-else rules from data using statistical methods), or decision trees (made by mapping the observations about a set of data in a tree made of arcs and nodes) [25].

At this point, the present approach consists of trying to understand (using an ML algorithm) all this statistical information and create a decision tree to be implemented as part of the H.264 encoder algorithm in order to replace the more complex reference one.

Training stage

The software used in this DM process was WEKA v3.6 [26]. WEKA is a collection of ML algorithms for DM tasks. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization and it is an open source tool available at [26]. The training files are formed using some statistical information extracted from the SI generation process and the corresponding MB codec decision. The minimum unit to apply these concepts is the MB; this procedure extracts the statistical information from every MB. This information about each MB is one instance to understand.

In our previous works [16,17], we introduced the use of DM techniques in the framework of WZ to H.264 video transcoders, as a part of the process for designing rules to let us classify or decide what the best option is for encoding an MB using an H.264 encoder. One of the first conclusions that we drew was that we should continue using rule learning algorithms for creating the decision trees, since the knowledge representation is very easy to interpret, analyse and modify, if you want to include your own knowledge. Additionally, you can modify the rules in order to observe the behaviour of one of them in particular, and identify those that are most important. Furthermore, and given the extra time that you obtain when combining different approaches for reducing the amount of computational time needed for the ME and compensated process (fast inter mode selection, MV refinement techniques, etc.), it is possible to let the decision tree be more flexible in the MB coding mode selection. In order to test this idea, this article is going to analyse the effect of having nodes of the tree that can select the best coding mode from a set of modes. In this way, we follow the same line as that of previously refereed works [16,17], while improving upon the results by using a better DM process. Basically, in this new process: (1) the statistical information extracted from the residual frame has fully been studied and new variables that are more correlated and stronger have been introduced in the DM process, and these new statistics give us more generic trees; (2) binary and more general trees have been introduced; each leaf on the tree is a big bin that includes similar (in terms of correlation) MB and sub-MB partitions, the algorithm is more flexible and lets the H.264 encoder choose any MB decision inside this bin; and (3) it uses a new and more accurate ML algorithm that leads to shorter and more efficient trees.

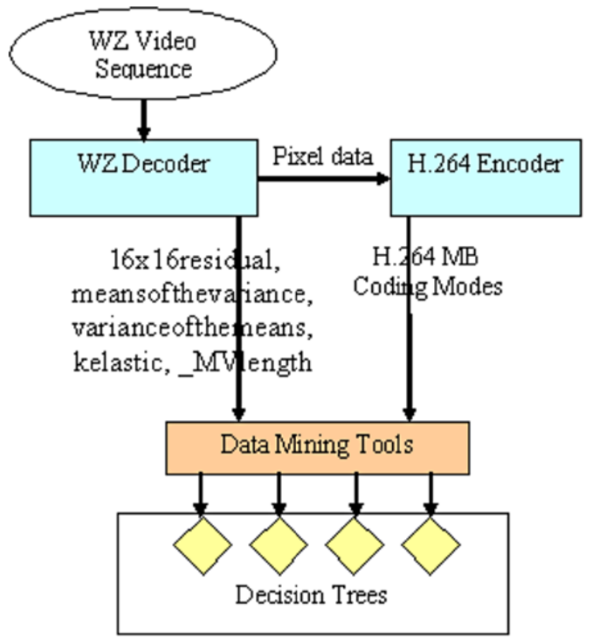


Figure 10 The decision tree.

The JRip rule learner (WEKA's implementation of the RIPPER rule learner [27], proposed by Cohen), a fast algorithm for learning "IF-THEN" rules, was used for creating the rules of the different nodes in the decision tree. RIPPER, like most classification learners, requires a set of examples to be represented as a vector of relatively simple features for learning and creating the rules. Figure 10 shows the overall operation of the proposed transcoder training process. The WZ video is decoded and we gather as much information as we can in order to give WEKA attributes to work with. From the SI generation process, the proposed algorithm stores (1) the 16×16 block residual; (2) the means of the variances; it computes the sixteen 4×4 variances of the sixteen sub-partitions that an MB can be divided into, and then returns the mean of all of them; (3) the variance of the means; in this case, the sixteen 4×4 means are calculated, and then their variance is returned; (4) the *kelastic* variable, defined as

$$\text{Kelastic} = \left(\frac{\text{Residual}}{\text{Count}}\right) * \left(1.0 + \frac{\sqrt{\text{dx} + \text{dy}}}{2}\right)$$

where *residual* is the amount of the residual of the whole MB; *count* is the number of pixels per MB (16×16, 256 pixels); and *dx* and *dy* are the corresponding MVs for this MB; and finally, (5) the MV length defined as $\sqrt{dx + dy}$. The class variable used for classifying the samples will be the decisions made by the H.264 reference software encoder (JM version 17.0 [13]), in the training sequence. In this process, we applied supervised learning because we found that, after some experimentation, there are MB partitions that are more correlated to each other than others. Therefore, in order to define mode sets when we need them, the H.264 reference software decisions will be modified to create binary decisions, i.e., instead of trying to determine the final MB mode decision (such as 16×16, 16×8, Intra or whatever), the proposed algorithm only determines whether the final MB codec decision belongs to LOW COMPLEXITY or HIGH COMPLEXITY sets, which are defined as follows: The first one is made up of {SKIP, 16 × 16, 16 × 8, 8 × 16} and the second one of {8 × 8, 8 × 4, 4 × 8 and 4 × 4}. Intra mode will always be checked, although at the beginning the algorithm also tries to determine whether an MB can be coded as Intra. After some experimentation, we found that the RD results are better if both classifications (LOW COMPLEXITY and HIGH COMPLEXITY) can check for Intra coding. All this procedure is depicted in Figure 11.

Under an exhaustive DM process, we developed many different training files to work out which is the best sequence for training, and also how many instances are enough for good knowledge acquisition. After trying with different kinds of sequences, and for each sequence a different number of frames, we found that sequences which contain varying regions from homogenous to high-detail serve as good training sets. Good sample sequences could be Flower and Soccer sequences. Basically, in this mode of operation, the testing file information is used to check the selection of the decision tree (which is generated by using the training file) and this selection is compared with the nominal class that appears in the testing file. This process was carried out for all the possible combinations (training with one file and testing with the rest).

In fact, the study was carried out by using 12 sequences and 30 different combinations of frames selected (starting with 1, 5,. . . up to the whole sequences in steps of 5 frames). Therefore, a total of 360 training files were obtained and we found that the soccer sequence using all frames obtained the best results in terms of percentage of good selection against the rest of the sequences. Finally, the best training set was made using 150 P-frames (the whole sequence) in the Soccer sequence (QCIF format, frame rate 15 Hz), encoded at matrix for quantization of the WZ frames of 7 [4], and key frames were encoded with the H.264 [1] QP 31. The H.264 reference software decisions for the training set were obtained by encoding the WZ decoded sequence with a QP of 28 (same parameters and conditions that will be used in the Performance Section for testing the model). Since the Soccer sequence is used for training the decision tree, this sequence will not be included in the tested sequence set. So, one of the basic principles of the DM is met. Finally, since the QP used by H.264 is designed to change the quantization step size and the relationship between the QPs is well defined, this relationship can be used to adjust the thresholds of the statistics (residual, variance of means, means of variance, etc.) used in the Decision Tree. The proposed transcoder uses a single decision tree developed for a mid-QP of 28 and then adjusted for other QPs. Since the quantization step size in H.264 doubles when QP increases by 6, the thresholds are adjusted by 12.5% for a change in QP of 1.

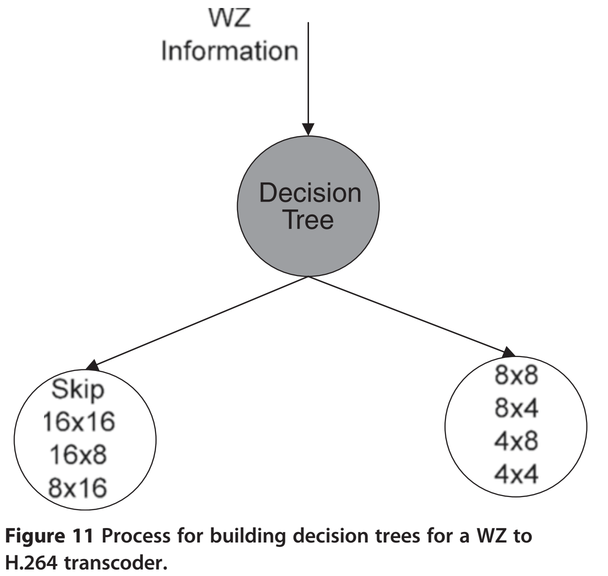


Figure 11 Process for building decision trees for a WZ to H.264 transcoder.

The decision tree

The decision trees proposed in this article as a solution to replace the MB coding mode decisions in WZ to H.264 video transcoders consist of leaves and branches, as is shown in Figure 10. The leaves are the classifications and the branches are the features that lead to a specific classification. A tree decision is a classifier based on a set of attributes allowing us to determine the category of an input data sample.

The gray circle in Figure 10 represents the decision tree and the white circles represent a set of MB partitions where the reference standard can choose. In other words, the proposed technique does not focus the final MB partition for the input block but instead focuses the different selections into a reduced set based on the correlations between the variables mentioned in this section and the final MB mode selection. The output of the tree is a set of H.264 MB modes.

The MB coding mode decision, or decisions, determined by the decision trees is used in the low complexity H.264 encoding stage. This is an H.264 reference software encoder with the MB mode decision replaced by simple mode(s) assignment from the decision tree. So, this article presents an unbalanced decision tree based on a non-pruning JRip algorithm [25] which gives a high level of freedom for choosing the MB partition which exhibits the minimum cost over the SAE-cost algorithm that has been implemented in the reference software encoder. The final decision is LOW COMPLEXITY or HIGH COMPLEXITY which corresponds with a sub-set of the MB-partitions (as it is depicted in Figure 10). The computation cost that we have to pay for the white tree freedom nodes will be compensated for by the time reduction that is achieved with a motion search window refinement, as will be explained in the following section. Figure 12 shows the different rules defined in the decision tree algorithm.

The decision tree could be extended with lower levels composed of more specific leaves, which would check fewer partitions. The DM procedure could be applied again and another decision tree could be generated for dividing both the LOW COMPLEXITY and/or HIGH COMPLEXITY leaves. This configuration will offer better results in terms of time reduction, but at the expense of an RD penalty. We found that in combination with the approach presented in the following method (Dynamic Motion Window—DMW), which reduces the ME procedure, only a two level decision tree as depicted in Figure 10 is enough in terms of a trade-off between time reduction and RD.

DMW for ME

As has been mentioned above, a big part of this H.264 encoding complexity depends largely on the search range used in the H.264 ME process, which is a consequence of the quantity of checking done. However, this process may be accelerated because the search range can be reduced by avoiding unnecessary checking without a significant impact on quality and bit rate. The MVs generated in the SI generation process for a given MB indicate the effectiveness of the ME in the WZ decoder, and this information can be used to reduce the ME complexity in the H.264 encoder. Adapting the search range based on the MVs of the incoming MB could reduce the ME complexity without severely impacting the PSNR. To achieve this aim, we propose to reuse the MVs calculated in the WZ decoding algorithm to define a smaller search range for each MB of H.264 including every sub MB partition. The MB with zero MVs in WZ represents very simple MBs, so the search range area is limited to 4 pixels. In the case of WZ MB modes that contain MV information, the search range is determined by the area created by the circumference equation, centred on the (0,0) point for each H.264 mode or sub-mode, with the length of the incoming WZ vector being the radius of the circumference. In this way, the length of the WZ MV will limit the search range area. So, the checking area is limited by the area S defined in expression (1):

$$S = \{(x, y)/(x, y) \in (A \cap C)\} \tag{1}$$

where (x,y) are the coordinates to check, A is the search range used by H.264 and C is a circumference which restricts the search with centre on the upper left corner of the MB. C is defined by Equation (2):

$$C^2 = r_x^2 + r_y^2 \tag{2}$$

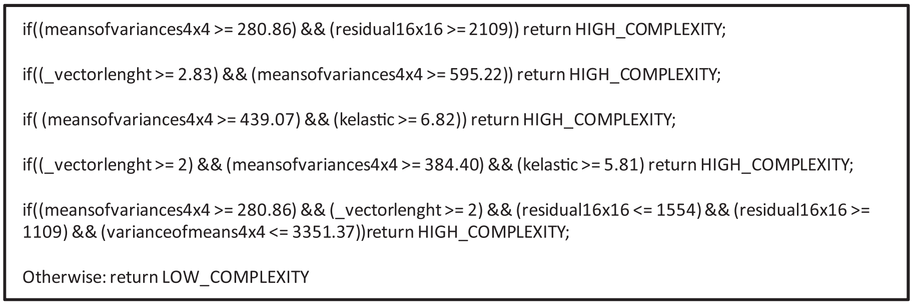
$$r_x = \max\left(\frac{\mathrm{MV}_x}{2}, 4\right) \tag{3}$$

$$r_y = \max\left(\frac{\mathrm{MV}_y}{2}, 4\right) \tag{4}$$

where r_x and r_y are calculated from Equations (3) and (4), depending on the MV halves $\left(\frac{\mathrm{MV}_x}{2} \text{and} \frac{\mathrm{MV}_y}{2}\right)$ provided by DVC or a minimum value of 4 to avoid applying search ranges that are too small. Figure 13 shows the procedure for this approach, termed DMW.

Performance evaluation

The proposed WZ to H.264 Transcoder for the Baseline Profile has been implemented in the H.264/AVC JM 17.0 reference software [13]. It consists of two parts: a WZ decoder followed by an H.264 encoder. Figure 5 shows the architecture of the proposed WZ to H.264 transcoder. Firstly, the transcoder fully decodes the WZ sequence, and the information required by the decision trees and the DMW mechanism is gathered in this stage. Then, the H.264 encoder encodes the sequences using our approaches for reducing the encoding time. The red modules in Figure 5 denote those modules that have been enhanced using various pieces of information coming from the WZ decoding stage. The MB coding mode decision determined by the decision trees is used in the low complexity H.264 encoding stage, and the H.264 MB



```
if((meansofvariances4x4 >= 280.86) && (residual16x16 >= 2109)) return HIGH_COMPLEXITY;

if((_vectorlenght >= 2.83) && (meansofvariances4x4 >= 595.22)) return HIGH_COMPLEXITY;

if( (meansofvariances4x4 >= 439.07) && (kelastic >= 6.82)) return HIGH_COMPLEXITY;

if((_vectorlenght >= 2) && (meansofvariances4x4 >= 384.40) && (kelastic >= 5.81) return HIGH_COMPLEXITY;

if((meansofvariances4x4 >= 280.86) && (_vectorlenght >= 2) && (residual16x16 <= 1554) && (residual16x16 >=
1109) && (varianceofmeans4x4 <= 3351.37))return HIGH_COMPLEXITY;

Otherwise: return LOW_COMPLEXITY
```

Figure 12 The decision tree algorithm.

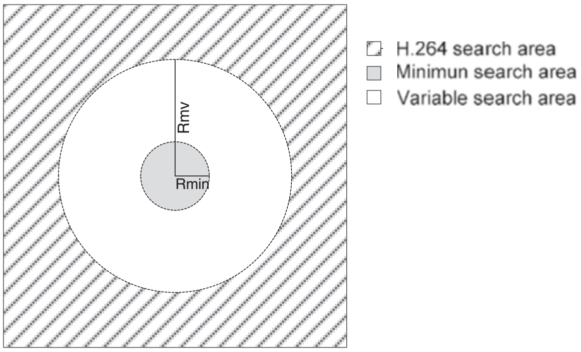


Figure 13 Search area reduction for H.264 encoding stage.

mode decision is replaced by a simple mode assignment, or a range of them. Moreover, the WZ MVs are used for the DMW approach, and the low complexity encoder only performs the ME for the final MB mode determined by the decision trees in the search range area fixed by the DMW mechanism, as we explained in Section 4.

This performance evaluation included an extensive set of experiments with videos representing a wide range of motion, texture and colour (depicted in Figure 14). Twelve sequences were fully transcoded, where the total number of frames for every one of them was simulated. Furthermore, experiments were conducted to evaluate the performance of the proposed algorithm when transcoding videos at commonly used frame rates: 15 and 30 frames per second. All of them were in QCIF (176 × 144) format, which is the most suitable resolution for mobile to mobile video conferencing.

The input to the transcoder (the WZ bitstream coming from the sender device) was generated using a quantification parameter which deals with an acceptable quality of the decoded video sequence because if we encoded the WZ video at a lower bitrate (higher quantization), the relation PSNR against bitrate in the transcoder (H.264 encoder) would very soon be saturated, and we want to avoid that. Although, the results achieved in this article could be extended to any WZ input bitrate. This is mainly because we will not have more quality in the H.264 stream output than the one that is given by the WZ decoder. The output bitrate can efficiently be controlled by the transcoders (the data injected to the network will be H.264 encoded video), and we can adjust the QP in order to satisfy the different quality

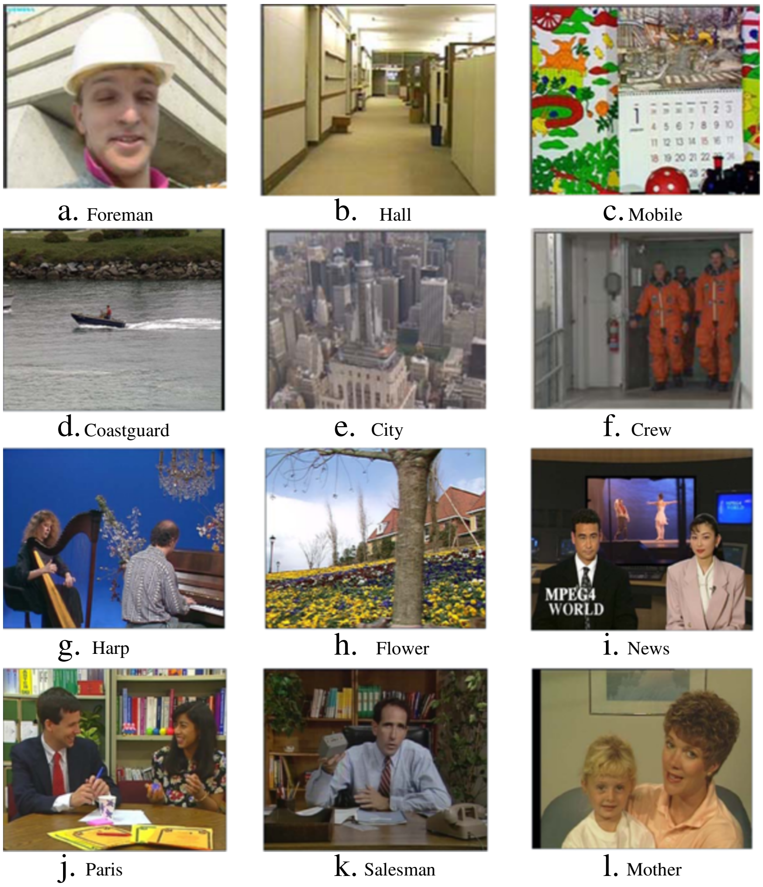


Figure 14 Test sequences.

requirements, as well as the bandwidth limitations depending on the possibilities of the end-user devices or the network conditions. Tables 1 and 2 show the bitrates, quality and the total number of frames for the encoded WZ video sequences used as input for the transcoder. WZ GOP will be changed between 2, 4 and 8, which allows different combinations of network requirements as well as WZ encoding complexities. These rate and distortion performances were obtained by selecting the seventh matrix of quantization in WZ encoder parameters in TD according to [10].

Encoding parameters

The parameters used in the H.264 encoder configuration file for testing the mechanism proposed against a cascade WZ to H.264 transcoder are those that are included in the encoder_baseline.cfg file contained in the JM17.0 reference software. Only four parameters have been changed in the encoder_baseline.cfg file, and the reasons are summarized in Table 3.

Other parameters which were changed in the encoder_baseline.cfg configuration file and that are not listed in Table 3 were assumed in order to obtain the results, such as QP range, width, height, etc. Furthermore, we should point out that the RDO option, in the configuration file, is disabled for all the simulations because this is not suitable for mobile to mobile video communications when the encoding time is critical.

Metrics

The metrics of interest are the RD function (bitrate) versus Quality (PSNR), the results in terms of ΔTime, ΔPSNR and ΔBitrate. These metrics are defined below:

- *RD function*: RD gives theoretical bounds on the compression rates that can be achieved using different methods. In RD theory, the rate is usually understood as the number of bits per data sample to be stored or transmitted. The notion of distortion is a subject of on-going discussion. In the simplest case (which is actually used in most cases), the distortion is defined as the variance of the difference between the input and the output signals (i.e. the mean squared error of the difference). In the definition of the RD function used to show the performance results, PSNR is the distortion for a given bitrate. The averaged PSNR values of luminance (Y) are used in the RD function graphs.
- *ΔTime, ΔPSNR and ΔBitrate*: In order to evaluate the timesaving of the fast MB mode decision algorithm, the following calculation is defined to find the time differences. Let T_{JM} denote the coding time used by the H.264/AVC JM 17.0 reference software encoder for the ME and compensation process, and T_{FI} be the time taken by the algorithm proposed or the mechanism that has been evaluated; ΔTime is defined as (Equation 5):

$$\Delta\mathrm{Time}(\%) = \frac{T_{\mathrm{JM}} - T_{\mathrm{FI}}}{T_{\mathrm{JM}}} \times 100 \tag{5}$$

T_{FI} also includes all the computational cost for the operations needed to prepare the residual information for our approaches, such as the mean, the variance and so on.

The detailed procedures for calculating these differences can be found in a JVT document authored by Bjøntegaard [28,29], which is recommended by the JVT Test Model Ad Hoc Group [30]. One of the outcomes is supposed to be RD-plots where PSNR and bitrate differences between two simulation conditions may be read. This mechanism is proposed for finding numerical averages between RD-curves as part of the presentation of results. This is a more compact and, in some senses, a4 more accurate way to present the data and it comes in addition to the RD-plots. The method for calculating

Table 1 Properties for the WZ input sequences (15 Hz)

Sequence	Number of frames	15 Hz scenario					
		GOP2		GOP4		GOP8	
		Bitrate (kbit/s)	PSNR (dB)	Bitrate (kbit/s)	PSNR (dB)	Bitrate (kbit/s)	PSNR (dB)
Foreman	150	280.1	35.56	306.7	34.69	337.4	33.92
Hall	150	200.7	37.34	158.5	37.05	139.6	36.67
Mobile	150	490.5	32.86	369.5	31.75	323.2	30.68
Coastguard	150	270.1	33.69	259.0	32.87	396.6	35.59
City	150	404.0	37.78	380.4	36.61	253.6	32.04
Crew	150	297.4	36.49	361.0	36.12	401.3	35.83
Harb	150	458.3	34,91	384..4	34.06	354.9	33.38
Flower	150	460.3	33.88	337.8	33.27	298.0	32.05

Table 2 Properties for the WZ input sequences (30 Hz)

Sequence	Number of frames	30 Hz scenario					
		GOP2		GOP4		GOP8	
		Bitrate (kbit/s)	PSNR (dB)	Bitrate (kbit/s)	PSNR (dB)	Bitrate (kbit/s)	PSNR (dB)
Foreman	300	503.9	36.95	469.1	36.25	489.9	35.35
Hall	300	433.6	38.8	315.6	38.43	260.8	37.91
Mobile	300	1013.4	34.01	708.6	32.99	573.2	31.88
Coastguard	300	608.3	36.24	462.5	35.2	418.4	33.78
News	300	601.8	40.4	404.2	39.1	311.5	37.86
Paris	300	901.7	40.17	582.4	38.79	436.1	37.29
Salesman	449	685.7	44.55	417.4	43.34	296.3	35.77
Mother	300	654.2	45.68	435.1	44.23	342.5	42.62

the average difference between two such curves is as follows:

- Fit a curve through four data points (e.g. bitrate/PSNR are obtained for QP = 28, 32, 36 and 40).
- Based on this, find an expression for the integral of the curve.
- The average difference is the difference between the integrals divided by the integration interval.

In order to show the transcoding results, the experiments were carried out on the test sequences with the four QPs, i.e. QP = 28, 32, 36 and 40, as specified in Bjøntegaard and Sullivan's common test rule [29]. The YUV files that will be compared to obtain the PSNR results are the original YUV file at the input of the WZ decoder and the one that is obtained after decoding the H.264 video with an H.264 decoder.

- *Grid Image*: Another metric that is used to analyze how the proposed approach works is the MB modes decision generated by the approach presented. The MB modes generated by the H.264 and the proposed algorithm are compared to measure the accuracy of the MB mode classification tree jointly with DMW. A Grid Image showing the MB modes overlaid on a corresponding frame is used to visually compare the MB mode classification.

Simulation results

Tables 4 and 5 show the results in terms of ΔTime, ΔPSNR and ΔBitrate. They also show the Average result for all the sequences for each GOP. In this way, an idea about the normal operation of a transcoder over all kinds of video contents can be extrapolated from this result. Compared with the cascade WZ to H.264 reference transcoder, and for the average of all the sequences, the proposed transcoder has a PSNR drop of at most 0.153 dB for a given bitrate, and a bitrate increase of at most 3.74% for a given quality. This negligible drop in RD performance is more than offset by the decrease in computational complexity, which is reduced by around 94% for the average of all the sequences. Time reduction is vital in real-time WZ to H.264 transcoders, since it determines the incoming stream delay in the end-user devices.

Figures 15 and 16 show the RD graphic results for the reference and the proposed transcoder, for different frame rate scenarios in the three different incoming GOP sizes analyzed, from a value of 28 to 40 for QP. As seen from the figures, the PSNR-bitrate obtained with the proposed transcoder deviates slightly from the results

Table 3 Parameters changed in the JM17.0 encoder_baseline.cfg file and the reason

Parameter	Value	Explanation
Frames To Be Encoded	Depends on the number of frames for the sequence	For encoding the full sequence in each case
Intra Period	12	This GOP format is commonly used in the literature for testing the baseline profile
Number Reference Frames	1	Same reason as the IntraPeriod parameter. Besides, the main goal is real time, so we reduce the complexity to get it
Search Mode	−1	Full search range mode for the ME and compensated process
Search Range	16	By default it is set to 32. 16 pixels are enough for transcoding the QCIF sequences. Besides, the main goal is real time, so we reduce the complexity to get it

Table 4 ΔTime, ΔPSNR, and ΔBitrate for the compared sequences (15 Hz scenario)

Sequence	RD performance of the WZ/H.264 video transcoder—15 fps								
	GOP 2			GOP 4			GOP 8		
	ΔBitrate (%)	ΔPSNR (dB)	TR (%)	ΔBitrate (%)	ΔPSNR (dB)	TR (%)	ΔBitrate (%)	ΔPSNR (dB)	TR (%)
Foreman	7.94	−0.308	92.02	8.30	−0.314	91.96	8.46	−0.303	91.69
Hall	0.98	−0.046	92.70	1.09	−0.048	92.46	0.93	−0.042	92.06
Mobile	1.19	−0.055	93.84	1.50	−0.073	93.65	1.32	−0.063	93.41
Coastguard	4.81	−0.215	93.95	5.36	−0.238	93.99	4.95	−0.216	93.70
City	3.21	−0.123	93.89	2.72	−0.101	93.78	2.50	−0.099	93.86
Crew	7.49	−0.310	92.56	8.30	−0.327	92.63	8.77	−0.327	92.69
Harb	1.52	−0.068	94.04	1.23	−0.059	93.99	1.47	−0.068	93.97
Flower	1.62	−0.073	93.67	1.40	−0.064	93.65	1.26	−0.058	93.61
Mean	***3.60***	***−0.150***	***93.33***	***3.74***	***−0.153***	***93.26***	***3.71***	***−0.147***	***93.12***

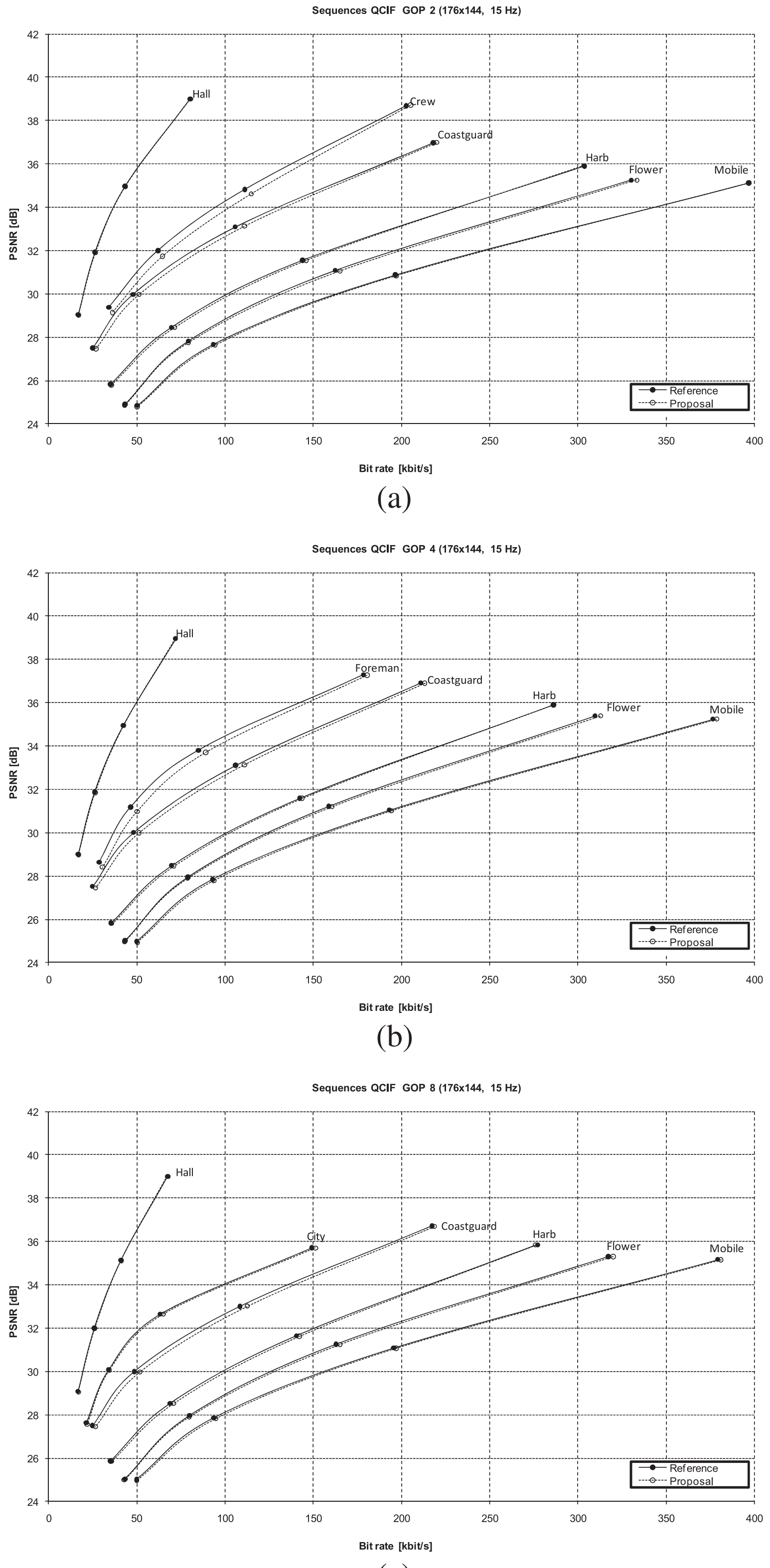
obtained when applying the considerably more complex cascade reference transcoder. Due to space limitations only a sequence sub-set for each resolution is shown, chosen for those that do not have an intersection between them, for each type.

Figure 17 shows the Grid Images obtained by using our proposed transcoder compared with the decision made by the cascade reference one, with a value of 28 for QP for the three sequences. From these figures, it is clear that the proposed algorithm obtains very similar results to those obtained using the full estimation process in the H.264 encoder.

As mentioned in Section 3, various WZ-based transcoders have recently been proposed in the literature. In Table 6, we undertake a comparative analysis of our proposal with some of the most relevant algorithms [15,18]. In these experiments, although the test conditions are not exactly the same as the ones reported in the literature, an objective comparison is still possible since all the algorithms follow Bjøntegaard and Sullivan's common test rule [28,29]. The video sequence used was Foreman (QCIF) coded in IPIP and IPPPI patterns, as in [15]. However, comparing the acceleration with [15] is impossible due to the fact that in [15] time results do not appear, which is shown in Table 6 as not available (n.a.). This is the major fault of the proposed transcoder based on H.263, as some RD penalty is allowed only if the corresponding time reduction is above acceptable values. As shown in Table 6, the performance of our WZ to H.264 video transcoder in terms of time saving, which is a critical issue in video transcoding applications, achieves the best results, with a negligible loss in video quality (0.2 dB), and with a slight increment in bit rate with respect to the other methods. This is due to the fact that the proposed approach reduces the H.264 MB mode computation process into a decision tree lookup with very low complexity and the transcoder only performs the fast ME for the final sub-set MB modes determined by the decision tree. Furthermore, the proposed transcoder can be implemented easily since it only requires some statistical information (mean, variance, the MVs and *kelastic*), which has been described in Section 4.2 of the WZ SI residual, and a set of rules to compare the mean and variance against a threshold.

Table 5 ΔTime, ΔPSNR, and ΔBitrate for the compared sequences (30 Hz scenario)

Sequence	RD performance of the WZ/H.264 video transcoder – 30fps								
	GOP 2			GOP 4			GOP 8		
	ΔBitrate (%)	ΔPSNR (dB)	TR (%)	ΔBitrate (%)	ΔPSNR (dB)	TR (%)	ΔBitrate (%)	ΔPSNR (dB)	TR (%)
Foreman	5.06	−0.198	92.40	5.07	−0.195	92.41	4.76	−0.177	92.24
Hall	1.75	−0.078	92.11	1.18	−0.052	92.08	0..83	−0.036	91.89
Mobile	1.25	−0.055	94.27	0.89	−0.041	94.16	0.63	−0.027	94.11
Coastguard	2.30	−0.091	93.91	1.99	−0.076	93.81	1.54	−0.060	93.69
News	1.60	−0.069	92.42	1.37	−0.059	92.41	1.05	−0.045	92.33
Paris	2.00	−0.084	93.22	1.26	−0.053	93.06	0.91	−0.037	93.09
Salesman	0.78	−0.034	93.21	0.63	−0.026	93.02	0.54	−0.022	93.03
Mother	0.46	−0.016	92.50	0.27	−0.009	92.33	0.30	−0.011	92.31
Mean	***1.90***	***−0.078***	***93.00***	***1.58***	***−0.064***	***92.91***	***1.32***	***−0.052***	***92.84***

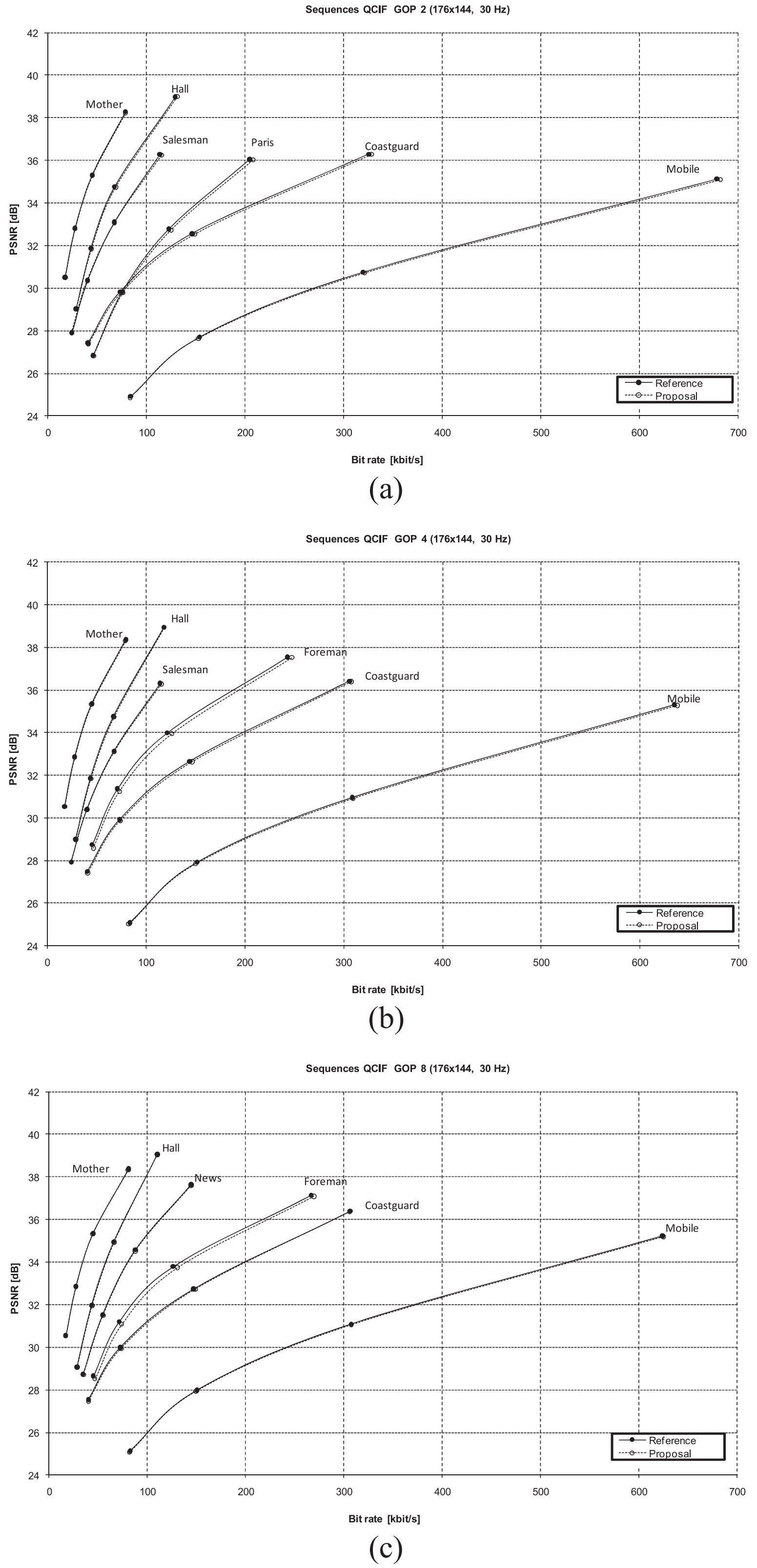


(a)

(b)

(c)

Figure 15 RD results comparing the performance of the proposed and reference WZ/H.264 transcoder. 15 Hz sequences: (**a**) GOP2, (**b**) GOP4, and (**c**) GOP8.

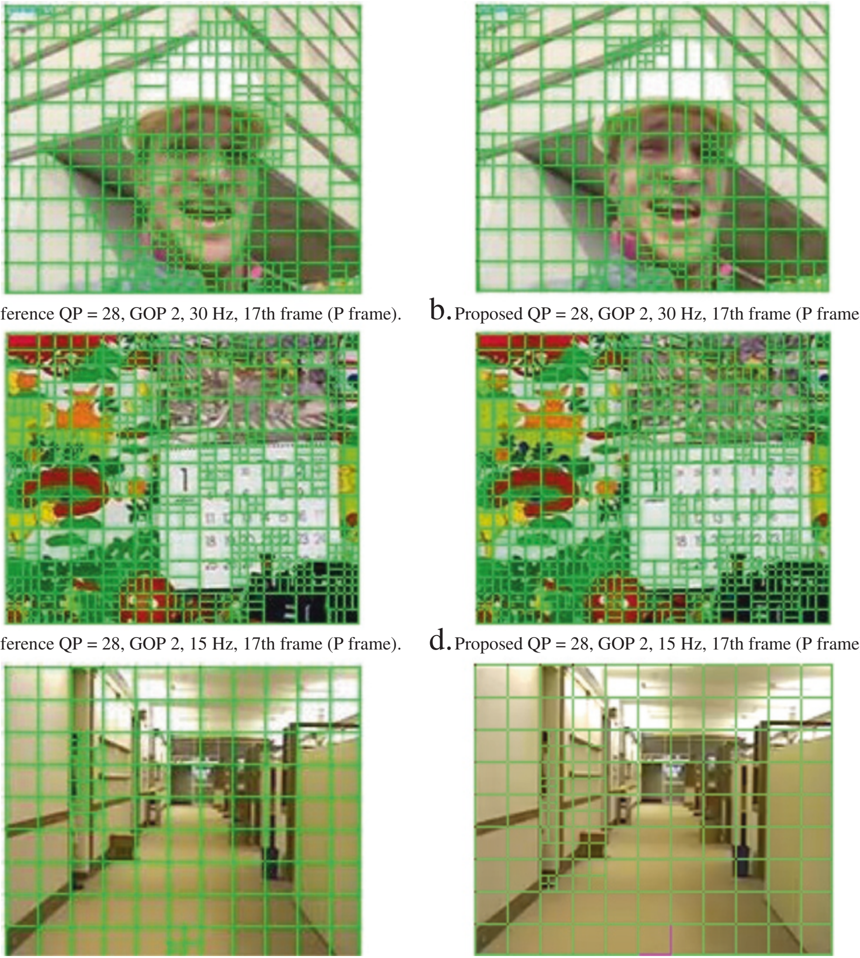


(a)

(b)

(c)

Figure 16 RD results comparing the performance of the proposed and reference WZ/H.264 transcoder. 30 Hz sequences: (**a**) GOP2, (**b**) GOP4, and (**c**) GOP8.



a. Reference QP = 28, GOP 2, 30 Hz, 17th frame (P frame).

b. Proposed QP = 28, GOP 2, 30 Hz, 17th frame (P frame).

c. Reference QP = 28, GOP 2, 15 Hz, 17th frame (P frame).

d. Proposed QP = 28, GOP 2, 15 Hz, 17th frame (P frame).

e. Reference QP = 28, GOP 2, 30 Hz, 17th frame (P frame).

f. Proposed QP = 28, GOP 2, 30 Hz, 17th frame (P frame).

Figure 17 MB mode decisions generated by the reference and the proposed transcoder for some QCIF sequences.

Conclusions

This study presents a low complexity and highly flexible WZ to H.264 transcoder for the baseline profile which is able to support mobile-to-mobile video communications. The architecture proposed permits conversion from each WZ GOP to the common IPPP GOP format, and consequently the transcoder can adapt the mapping to the framework requirements. In addition, the complex MB mode coded partition process is replaced by a decision tree algorithm which can determine a sub-set of the MB partitions and sub-partitions in which the H.264 encoder can search. This decision tree is based on a DM process which reuses most of the operations carried out in the WZ decoding algorithm, or to be more precise, in the SI generation process. Moreover, the proposed transcoder includes a dynamic algorithm which reuses the MVs

Table 6 Comparison of different WZ-based transcoding algorithms

RD performance comparison						
Proposal	**Sequence**	**Pattern**	**fps**	**ΔBitrate (%)**	**ΔPSNR (dB)**	**TR (%)**
Proposed transcoder	Foreman	IPIP	15	−2.01	−0.099	91.48
Peixoto et al. [5]	Foreman	IPIP	15	−1.08	−0.063	n.a.
Martinez et al. [18]	Foreman	IPIP	15	−0.27	−0.011	29..84
Proposed transcoder	Foreman	IPPPI	30	−2.26	−0.104	92.31
Peixoto et al. [5]	Foreman	IPPPI	30	−2.83	−0.067	n.a.
Martinez et al. [18]	Foreman	IPPPI	30	−0.11	−0.004	30.84

provided by the DVC decoding phase to accelerate the H.264 ME. In this way, a complexity reduction of 93% is achieved with a negligible RD loss.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This study was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. It was also partly supported by the JCCM funds under grant "PEII09-0037-2328" and "PII2I09-0045-9916". The present approach has been implemented by using the VISNET2-WZ-IST software developed in the framework of the VISNET II project. The authors would like to thank Eduardo Peixoto for his valuable support, which helped to improve the manuscript.

Author details

[1]Instituto de Investigación en Informática de Albacete (I3A), University of Castilla-La Mancha, Campus Universitario, Albacete 02071, Spain.
[2]Department of Computer Science and Engineering, Florida Atlantic University, 777 Glades Rd, Boca Raton, FL 34341, USA.

Received: 8 July 2011 Accepted: 15 May 2012
Published: 9 July 2012

References

1. ISO/IEC 14496–10 International Standard, *"Information Technology—Coding of Audio–1001 Visual Objects—Part 10: Advanced Video Coding"* (2003).
2. A Aaron, R Zhang, B Girod, *Wyner–Ziv coding for motion video, in Proceeding of Asilomar Conference on Signals* (Systems and Computers, Pacific Grove, CA, USA, 2002). November
3. AD Wyner, J Ziv, The rate-distortion function for source coding with side information at the decoder. IEEE Trans Inf Theory **IT-22**, 1–10 (1976)
4. C Brites, J Ascenso, JQ Pedro, F Pereira, Evaluating a feedback channel based transform domain Wyner–Ziv video codec. Signal Process. Image Commun. **23**(4), 269–297 (2008)
5. X Artigas, J Ascenso, M Dalai, S Klomp, D Kubasov, M Ouaret, *The discover codec: architecture, techniques and evaluation, in Proceeding of Picture Coding Symposium (PCS)* (Lisboa, Portugal, 2007). November
6. R Martins, C Brites, J Ascenso, F Pereira, Refining side information for improved transform domain Wyner–Ziv video coding. IEEE Trans. Circ. Syst. Video Technol. **19**(9), 1327–1341 (2009)
7. C Brites, F Pereira, Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding. IEEE Trans. Circ. Syst. Video Technol. **18**(9), 1177–1190 (2008)
8. JL Martinez, WARJ Weerakkody, WAC Fernando, G Fernandez-Escribano, H Kalva, A Garrido, Distributed video coding using turbo trellis coded modulation. Visu. Comput. **25**(1), 69–82 (2009)
9. VISNET II project, (2010). http://www.visnet-noe.org, Accessed September
10. J Ascenso, C Brites, F Dufaux, A Fernando, T Ebrahimi, F Pereira, S Tubaro, *The VISNET II DVC codec: architecture, tools and performance, in EURASIP European Signal Processing Conference (EUSIPCO)* (Aalborg, Denmark, 2010). August
11. ISO/IEC 13818–2, *Generic Coding of Moving Picture and Associated Audio* (MPEG-2 Draft International Standard, 1994).
12. ISO/IEC 14486–2, *Information Technology—Generic Coding of Audio-Visual Objects—Part 2: Visual* (PDAM1, 1999).
13. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *Reference Software to Committee Draft. JVT-F100 JM17.0* (2010)
14. IEG Richardson, *H.264 and MPEG-4 Video Compression* (John Wiley & Sons Ltd, New Jersey, 2003)
15. E Peixoto, RL Queiroz, D Mukherjee, A Wyner–Ziv video transcoder. IEEE Trans. Circ. Syst. Video Technol. **20**(2), 189–200 (2010)
16. JL Martínez, G Fernández-Escribano, H Kalva, WAC Fernando, P Cuenca, Wyner–Ziv to H.264 video transcoder for low cost video communications. IEEE Trans Consum Electron **55**(3), 1453–1461 (2009)
17. JL Martínez, H Kalva, G Fernández-Escribano, WAC Fernando, P Cuenca, *Wyner–Ziv to H.264 video transcoder* (in 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 2009), pp. 2941–2944. November
18. JL Martínez, G Fernández-Escribano, H Kalva, P Cuenca, Motion vector refinement in a Wyner–Ziv to H.264 transcoder for mobile telephony. IET Image Process **3**(6), 335–339 (2009)
19. B Girod, A Aaron, S Rane, D Rebollo-Monedero, Distributed video coding. Proc IEEE **93**(1), 71–83 (2005)
20. E Peixoto, RL de Queiroz, D Mukherjee, *Mobile video communications using a Wyner–Ziv transcoder, in Symposium on Electronic Imaging, Visual Communications and Image Processing (SPIE)* (San Jose, USA, 2008). January
21. T Wiegand, G Sullivan, G Bjntegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circ. Syst. Video Technol. **13**(7), 560–576 (2003)
22. A Aaron, S Rane, B Girod, *Wyner–Ziv video coding with hash-based motion compensation at the receiver, in IEEE International Conference on Image Processing (ICIP)* (Singapore, October, 2004)
23. J Ascenso, F Pereira, *Adaptive hash based side information exploitation for efficient Wyner–Ziv video coding, in IEEE International Conference on Image Processing (ICIP)* (San Antonio, USA, 2007). September
24. J Ascenso, C Brites, F Pereira, *Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding, in 5th EURASIP Conference on Speech and Image Processing* (Multimedia Communications and Services, Smolenice, Slovakia, 2005). June
25. IH Witten, E Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. (Morgan Kaufmann, San Francisco, 2005)
26. WEKA, (The University of Waikato, 2010). http://www.cs.waikato.ac.nz/ml/weka/, Accessed October
27. WW Cohen, Y Singer, *A simple, fast, and effective rule learner* (in Proceedings of the Sixteenth National Conference on Artificial Intelligence 1999, Orlando, USA, 1999), pp. 335–342. July 18–22
28. G Bjøntegaard, *Calculation of average PSNR differences between RD-curves, Presented at the 13th VCEG-M33 Meeting* (Austin, TX, 2001). April
29. G Sullivan, G Bjøntegaard, *Recommended simulation common conditions for H.26 L coding efficiency experiments on low-resolution progressive-scan source material. ITU-T VCEG, Doc. VCEG-N81, 14th meeting (Santa Barbara, USA, 2001).* September
30. JVT Test Model Ad Hoc Group, *Evaluation Sheet for Motion Estimation", Draft version 4* (2003). February

doi:10.1186/1687-6180-2012-135
Cite this article as: Corrales-García *et al.*: Wyner–Ziv to Baseline H.264 Video Transcoder. *EURASIP Journal on Advances in Signal Processing* 2012 2012:135.