Video Compression for Camera Networks: a Distributed Approach

Marco Dalai and Riccardo Leonardi

Department of Electronics for Automation,
University of Brescia, Via Branze 38, 25123, Brescia - Italy

Abstract

The problem of finding efficient communication techniques to distribute multi-view video content across different devices and users in a network is receiving a great attention in the last years. Much interest in particular has been devoted recently to the so called field of Distributed Video Coding (DVC). After briefly reporting traditional approaches to multi-view coding, this chapter will introduce the field of DVC. The theoretical background of Distributed Source Coding (DSC) is first concisely presented and the problem of the application of DSC principles to the case of video sources is then analyzed. The topic is presented discussing approaches to the problem of DVC both in single-view and in multi-view applications.

Key words: MVC, Distributed Source Coding, Distributed Video Coding, Multicamera Systems, Wyner-Ziv Coding
PACS:

1 Introduction

A problem that is invariantly involved in any application of networked cameras is the problem of coding and transmission of the video content that has to be shared among users and devices of the network. For this task, video compression techniques are employed to reduce the bandwidth required for communication, and possibly to efficiently store the video sequences on archival devices. Video coding has been growing in the last decades as a fundamental field of research in multimedia technology, since it enables the advent of highly modern devices and applications that would need otherwise to manage a huge amount of uncompressed data. In the first considered setting, historically, video coding was concerned with the problem of compressing as much as possible single video sequences. Many progresses have been made toward this end from the H.261 video coding standard ([40]) until the latest developments of the H.264/Advanced Video Coding (AVC) standard ([34,27]) and its extensions.

While the performance of these codecs in terms of compression efficiency has been growing continuously in these decades, only in recent years there has been an increasing interest in investigating more general video coding problems that are inherently motivated by the explosion of consumer-level technology. For example, more attention is now being paid to error resilient video coding for error-prone channels ([13]) and to scalable video coding to deal with different display devices in broadcasting scenarios ([41]). Similarly, an emerging interest motivated by appealing applications is the field of multi-view video coding. With the advent of camera networks and camera arrays, indeed, new applicative perspectives such as 3D Television or free-viewpoint Television appear nowadays as feasible targets of the next forthcoming future.

The problem of multi-view video coding is thus being studied with increasing interest. Accordingly, an extension of H.264/AVC like approaches to multiple camera systems has been proposed in the literature ([21,42]) and is being considered by the standardization bodies ([19]). These methods combine different compression tools into specialized architectures, which try essentially to exploit the redundancy in video sequences to compress the data by means of predictive coding. In a nutshell, while single source video coding concentrates on temporal predictions between frames of a same sequence, multi-view video coding tries to extend the idea to also consider existing spatial disparity between frames of different sequences.

In order to apply predictive coding between different views, obviously, the encoder must have access to the different video sequences. This implies that communication must be enabled between the cameras or that, alternatively, all the cameras are connected to a joint encoder that exploits such a redundancy. In certain situations, like for example in large low power camera arrays, the communication of raw data between cameras may result in excessive power consumption or bandwidth requirements. In this perspective, the emerging field of Distributed Video Coding (DVC) has been proposed as an alternative framework for the efficient independent compression of video data from multiple cameras, that means, exploiting the redundancy without the need of inter-camera communication. The idea of DVC moves from information theoretic settings of the late '70s that demonstrate that it is possible in theory to separately compress correlated sources at their joint entropy rate provided a single joint decoder will be in charge of the decoding process.

The purpose of this chapter is to provide an introduction to the field of DVC in this multicamera context. The structure of the chapter is as follows. In Section 2, a brief description of classic video coding techniques is presented in order to better appreciate the different approach proposed by DVC. In Section 3 an introduction to the information theoretic field of DSC is provided in order to clarify the underlying concept of DVC. In Section 4, the first approaches to DVC in the monoview setting are described, which are then discussed in the more general multi-view case in Section 5. Section 6 concludes the chapter.

2 Classic Approach to Video Coding

In this section we aim at giving a very concise description of the techniques used in classic video coding to exploit the redundancy of typical video sequences. It is our intention to provide a high level description to establish a reference framework with respect to which DVC needs to be compared. The architectural complexity of standard video codecs has evolved from the first H.261 until the more recent H.264/AVC codec. Over the years, many tools have been included to improve the performance. However there was no real paradigm shift.

The video sequence is usually partitioned in Groups of Frames (GOP) that are processed with a certain predictive structure in order to exploit the temporal dependencies among frames. In Figure 1 an example of GOP structure of length 4 is shown that uses hierarchical B frames, which means frames the encoding of which is based on predictions from both sides, but different prediction structures can be used.

The encoding procedures for the frames of a GOP is essentially based on the use of motion compensated prediction and block-based transform coding. The block diagram of the encoding procedure is shown in Figure 2. Every frame to be encoded is first partitioned in macroblocks and the content of each macroblock is searched

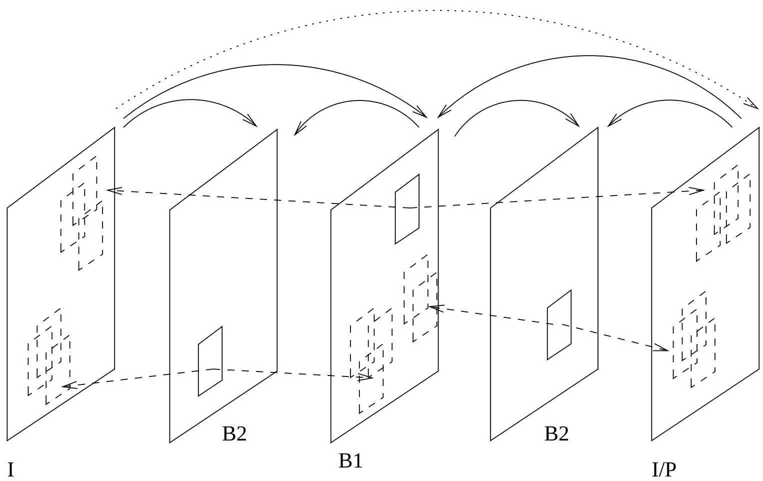


Figure 1. Predictive coding dependencies of a GOP in a classic video encoder. The Frames are encoded in different ways as Intra frames (I), Predicted frames (P) or Bidirectionally predicted frames (B), the latter modality being possibly hierarchically repeated on more levels. The dotted arrows represent motion searches that are used to find predictors in the reference frames. In this case the GOP length is 4 if the last frame is encoded as an Intra frame (I).

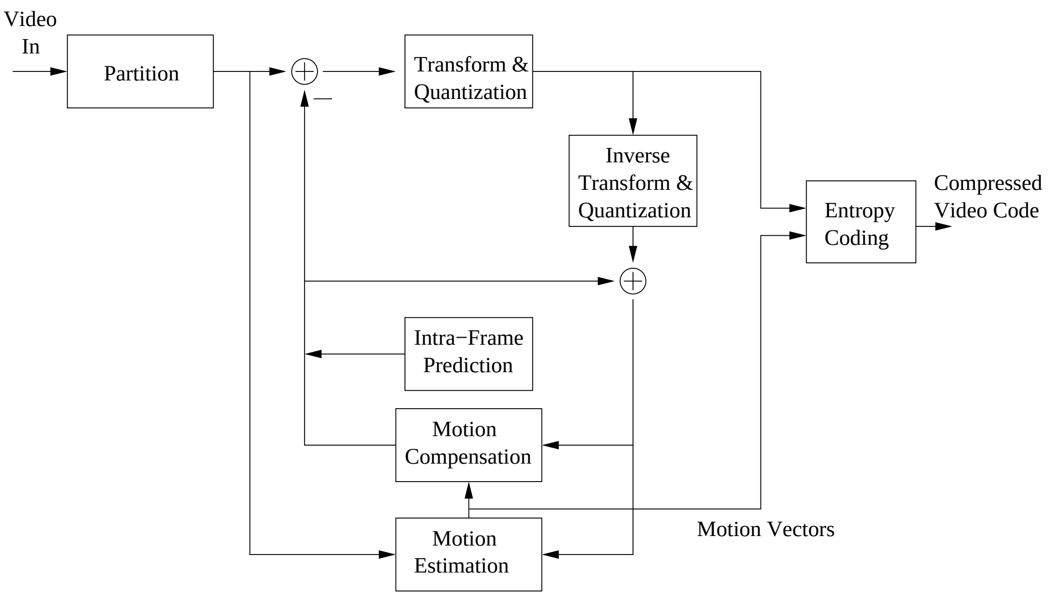


Figure 2. High level block diagram of the predictive encoding procedure in a classic video codec.

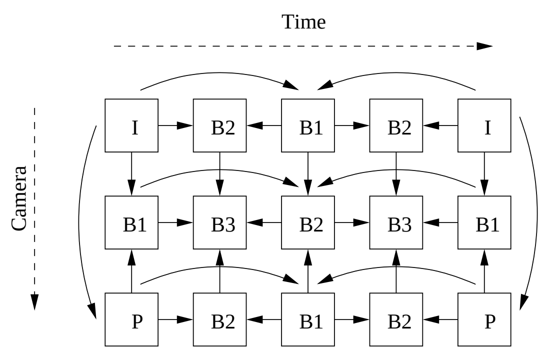
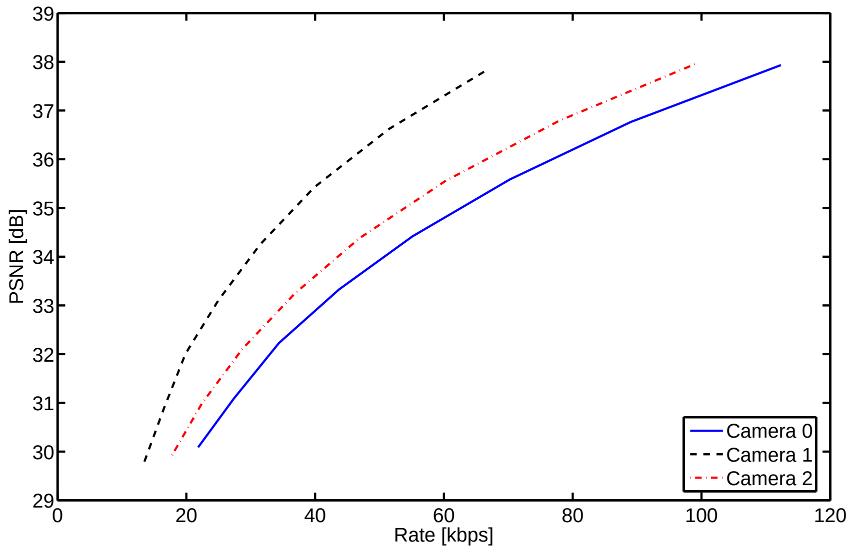


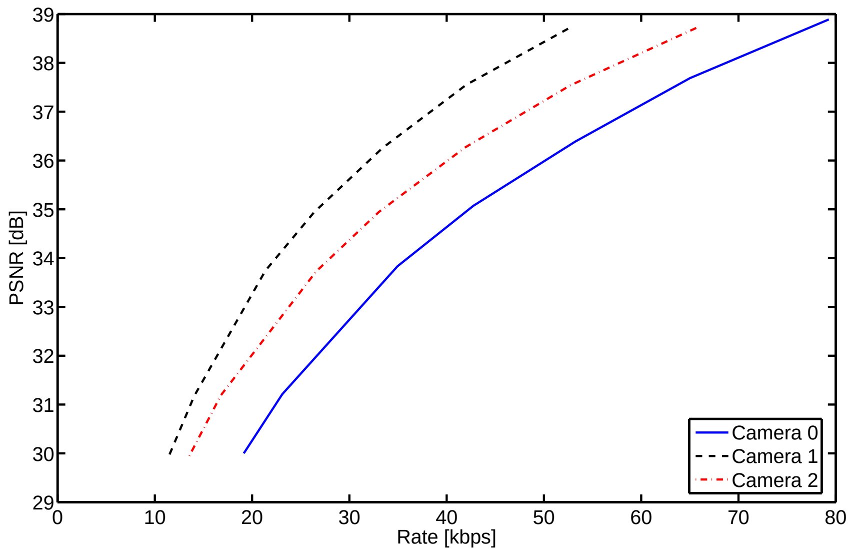
Figure 3. An example of predictive coding structure for a multiview system.

in reference frames, that is frames that have already been encoded. This is done so as to apply predictive encoding for the current macroblock; in order to avoid a drift between encoder and decoder, it is necessary to use a closed predictive loop, which means that the encoder replicates the decoder behavior. For every macroblock, the best found predictor, or an intra-frame prediction if no such similar blocks exist in reference frames, is subtracted from the original so that only the residual information can be then encoded. Typically, to remove any possibly remaining spatial redundancy, a spatial transform is applied prior to quantization. Both the indication of the used predictor, which also includes the motion information, and the code of the transformed and quantized blocks are entropy coded in order to compact the information as much as possible. This is the coarse description of the structure of a typical video encoder, where additional tools such as variable block size and sub-pixel motion search, deblocking or sophisticated intra-frame prediction can be applied jointly to further improve the performance (see [34]).

The encoder architecture described for single view video coding can be easily ex-



(a) Breakdancers - 256×192, 15 fps.



(b) Exit - 192×144, 25 fps.

Figure 4. Rate-Distortion operational curves of the multivew extension of the H.264/AVC, JMVC version 1.0 (see [19]). The plots refer to sequences taken from three cameras. Here, the sequence from Camera 0 is encoded in a traditional H.264/AVC single view way. The sequence from Camera 2 is encoded using Camera 0 as reference and the sequence from Camera 1 uses both Camera 0 and Camera 2 as references. It is clearly visible the advantage of inter-camera predictions in term of bitrate savings for a given target quality.

tended to the case of multiview coding. The main innovation needed is the predictive structure. An example of this is shown in Figure 3, where the same dependencies used in the temporal direction are also applied between cameras. The assumption here of course is that the cameras are placed so that contiguous cameras capture similar video sequences. The video coding community has been devoting a great deal of work in recent years to the understanding of specific multiview video coding problems (see for example [21,42]) and ongoing activities are leading to the definition of a multiview video coding standard ([19]).

The most important remark at this point, is the fact that the classic approach to video coding in a multiview setting involves the use of prediction between the frames of different sequences. This implies that the video content must be analyzed jointly by the encoder and thus that either the cameras can communicate between each other, or they all send the raw video content to a central encoder which has to jointly process the received data. In the next sections, the completely different approach proposed by DVC will be introduced. In this context, as mentioned in the Introduction, the predictive encoding is substituted by an independent encoding of the sources where existing correlation between them is only exploited at the decoder side. Before tackling the problem of DVC, however, it is necessary to understand the theoretical setting that is at its base, which is called Distributed Source Coding and is exposed in the next section.

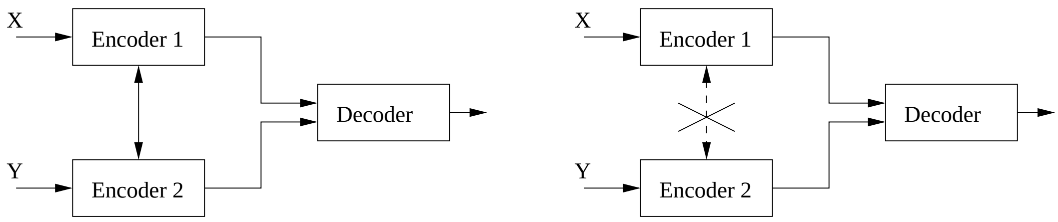
3 Distributed Source Coding

In this section we aim at providing a brief introduction to the information theoretic field of Distributed Source Coding, which is a necessary prerequisite to understand the ideas underlying Distributed Video Coding techniques. In its first and basic version, DSC is the study of the independent encoding of two correlated sources that are to be transmitted to a common receiver. This problem was first studied in a paper by Slepian and Wolf [29] in 1973; their famous result, together with the results obtained in a successive paper by Wyner and Ziv [36], yielded the development of DSC as a whole branch of information theory.

3.1 Slepian-Wolf Theorem

Following Slepian and Wolf, consider a situation where two correlated sources X and Y are to be encoded and transmitted to a single receiver. For the sake of simplicity we will deal here only with the case of discrete memoryless sources with a finite alphabet, and we will specify what are the different necessary hypotheses for ensuring the validity of the demonstrated results. We are interested in studying two different scenarios as shown in Figure 5.

Let us focus first on the case depicted in Figure 5(a). What we want to study is the amount of rate that is required in order to have a *lossless* transmission of X and Y from the encoders to the decoder, that is the rate required in order to let the decoder recover without distortion the values of X and Y. Note that in this scheme the two encoders are allowed to communicate between each other (assuming there is no limitation in the amount of information they can share). So, in this case we can



(a) Joint encoding (b) Distributed encoding

Figure 5. Two different scenarios for a two-source problem.

consider that both Encoder 1 and 2 know the values of both X and Y. In this case, it does not make much sense to consider the rates spent individually by each encoder, as the whole information may be sent by one of them. We are thus interested in studying the total rate required. It is very well known from the information theory that the minimum total rate that has to be spent in order to have a lossless encoding of the sources X and Y is their joint entropy $H(X, Y)$ ([11]).

Consider now the problem of encoding X and Y when the situation is as depicted in Figure 5(b). In this case the two encoders cannot communicate each other and they have to separately encode X and Y and send their codes to the common decoder. We ask what the admissible rates are for lossless communication in this case. It is clear that Encoders 1 and 2 could send X and Y using respectively a rate equal to $H(X)$ and $H(Y)$ bits. The total rate would be in that case $H(X) + H(Y)$ which is greater than $H(X, Y)$ under the hypothesis that X and Y are correlated. In this case, however, the decoder would receive part of information in a redundant way. Suppose that the decoder decodes first the value of Y; then, the value of X, being correlated with Y, is already "partially known" and the complete description received by Encoder 1 would be somehow redundant. We can thus guess that some rate could be saved by proper encoding. The surprising result obtained by Slepian and Wolf [29] is that not only the rate for X and Y can be actually smaller than $H(X)$ and $H(Y)$, but that there is no penalty in this case with respect to the case of Figure 5(a) in terms of total required rate. The only additional constraint in this case is that there is a minimum rate equal to the conditional entropy $H(X|Y)$ to be spent for X and a minimum rate equal to $H(Y|X)$ for Y, which represent the intuitive idea that every encoder must send at least the amount of information of its own source that is not contained in the other source. In particular, Slepian and Wolf formulated the following theorem for the case of memoryless sources.

Theorem 1 (Slepian-Wolf, 1973, [29]) *Let two sources X and Y be such that (X_1, Y_1), $(X_2, Y_2), \ldots$ are independent drawings of a pair of correlated random variables (X, Y). Then it is possible to independently encode the source X and the source Y at rates R_X and R_Y respectively, so that a common receiver will recover X and Y with arbitrarily small probability of error, if and only if $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ and $R_X + R_Y \geq H(X, Y)$.*

The above theorem holds for memoryless sources as considered in the Slepian'

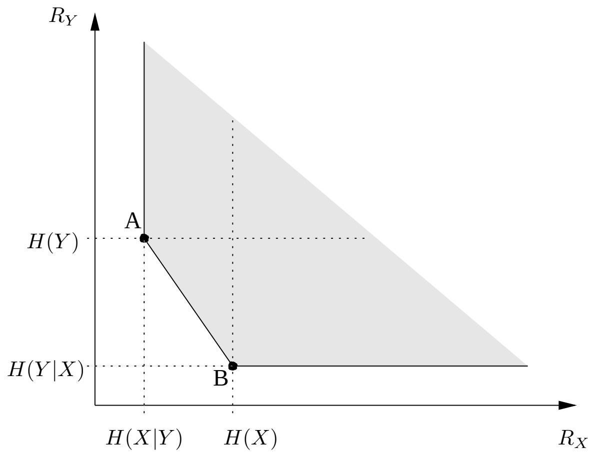


Figure 6. Slepian-Wolf region.

and Wolf's paper. Few years later, Cover [10] extended the theorem to the more general case of multiple stationary ergodic sources, giving a simple proof based on the asymptotic equipartition property, i.e. the Shannon-McMillan-Breiman theorem [20]. In this more general case the theorem is obviously reformulated by substituting entropies with entropy rates in the inequalities.

The set of all (R_X, R_Y) rate pairs satisfying the theorem is called *achievable region* and it is shown in Figure 6. The two points labeled with A and B in the figure represent an important special case of the theorem. Consider for example point A. This point in the region represents a situation where the source Y is encoded in a traditional way using a rate R_Y equal to its own entropy $H(Y)$, while the source X is encoded using the minimal rate $R_X = H(X|Y)$. This problem is of particular interest, and it is usually referred to as coding X *with side information* Y *at the decoder*.

It is important to clarify that Theorem 1 considers rate pairs (R_X, R_Y) such that the decoder will losslessly recover X and Y *with arbitrarily small probability of error*. This means that the encoding is considered to operate on blocks of n symbols, and that for sufficiently large n the probability of having an error in the decoding phase can be made as small as desired. It is worth noticing that in this sense there is a penalty in the case of distributed encoding with respect the case of joint encoding. In the latter case, in fact, by using variable length codes it is possible to encode the two sources X and Y to a total rate as close as desired to the joint entropy $H(X, Y)$ even with a probability of decoding error exactly zero.

3.2 A Simple Example

It is useful to clarify the idea that is behind the Slepian-Wolf theorem by means of a simple example. Suppose that the two sources X and Y are such that Y is an integer uniformly distributed in [0,999] and that $X = Y + N$, with N uniformly distributed on the integers between 0 and 9. Consider the number of decimal digits necessary to describe X and Y in the case of a joint encoding as shown in Figure 5(a). We easily note that it is possible to encode Y using three decimal digits and then, given the value of Y, encode X with the only digit required to describe the value of $N = X - Y$. For example, if $X = 133$ and $Y = 125$, then Y is simply encoded with its own representation, and X is encoded by specifying the value $N = 8$. So, a total of 4 decimal digits allows to encode both values of X and Y.

Suppose now that the two encoders cannot communicate, as depicted in Figure 5(b). Then, supposing Y is encoded with all its 3 decimal digits, we are faced with the encoding of X with side information Y at the decoder, as in point A of Figure 6. This time, the encoding of X cannot be based on the value of N, since N cannot be computed by Encoder 1, which ignores the value of Y. Still, it is possible to encode X using only one decimal digit if the value of Y is known to the decoder. The trick is to encode X by simply specifying the last digit. In our case, for example, where $Y = 125$, since $Y \leq X \leq Y + 9$, knowing that the last digit of X is 3 suffices to deduce that $X = 133$. So, the knowledge of Y at Encoder 1 does not impact the rate required for X, and a total rate of 4 digits allows to describe both X and Y.

Now note that all points on the segment between A and B in Figure 6 are achievable. These points can be obtained by properly multiplexing points A and B in time, but it is also possible to actually construct a *symmetric* encoding of X and Y. In our simple toy example, this can be shown by demostrating that it is possible to encode X and Y using two decimal digits for each source. The trick is to let Encoder 2 send the last two digits of Y, and Encoder 1 send the first and the third digits of X. With our example, where $X = 133$ and $Y = 125$, Encoder 2 sends '_25' and Encoder 1 sends '1_3'. It is not difficult to realize that for the receiver this information, together with the constraint $Y \leq X \leq Y + 9$, is sufficient to recover that $X = 133$ and $Y = 125$.

This simple example reveals an interesting insight on the real essence of the the Slepian-Wolf theorem. With real sources, obviously, the encoding techniques must be usually much more complicated. Neverthless, the main idea is maintained, the principles used both for the encoding with side information or with symmetric rates are “only” a generalization of the described approach to more general and practical situations. In the next section, with a meaningful example it is shown that channel codes can be used for distributed source coding in the case where X and Y are binary sources correlated in terms of Hamming distance.

There is a close connection between the Slepian-Wolf problem and channel coding. This relation was first noticed by Wyner in [35], where the author used an example of binary sources to present an intuitive proof of the Slepian-Wolf theorem. In this section, an example of distributed encoding of binary sequences is provided with a description of the use of channel codes for this problem. The discussion parallels the example given in Section 3.2. We assume the reader has familiarity with the basic theory of algebraic channel codes (see [7] for an introduction). A more detailed analysis of the use of channel codes for DSC can be found in [24] and [15].

In this example we consider two sources X and Y that are 7-bits words, where the correlation is expressed by the fact that the Hamming distance between X and Y is at most 1, that is, they differ for 1 bit at most. As a reference, note that the joint encoding of X and Y requires 10 bits. For example one can raw encode Y with 7 bits and then encode X by specifying the difference with respect to Y with 3 bits, since there are 8 possible choices.

Consider the case of coding X with side information Y at the decoder, or equivalently, when 7 bits are used for the encoding of Y. We show here that by using a proper channel code it is still possible to encode X using only 3 bits. We use the systematic Hamming (7,4) code. The generating matrix G and the parity check matrix H of this code are respectively

$$G = \begin{pmatrix} 1\,0\,0\,0\,0\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix} \quad H = \begin{pmatrix} 0\,1\,1\,1\,1\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0 \\ 1\,1\,0\,1\,1\,0\,1 \end{pmatrix}. \tag{1}$$

Let us first consider how the code is used in channel coding. In that case, the Hamming code, in the encoding phase, maps a 4-bits word w into 7-bits codeword $c_t = w \cdot G$, which is transmitted on the channel and received, say, as c_r. The decoding phase computes then the so called *syndrome* $s = c_r \cdot H'$. By construction, the matrices H and G satisfy $G \cdot H' = \mathbf{0}$. Thus, if the codeword is received without errors, one has $s = c_r \cdot H' = c_t \cdot H' = w \cdot G \cdot H' = \mathbf{0}$. If instead an error word e is added to the codeword during the transmission, then one has $s = c_r \cdot H' = (c_t + e) \cdot H' = (w \cdot G + e) \cdot H' = \mathbf{0} + e \cdot H' = e \cdot H'$. It is easy to note that if e has *Hamming weight* equal to 1, i.e. one bit is corrupted in the transmission, then s equals the column of H indexed by the position of the error. Thus, s allow to identify the position of the error and thus to correct the codeword c_r to restore c_t and thus recover w. Thus, the notorious fact that the (7,4) Hamming code can correct one error.

Now, let us focus on the use of this code for coding X with side information Y at the decoder. The correlation assumption between X and Y can be modeled by saying that $X = Y+e$, the word e having Hamming weight at most 1. Suppose now that Y is known at the decoder. We encode X by computing its three-bits syndrome $s_X = X \cdot H'$ and sending it to the decoder. There, we can compute $s_Y = Y \cdot H'$. Using s_X and s_Y the decoder can compute $s = s_X + s_Y = X \cdot H' + Y \cdot H' = (X + Y) \cdot H' = e \cdot H'$. Again, assuming e has Hamming weight at most one, the decoder can detect the position of the difference between X and Y and then, since Y is given, deduce X.

With a smart trick, furthermore, it is also possible to use the Hamming code to encode the two sources X and Y in a symmetric way using 5 bits for each one. We split the generating matrix G in two submatrices G_1 and G_2 by taking respectively the first two rows and the last two rows of G, that is

$$G_1 = \begin{pmatrix} 1\,0\,0\,0\,0\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 0\,0\,1\,0\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix}. \tag{2}$$

This two matrices are used as generating matrices for two codes C_1 and C_2, which are subcodes of the Hamming code with parity check matrices

$$H_1 = \begin{pmatrix} 0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,1\,0 \\ 1\,1\,0\,0\,0\,0\,1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,0\,1\,0\,0\,1 \end{pmatrix}. \tag{3}$$

The encoding of X and Y is done by computing $s_X = X \cdot H_1'$ and $s_Y = Y \cdot H_2'$. It is possible to show that the decoder, given the pair of syndromes s_X and s_Y can uniquely determine the words X and Y using the constraint that they differ by at most one bit. In fact, suppose on the contrary that there is also a different pair of words $(\bar{X}, \bar{Y})$ satisfying the same syndrome and the same distance constraints. Then, as $s_X = s_{\bar{X}}$, $X + \bar{X}$ has null syndrome, and it is thus a codeword for C_1; for similar reasons, $Y + \bar{Y}$ is a codeword for C_2. Thus, as C_1 and C_2 are subcodes of the Hamming code, $(X + \bar{X}) + (Y + \bar{Y})$ is a codeword for the Hamming code. But $(X + \bar{X}) + (Y + \bar{Y}) = (X + Y) + (\bar{X} + \bar{Y})$ has at most weight equal to 2 and, since the Hamming code has distance 3, the only word with weight smaller than 3 is the null word. So, $(X + \bar{X}) = (Y + \bar{Y})$, but $(X + \bar{X})$ is in C_1 while $(Y + \bar{Y})$ is in C_2. As the rows of G_1 and the rows of G_2 are independent (being G_1 and G_2 submatrices of G), the only intersection of C_1 and C_2 is the null word, that is $X = \bar{X}$ and $Y = \bar{Y}$. So, there is a unique solution, which means that X and Y can be recovered at the decoder.

Few years after the publication by Slepian and Wolf [29], Wyner and Ziv [36] obtained an important result for the problem of lossy coding with side information at the decoder, that is the case when Y is available at the decoder and the source X does not have to be recovered perfectly, but within a certain distortion. For lossy source coding, as it is known, the theoretical bounds are described through the computation of the *rate distortion function* [6,14,11]. We do not want to enter into the details of the rate distortion theory, the interested reader can refer to [6] for this. Here we just recall that for the single source problem, supposing that X is an i.i.d. source with marginal p.d.f. $q(x)$ and $d(x, \hat{x})$ is the distortion measure between a reproduction symbol $\hat{x}$ and the original value x, the rate distortion function is given by

$$R(D) = \min_{p \in \mathcal{P}(D)} I(X; \hat{X}) \tag{4}$$

where $I(\cdot;\cdot)$ is the *mutual information* and $\mathcal{P}(D)$ is the set of all conditional probability functions $p(\hat{x}|x)$ such that $E[d(X, \hat{X})] \leq D$, that is, the expected value of the distortion is at most D. In the case when there is side information Y available to both encoder and decoder the rate distortion function simply changes to (see [6])

$$R(D) = \min_{p \in \mathcal{P}(D)} I(X; \hat{X}|Y) \tag{5}$$

where P is now the set of all $p(\hat{x}|x, y)$ such that $E_{x,y,\hat{x}}[d(x, \hat{x})] \leq D$. Wyner and Ziv obtained a characterization of the rate-distortion function when the side information Y is only available at the decoder [36].

Theorem 2 (Wyner-Ziv, 1976, [36]) *Let two sources X and Y be as in Theorem 1, and let $q(x, y)$ be their joint distribution. The rate distortion function for the encoding of X with side information Y available to the decoder is*

$$R^{WZ}_{X|Y}(D) = \inf_{p \in \mathcal{P}(D)} [I(X; Z) - I(Y; Z)] \tag{6}$$

where Z is an auxiliary variable and $\mathcal{P}(D)$ is the set of all $p(z|x)$ for which there exists a function f such that $E[d(X, f(Y, Z))] \leq D$.

A detailed analysis of the theorem is out of the scope of the present work and we only add some comments that may be interesting for the reader. In addition to prove the above theorem, in [36] the authors observe he following facts:

(1) In the general case, for positive distortion values D there is a penalty in the rate distortion bound when the side information is not available to the encoder with respect to the case when it is. This means that the result of Slepian and Wolf does not extend to the lossy case. It has been shown more recently [38], however, that the rate loss is bounded by a quantity that equals half a bit per sample for the case of the quadratic distortion $d(x, \hat{x}) = (x - \hat{x})^2$.

(2) Theorem 2 is valid in a broader setting than to the limited case of finite alphabet sources [37]. In particular it is valid if X is a Gaussian source and $X = Y + N$ with N Gaussian with variance σ_N^2 and independent of Y. In this particular case, under the euclidean distortion criterion, the rate distortion function can be computed analytically and one has

$$R_{X|Y}^{WZ}(d) = \frac{1}{2}\left(\log \frac{\sigma_N^2}{d}\right)^+, \tag{7}$$

where $(\cdot)^+$ is the positive part function, i.e. $(x)^+ = \max\{0, x\}$. In this particular case, the rate distortion function is the same obtained for the case when Y is also available to the encoder, and hence the Slepian-Wolf result does extend to the lossy case.

4 From DSC to DVC

The application of DSC principles to the problem of video coding was independently proposed by two different groups from Stanford University [1] and from UC Berkeley [25]. Starting from these pioneering works, DVC has now become an active field of research, see for example [17,22] for an overview. Singularly enough, while DSC lies in the so called field of *multiterminal* or *multiuser* information theory, DVC was initially concerned with the application of the DSC ideas to the problem of encoding single video sequences. After the first published works that considered single source video coding, the field rapidly grew, and DVC is now intended as the application of DSC to the more general problem of multi-source (or multi-view) video coding. In this section the DVC approach to single source coding is presented, which is a meaningful introduction to the topic. Most of the ideas will be easily reused in multi-camera contexts later on in the chapter, and the most important differences will be discussed in detail in the next sections.

4.1 Applying DSC to Video Coding

Let us focus on the single source video coding problem. The use of DVC in this context was proposed as an alternative solution to the traditional video coding techniques, mainly centered around the use of motion compensation in a prediction loop inside the encoder. There are different motivations for this alternative proposal. The most important motivations are probably the shift of the computational complexity from the encoder to the decoder and an expected higher error robustness in presence of error-prone communications. In short, as already described in the Section 2, classic video coding techniques such as H.264/AVC (see [34,27] and references therein for details) adopt motion estimation at the encoder for motion compensated prediction encoding of the information contained in the frames of a sequence. This leads to codecs with very good rate distortion performance but at the cost of computationally complex encoders and of fragility with respect to transmission errors over the channel. The computational complexity of the encoder is high due to the motion search that is required in order to properly perform predictive coding from frame to frame. Fragility, then, is due to the drift caused by error propagation through the prediction loop. Therefore, the fragile source coding approach must be followed by powerful channel coding for error resilience. In addition further processing must be designed often at the receiver to adopt effective error concealment strategies. DSC techniques are intrinsically based on the idea of exploiting redundancy without performing prediction in the encoding phase, and leaving to the decoder the problem of deciphering the received codes using the correlation or redundancy between the sources. For these reasons the use of DSC in single source video coding has appeared as a possible solution for a robust encoding with the possibility of flexibly allocating the computational complexity between encoder and decoder.

Consider a video sequence composed by frames $X_1, X_2, \cdots, X_N$, let R and C be the number of rows and columns in every frame X_i, and let $X_i(r, c)$ represent the pixel value at location (r, c) in a frame. It is clear that the frames of a video sequence are very redundant, i.e., a video is a source with strong spatial and temporal memory. Spatial memory means that the if we model the frames as stochastic processes, the random variables representing pixel values that are spatially close in the same frame are correlated. Temporal correlation means that consecutive frames are very similar, the only difference being usually small movements of the objects, unless a scene change, a flash or some similar "rare" event occurs. We will refer to *intra-frame* correlation for the spatial correlation and to *inter-frame* correlation for the temporal one. Later on in this chapter, when referring to multi-camera systems, we will call for obvious reasons *intra-sequence* correlation the correlation within a sequence and *inter-sequence* correlation the correlation between different sequences.

The classic techniques for video coding, starting from H.261 and MPEG1 until the most recent developments such as H.264/MPEG-4 AVC, exploit the correlation of a video sequence by combining the use of transforms, for removing the intra-frame correlation, and the use of motion compensated prediction for dealing with the inter-frame correlation. We are mostly interested here in this second aspect, i.e. the motion compensated prediction between frames. In the basic situation we can consider the problem of encoding a frame X_i when the previous frame X_{i-1} has already been encoded and it is available in an approximated form, say as $\tilde{X}_{i-1}$, at the decoder. In this case, what a classic video coding technique would do is to estimate the motion field M_i between the reference frame $\tilde{X}_{i-1}$ and X_i; then, by "applying" this motion to the frame $\tilde{X}_{i-1}$, obtain an approximation of X_i, say $X'_i = M_i(\tilde{X}_{i-1})$. The encoding of X_i is then performed using the prediction, and, instead of directly encoding X_i, the motion field M_i and the prediction error $e_i = X_i - M_i(\tilde{X}_{i-1})$ are coded. The encoding of e_i is usually achieved by transform coding so as to exploit the remaining intra-frame correlation. This is only a very coarse descriptions of modern video codecs, as an accurate fine-tuning of tools is necessary to achieve high Rate-Distortion performance as proposed in the different standards (MPEG1/2/4). Nevertheless, the main point is sufficiently described in this form: in classic video coding standards a frame is encoded by applying motion compensated prediction from previously encoded frames. At the decoder, the motion field is applied to the available reference frame (or frames) and used to generate the prediction, which is then successively updated with the received prediction error.

The use of DSC for the problem of video coding is based on the idea that we can consider the frames (or portions of frames) of a video sequence as different correlated sources. So, when a frame X_i has to be encoded based on a previously encoded frame X_{i-1}, by invoking Slepian-Wolf' and Wyner-Ziv' results, we can consider $\tilde{X}_{i-1}$ as a side information that is known to the decoder and that need not be known at the encoder. This way the coding technique for X_i exploits the correlation with $\tilde{X}_{i-1}$ in the decoding phase without using prediction in the encoding step.

This is the very basic idea under DVC, which has then to be further refined in order to lead to concrete coding schemes. Note that the DSC scenario considered in this case is the problem of source coding with side information at the decoder and, for video sequences, one is usually interested in lossy compression. For this reason DVC is often also referred to as Wyner-Ziv (WZ) coding of video and, more generally, we call WZ coding whatever encoding technique based on the presence of side information at the decoder. By extension, we will often refer to the bits associated to a WZ encoding as the WZ bits and we will often refer to the part of video already available at the decoder as Side Information (SI), in some cases referring to a whole frame or in other cases to portions of frames or even to groups of frames.

4.2 PRISM Codec

In this section we will describe the so called PRISM codec, proposed by Puri and Ramchandran [25] in 2002. The encoding approach for the frames of the video sequence is shown in Figure 7 for a single GOP.

Let again $X_1, X_2, \dots, X_n$ be the frames. The first frame X_1 is encoded in an *intra-mode* using for example a block based approach similar to the ones used in JPEG [33]. For the following frames, a block based process is considered. The generic frame X_i is divided in 8×8 pixel blocks; let X_i^k be the k-th block, and let $X_i^k(r, c)$ be its pixel values. The following chain of operations is then performed:

Encoding

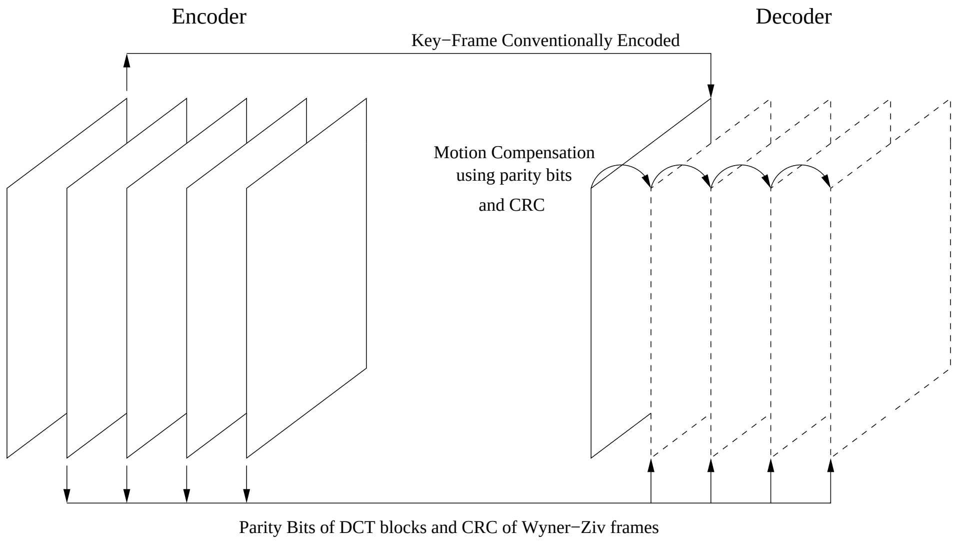


Figure 7. Scheme of frame encoding and decoding in PRISM.

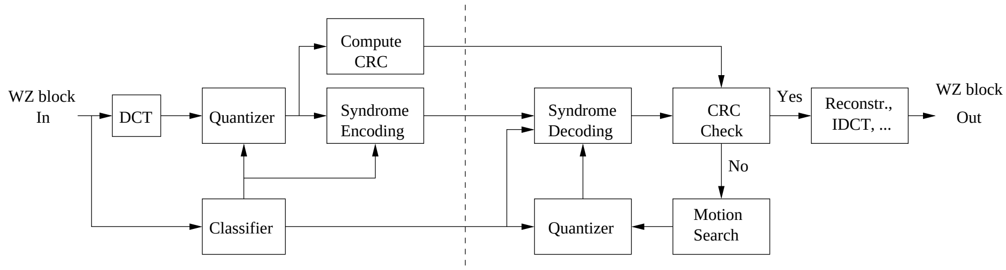


Figure 8. Block diagram of the encoding and decoding processes in PRISM.

(1) Every block is analyzed so as to estimate its correlation with the content of the previous frame: block X_i^k is compared with X_{i-1}^k and the sum of absolute differences is computed, i.e., $\epsilon_i^k = \sum_{r,c} |X_i^k(r,c) - X_{i-1}^k(r,c)|$. The value ϵ_i^k is an estimate of the correlation between the current block and the previous frame with low computational cost.

(2) Depending on the value of ϵ_i^k every block is classified in one of the three following categories:
   (a) if ϵ_i^k is smaller than a given threshold, say $\epsilon_i^k \leq \epsilon_{\min}$, then block X_i^k is classified as a *SKIP* block;
   (b) if ϵ_i^k is larger than a given threshold, say $\epsilon_i^k \geq \epsilon_{\max}$, then block X_i^k is classified as an *INTRA* block;
   (c) otherwise, block X_i^k is classified as a *WZ* block. WZ blocks are further divided in 16 different classes $C_1, C_2, \ldots, C_{16}$, depending on their ϵ_i^k value, so that the encoder can operate differently on blocks exhibiting different level of correlation.

(3) A flag is transmitted indicating the type of block (SKIP/INTRA/WZ) and the code of the block is then emitted:
   (a) If X_i^k is a SKIP block, no further information is encoded. SKIP mode means that the decoder replaces the block with the same position block in

the previous frame.

(b) If X_i^k is an INTRA block, it is encoded in a traditional way through transform coding followed by a Run-Amplitude (RA) code such as in JPEG. The decoder can thus decode this type of blocks without any reference to other frames.

(c) If X_i^k is a WZ block, instead, the index specifying the associated class is added. The block is then encoded, as indicated in Figure 9, in the following way. A DCT transform is applied followed by a quantization tuned depending on the class. The Least Significant (LS) bits of the quantized low pass coefficients are encoded in a distributed fashion using a trellis code. Then, refinement bits of the low frequency coefficients are encoded (to reach a given target quality) whereas high pass coefficients are encoded with a classic RA procedure. Furthermore, a 16-bits CRC is computed on the quantized low pass coefficients; the use of the CRC will be clarified later.

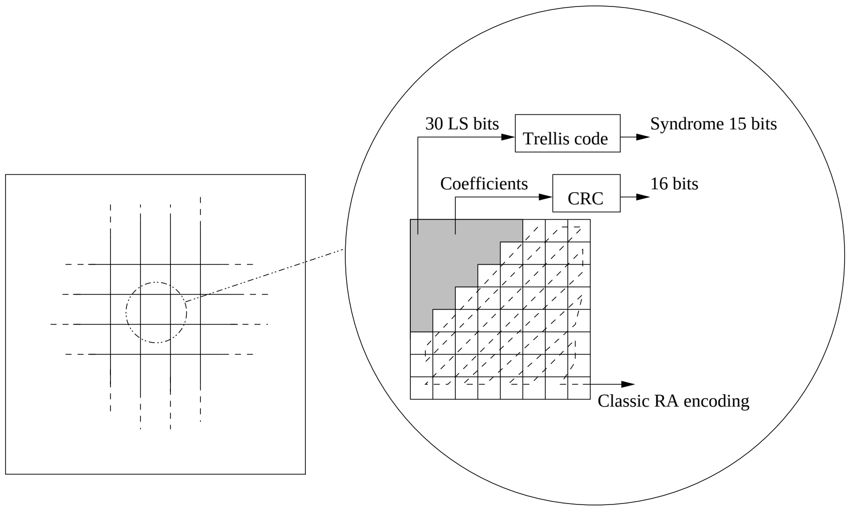


Figure 9. Encoding procedure for the WZ blocks in PRISM.

The above explained procedure for the encoding of the WZ frames is not completely specified since it is not clear how the values of the used parameters are established. We refer here to the thresholds $\epsilon_{\min}$ and $\epsilon_{\max}$, to the quantization parameters and even to how the C_j, $j = 1, \ldots, 16$ classes are determined based on the value of ϵ_i^k. All such values are established by properly training the codec on test video sequences. These details are not relevant for the purpose of the present chapter and we refer to [26] for more details.

Decoding (WZ blocks)

The decoding for a block X_i^k is performed by combining a sort of motion estimation and a WZ decoding in the following way:

(1) For a WZ block X_i^k in the frame X_i, different blocks in the frame X_{i-1} around

the position of X_i^k are tested as side information at the decoder.

(2) Every candidate block is used as SI; it is transformed and quantized using the specific quantizer for the class containing X_i^k and the LS bits of the low frequency coefficients are extracted and used as side information for a WZ decoding that uses the parity bits of the correct block X_i^k sent by the encoder.

(3) The CRC-16 is computed on the so obtained "corrected" side information coefficients. If the CRC matches, the decoding is considered correct and the procedure stops, otherwise another block is selected from the previous frame and the process is repeated from step (2). If no available SI block allows to match the CRC, then it is not possible to reconstruct the low pass coefficients and a concealment strategy must be adopted.

(4) When the procedure for low frequency coefficients is terminated, the high frequency coefficients are decoded in a traditional mode and inserted to fill the DCT transform of the block. The inverse transform is then applied to obtain the pixel values of the block.

4.3 Stanford Approach

With respect to the PRISM codec, the Stanford architecture adopts different choices for the application of WZ principles to the case of video sequences (see [1,16]). The main difference is that the frames of the sequence are considered as a whole, and the WZ coding is applied to a whole frame and not to single blocks. So, we can actually identify some WZ frames that are completely encoded in a WZ way, without differentiating the processing on a block by block basis. The key idea, in this case, is to estimate the motion at the decoder and create a complete SI frame to be corrected as a whole by the WZ decoding.

The coarse idea is to split the frames of the sequence at the encoder dividing them in two groups. Let again $X_1, X_2, \ldots$ be the frames; in the simpler version of the codec, odd-indexed frames $X_1, X_3 \ldots$ are encoded in an intra-mode conventional way, that is as a sequence of images, while even indexed frames $X_2, X_4 \ldots$ are encoded in a WZ fashion. At the decoder, the intra-coded frames are used in order to create an approximation for the WZ frames by motion compensated interpolation. Then, the parity bits are used to "correct" these approximations and recover the frames. This idea is graphically represented in Figure 10.

This general idea gave rise to many research papers that proposed different variations on this scheme and the description we give in the following part of this section is obtained by combining interpretations of details from different authors (see for example [22] for an overview). It is necessary to clarify in advance one particular characteristic of this architecture, which is the need of a feedback channel from the decoder to the encoder (see [8]). This feedback channel is used in the process of WZ decoding in order to request more parity bits from the encoder if the received

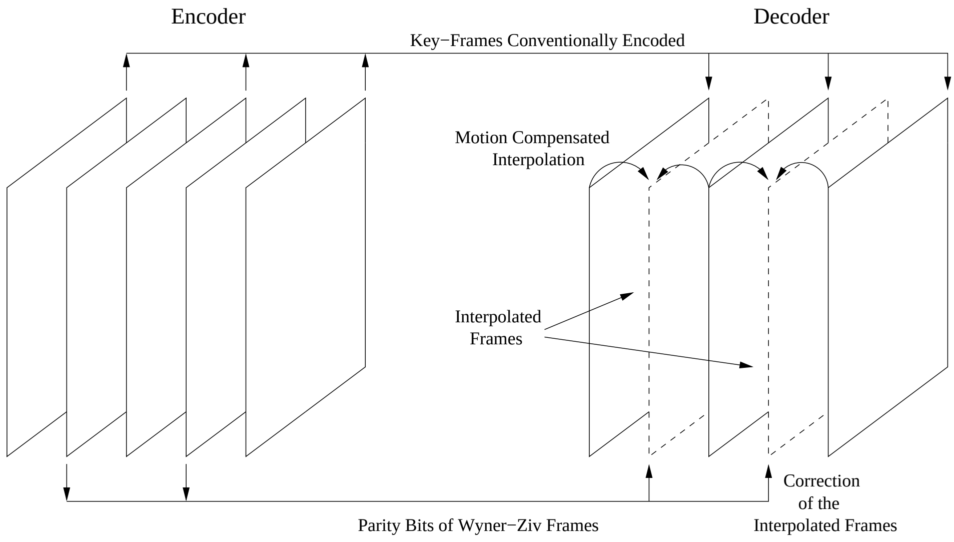


Figure 10. Scheme of frame encoding and decoding in the Stanford approach.

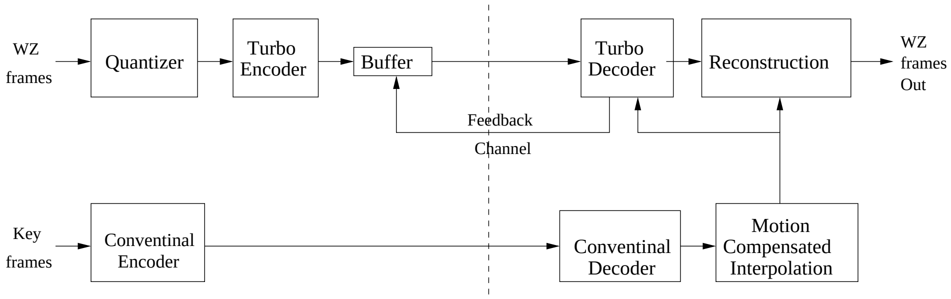


Figure 11. Block diagram of the encoding and decoding processes in the Stanford codec.

ones are not sufficient to properly decode the source. Even if, from a theoretical point of view, this feedback channel could be removed by introducing higher functionalities at the encoder side with the drawback of an increased complexity (see [9]), up to now it still not clear what the achievable performance are in terms of balancing between required encoder complexity and rate-distortion performance.

Encoding of WZ frames

The encoding of a WZ frame, say X_{2n}, is performed in the following way:

(1) A block based DCT transform is applied to the frame and a quantization mask is applied to the transformed coefficients. These coefficients of the blocks are then reordered in frequency bands and the bitplanes of every band are extracted and prepared for WZ encoding.
(2) The extracted bitplaes are fed into a turbo encoder [1] and the resulting parity bits are stored in a buffer. These parity bits are ready for transmission to the

[1] Other implementations use LDPC codes, see for example [3], but there is no essential difference for the purpose of this chapter.

decoder, which will request them iteratively until it has enough bits to perform the WZ decoding.

The encoding procedure as shown above is notably simple. We now present the decoding operation for the WZ frames

Decoding of WZ frames

The decoding process for a WZ frame X_{2n} is as follows

(1) Let X'_{2n-1} and X'_{2n+1} be the two reconstructed key frames adjacent to X_{2n}. By applying a motion compensated interpolation, X'_{2n-1} and X'_{2n+1} are used for the construction of an approximation Y_{2n} of X_{2n}, which is the Side Information for the WZ decoding.
(2) The SI is assumed to be a noisy version of the original frame. In particular, it is assumed that every DCT coefficient of Y_{2n} differs from the corresponding coefficient of X_{2n} for an additive noise with a hypothetical distribution (usually a Laplacian distribution, see [16]). Given the value of the side information coefficient, thus, it is possible to compute the probability of every bit of the original coefficient to be '0' or '1'. These probabilities are used for the WZ decoding.
(3) The WZ decoding operates bitplane-by-bitplane, starting with the most significative one and using, for every bitplane, the previously decoded ones to compute the bit probabilities. The probabilities are fed to the turbo decoder as "channel values" of the information bits. The turbo decoder, using a feedback channel, asks for parity bits from the encoder which sends them by progressively puncturing the parity bits in a buffer. The turbo decoder tries to decode the channel values with these parity bits to recover the original bitplane. If the turbo decoding process fails, more parity bits are requested and the process is repeated until the turbo decoder is able to correctly recover the bitplane. Note that, even if not discussed in the first Stanford publications, it is necessary to adopt ad hoc tools in order to detect the success/failure of the decoding process, see for example [18].
(4) After all bitplanes have been recovered, the best estimate X'_{2n} of X_{2n} is constructed by taking for every DCT coefficient, the expected value, given its quantized version and the SI, under the assumed probabilistic model. The DCT block transform is then inverted and the the sequence of WZ frames is interleaved with the sequence of key frames.

It is worth saying that the operation performed in step (1), that is, the motion compensated interpolation, plays an important role in this architecture and hides a lot of details that can greatly impact the performance of the system (see for example [5]). In particular, some variations on the scheme deal with the possibility of performing an extrapolation based on past frames rather than an interpolation. Furthermore, it has been noted in the literature that if the encoder sends a coarse description of the original frame, then it is possible to greatly improve the motion estimation and thus the quality of the generated SI. This aspect will be rediscussed later.

4.4 Remarks on DVC

There are a number of important comments that are helpful in understanding the relations and the differences between DSC and DVC, and that can thus serve as guidelines for the design of a concrete DVC system. The first difference between DSC and DVC is found in the a priori assumptions on the correlation between information sources. In the theoretical setting for DSC, Slepian-Wolf and Wyner-Ziv theorems are based on the assumption that the encoders and the decoder are completely aware of the statistical correlation between the sources. This assumption is critical since the encoding and decoding operations are strongly based on it. In particular, as usually happens with Information Theoretic results, there are assumptions of ergodicity and stationarity of the sources and it is assumed that the length of the blocks to be encoded can be increased as desired. In the field of DVC, as described above, the sources are interpreted as frames or portions of frames of a video sequence or, in the case of multicamera systems, of possibly different video sequences. A first comment is that it is difficult to match the characteristic of video sequences or portions of video sequences with those of a stationary ergodic source. However, the most important remark is that in DVC the correlation between the sources is in general not known and it must be somehow estimated.

The term “correlation” itself is not immediately clear in the case of DVC. In DSC “correlation” simply refers to the joint probability density functions of the sources. In the case of DVC, we can reasonably think of a dependency between sources that can be separate in two factors. The first factor includes the geometrical displacements and deformations, that differ from frame to frame, due to the motion or to the relative position between cameras. The second term includes what is usually considered the real “innovation”, that is the uncovered regions and the differences between the chromatic values of the same physical regions in different frames due to the difference in the sampling point, illumination, noise and so on. That is, there is a correlation in the sense of geometrical deformation that reflects the 3D difference between scenes, and there is another correlation in the sense of differences that cannot be compensated by means of geometric transformations. For this reason we can interpret the WZ decoding in DVC as a composition of two basic operations, that is a *compensation* used to match the WZ data to the SI data, and a *correction* operation to recover the original WZ data numerical values from the approximation obtained by compensation. These operations are also performed in a classic video codec, but it is important to understand that, while they do not have an essentially different role in a classic approach, they do in a distributed setting. The reason is that in a classic codec, where both the original data to be encoded and the reference are available, it is easy to find the best possible compensation and then to encode the prediction error. In a distributed setting, instead, the encoding is performed using only the WZ data and compensation has thus to be performed at the decoder. It is somehow easy to encode in a WZ fashion the original data supposed the compensation is already done, but it is much more difficult to perform the compensation at the decoder, since the original data is not available.

The compensation is thus the first crucial difficulty of DVC, and it is still not well understood how this problem could be efficiently solved. In the PRISM codec, the compensation process is actually bypassed by means of a looped correction process with a CRC-check for detecting the successful decoding. [2] Thus, PRISM is not really interested in the problem of estimating the correct motion or the disparity, but it only relies on the hypothesis that a good prediction will be available and that this prediction will allow the WZ decoding. When this does not happen however, for example because the parity bits do not suffice, both the parity bits and the CRC are unusable. That is, not only the information sent to correct is not successfully used, but there is not even an estimation of a possible compensation to apply to the reference to approximate the WZ data. In the basic implementation of the Stanford codec, the compensation is performed using only the key frames and it is thus not based on information on the WZ data. As said, in further developments of the codec it was considered that the encoder could send a coarse description, basically a low-pass or high-pass version, of the WZ frame in order to help the compensation process ([2]). This solution, however, has not been studied in detail in terms of efficiency. More precisely, it has never been theoretically studied as a distributed coding strategy, but only as a trick to apply *before* considering the proper distributed source coding problem. This can be accepted, but it must be clear that if those coarse descriptions of the images allow to find the disparity using classic estimation techniques, such as block matching, then those images are themselves correlated, and encoding them in a classic fashion is surely suboptimal. It is thus reasonable to consider that, using this technique, a concrete portion of the similarity between the images is not necessarily exploited. The reader is referred to [12] for a more detailed discussion of this point.

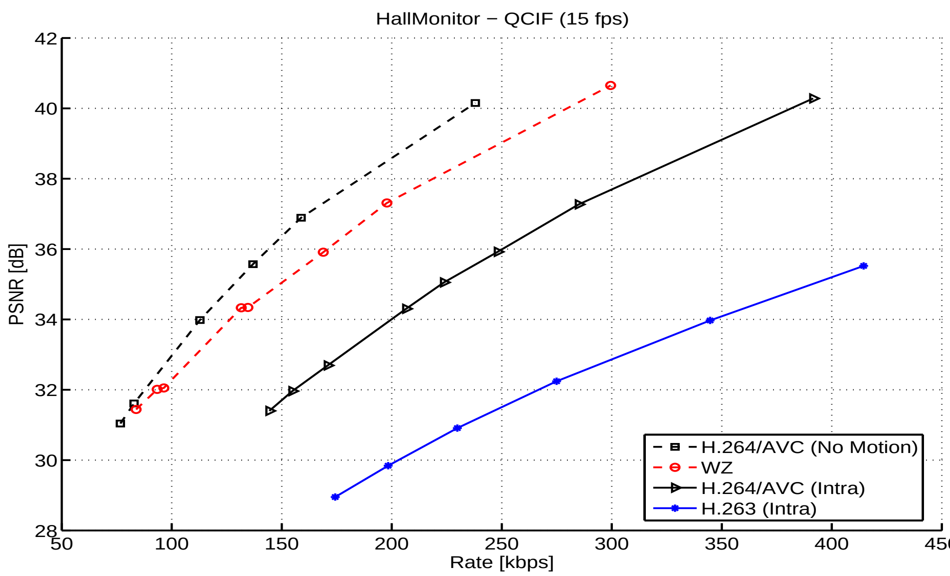
Furthermore, every solution proposed for the compensation problem indirectly impacts the possible solutions for the correction problem. Consider the two single camera architectures described in the previous sections. PRISM tries to perform the compensation jointly with the correction while the Stanford decoder first completely compensates a frame and then corrects it. Both choices have *pros* and *cons*. The PRISM codec has the advantage that it allows to use only one frame as a reference and guess the motion during the WZ decoding, while the Stanford solution needs to use more than one reference frame and separately estimate the motion. The

[2] From a theoretical point of view, it is interesting to note that the CRC is not really different from the syndrome of a channel code. Thus, one may object that instead of using a hard decision with a CRC it would be better to increase the channel code correction capability.

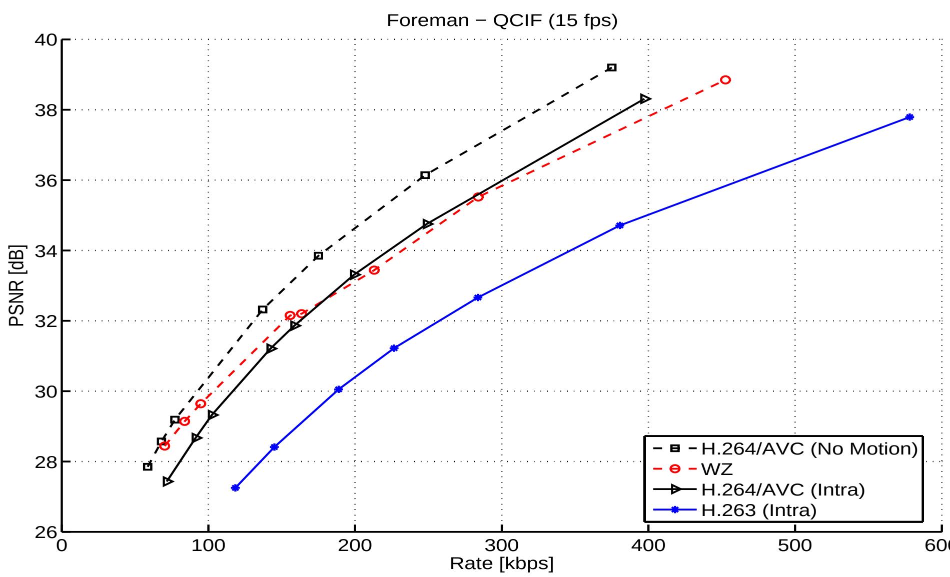
first choice is more DSC oriented, since the motion is itself part of the information that should ideally be encoded in a distributed way. However, the motion search embedded in the WZ correction phase prevents PRISM from using WZ codes on large blocks, since an exhaustive search of the motion within the combination of all possible motion fields would not be feasible. The Stanford scheme, on the contrary, allows to use more powerful channel codes since the correction is applied to the whole frame, and also takes advantage of the large data size to exploit the "average" correlation. The requirement for PRISM of performing the correction independently on small blocks is a great penalty since it is surely more difficult to efficiently estimate the correlation separately for each block, and it is not possible in this case to invoke the large numbers statistics.

This is related to the second great difficulty in DVC, which is the problem of rate allocation. As already explained, even after the compensation is performed, the correlation between the WZ data and the SI, and thus the required rate for the correction operation, is uncertain. This has an important impact on the allocation of rate, since an underestimate of the required rate leads to failures in the correction process. It is not clear up to now how to have a gradual degradation of the quality of the decoded data with the reduction of the rate. There is usually instead a threshold, below which the decoding fails and returns useless information, and above which the decoding is instead successful, but the quality does not increase further with the rate. This problem is clearly perceived in the codecs presented. The Stanford solution bypasses the problem by means of a return channel. This is clearly an unfair solution in the context of the DSC strictly speaking. It can of coarse be a reasonable approach to specific applicative problems, but it changes the theoretical setting of the problem. In the PRISM codec, the problem is noticed in that, by simulations, the real performance of the coding of the INTER blocks is low, often lower than that of the INTRA blocks. It should be thus clear that the two proposed codecs are first important steps toward the realization of DVC systems, improvements are being obtained by different research groups in these years (see [22]), but many fundamental problems still remain to be solved. In order to provide a comparison between the performance of a distributed codec and the performance of classic codecs, Figure 12 shows the results obtained with the software developed within DISCOVER, a European Project, funded under the European Commission IST FP6 programme [3].

[3] The DISCOVER software started from the so-called IST-WZ software developed at the Image Group from Instituto Superior Técnico (IST), Lisbon-Portugal (http:amalia.img.lx.it.pt), by Catarina Brites, João Ascenso, and Fernando Pereira.



(a)



(b)

Figure 12. Rate distortion performance of the WZ codec developed within the european project DISCOVER. Here the results refer to sequences Hallmonitor (a) and Foreman (b) QCIF format taken at 15 frames per second ([3]).

5 Applying DVC to Multi-View systems

In the previous section an introduction on DVC has been given using two important examples from the literature on single source DVC. In this section we aim at providing an introduction to the use of DVC techniques in the context of multi-view video coding. The idea of using DVC for multi-camera systems appeared soon as

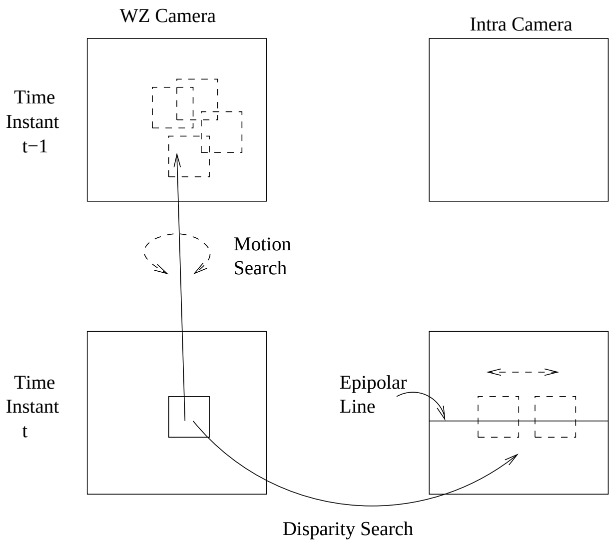


Figure 13. Search of correspondences in the PRISM decoder in a Multi-View setup.

an appealing option with respect to the use of H264 MV extensions. Compared to single camera DVC, multicamera DVC is clearly more representative of the application of DSC to video applications, since there is indeed in this context the need to compress different correlated sources without communication between the encoders. The two motivations for single camera DVC, that is the computational complexity flexible allocation and the error resilience, are still interesting for multicamera problems, but the possibility of exploiting the inter-camera correlation without requiring communication between cameras is in many cases of broader interest. This can lead to some important differences in the practical implementation of multi-camera DVC systems with respect to single-camera systems, as we shall discusse later. What is however not different is the general philosophy behind the underlying coding paradigm.

5.1 Extending Mono-View Codecs

There is of course a great variability of possible scenarios in multi-camera systems, and it is thus not possible to provide a general treatment of multicamera DVC without spacifying which types of configurations are considered. For example one may have many cameras positioned in regularly spaced points, all with the same importance, that have to communicate with one single receiver or one can have a system based on different types of cameras that have thus to operate in different ways. A configuration that is often considered is the case where some cameras - usually called intra cameras - encode their video sources in a classic sense, and these sources are used at the decoder as SI for other cameras - called WZ cameras - that operate in a WZ fashion.

The two architectures described in the previous section for single-source DVC can be used also in the context of multicamera DVC. The Side Information for the WZ (block of) frames can be composed in this case by both frames from the intra cameras and intra-coded key frames of the WZ cameras. There is not obviously a single possible choice to do this, and we only provide here an example based on publications by the same research groups that proposed the original single-view codecs (see for example [31,32,39]).

The PRISM codec can be extended, as proposed in [31], to deal with multi-camera systems. The extension can be easily defined on a system with two cameras, an intra-camera and a WZ camera. The intra-camera provides side information for the WZ camera. This means that for the decoding of a WZ frame at the generic time instant t, the side information available to the decoder is not composed in this case only by the previously decoded frame of the same camera, but also by the frame from the intra-camera at the same instant t. This implies a minimal modification in the codec with respect to the single camera, the only difference being that, for every WZ block, additionally to the usual motion search of PRISM, there is a disparity search used to detect estimators from the different view, rather than from the past frame, see Figure 13. It is worth noticing that if the relative position of the cameras is known, it is possible to reduce the region of the disparity search in the intra-view frame, using the multiview geometry, to a segment over the epipolar line associated to the position of the WZ block.

The codec proposed at Staford can be extended to multi-view scenarios as well. In [39] the authors propose the use of a generalized version of the original codec to the case of large camera arrays. The idea is that in a large camera array some of the cameras can be used as intra-cameras and the remaining ones as WZ cameras. The encoding of the WZ frames then proceeds as in the case of the single-camera codec, while the decoding is different for what concerns the generation of the Side Information. Indeed, instead of using only an interpolation between key frames to construct the approximation, it is possible to use the intra-camera views to generate a rendered view of the WZ frame, which is an additional approximation available to be used for the WZ decoding, see Figure 14. As already mentioned for the motion interpolation used in the single view codec, there are a lot of technical details that would need to be discussed with respect to the rendering method. It is first useful to say, here, that in practical contexts there is usually a higher correlation between frames of the same sequence rather than between frames of different sequences. In any case, the technique used for the generation of the SI has a dramatic effect on the quality of the obtained approximation and thus on the performance of the codec (see [4]). It is worth noticing that these details are however not usually specified in papers dealing with DVC, and this contributes to the difficulty in properly evaluating the performance of different implementations of the architecture.

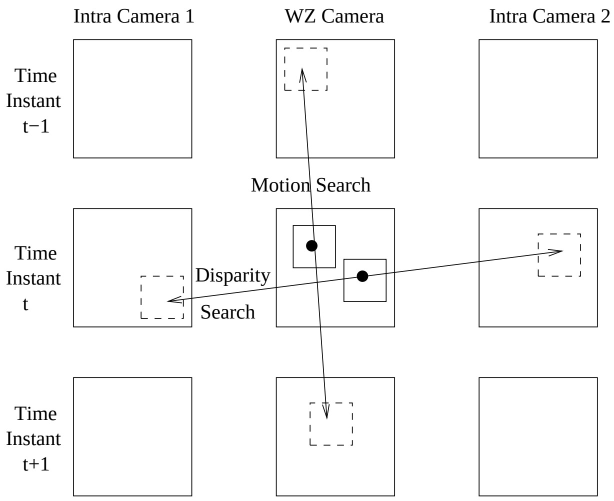


Figure 14. Stanford's codec in the multiview setting.

5.2 *Some Remarks on Multi-View problems*

The architectures for single and multiple camera systems based on PRISM and Stanford's codecs have been intensively studied by many research groups over the last years. There are so many details that can be implemented in different ways, or variations that can be easily incorporated in the same schemes, that a complete discussion of all their possible combinations is impossible here. We think that it is however necessary to clarify that the problem of finding a satisfying approach to DVC is still unsolved. Both PRISM and the Stanford codec suffer problems of practical usability in real contexts. The main problems have already been presented for the single camera systems, but in the multi-view case they assume a really different importance. Recall that we mentioned basically two problems, that are the difficulty in performing compensation at the decoder and in allocating, at the encoder, the required rate for the correction of data, due to the "unknown" correlation between the WZ data and the side information. In the single camera systems these problems can somehow be mitigated if a trade-off is allowed in the requirements. That is, since there is a unique source to be compressed, the use of DVC is motivated by computational complexity allocation and error resilience. If a certain complexity is allowed at the encoder and more importance is left to error robustness, then it is possible to perform in the encoding phase a number of operations that can allow the encoder to estimate the motion, so as to facilitate the decoder task and also appropriately estimate the rate required in the decoding phase. The encoding-decoding technique for the correction phase can then still be based on WZ principles, but at least the rate allocation and the compensation problems are somehow mitigated. In a multi-view scenario there is no such possibility unless the relative position of the various cameras and the scene depth field are known. Given that different sources are available at different encoders, it is in no way possible to balance computational-complexity with disparity estimation or to improve the estimation on the required rate.

Consider for example the PRISM codec. Suppose a given block on a WZ frame has no good predictor in the side information frames due to occlusions. In a single camera system this situation can be detected if a certain amount of operations - such as a coarse motion search - can be performed by the encoder. In a multi camera system instead it is not possible to distinguish if a certain block is present in the intra-camera side information or not. This of course implies that it is not possible to efficiently apply DSC principles at the block level. This problem may be partially solved with the Stanford approach, since the WZ decoding operates at the frame level and thus exploits the "average correlation" with the SI in a frame. The Stanford codec, however, suffers the problem that the compensation at the decoder is completely performed before the WZ decoding, and can thus be based only on a priori information on the geometrical deformations to be applied to intra-camera views to estimate the WZ camera view. This implies that the solution cannot be flexible to realistic cases. As for the monoview case, one may consider the possible encoding of a coarse low or high pass description of the WZ frame to be sent from encoder to decoder and used for the compensation task. This however eludes somehow the real challenge of the application of DSC to multiple view video coding, since a great deal of work precisely consists in exploiting in a distributed fashion the geometrical similarities between different views.

6 Conclusions

In this chapter we have introduced the topic of DVC, which has been one the most studied ones in the field of video coding in the last years. The main difference between the classic coding techniques and DVC is that the predictive coding used in classic approaches is substituted in DVC by a completely different framework, where it is the decoder task to find similarities between already encoded portions of data. We have shown that channel codes can be used in the case of binary data, and we have also shown examples of video codecs that use channel codes as basic tools to apply distributed compression to some portions of the video data, after appropriate transform and quantization. As shown, the examples discussed for the case of single source coding are also meaningful in the case of multi-view systems, but different strategies can be investigated. An example of a different approach to the problem of distributed coding of multi-view images is given in the next chapter.

References

[1] A. Aaron, R. Zhang and B. Girod. Wyner-Ziv coding for motion video. *Asilomar Conference on Signals, Systems and Computers*, Pacific Groove, USA, 2002.

[2] A. Aaron, S. Rane and B. Girod. Wyner-Ziv video coding with hash-based motion-

compensation at the receiver. In *Proc. IEEE Int. Conf. on Image Proc.*, Singapore, October 2004.

[3] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret. The Discover Codec: Architecture, Techniques And Evaluation. *Picture Coding Symposium (PCS 2007)*, Lisboa, Portugal, November 2007.

[4] X. Artigas, F. Tarres, L. Torres. Comparison of Different Side Information Generation Methods for Multiview Distributed Video Coding. *International Conference on Signal Processing and Multimedia Applications SIGMAP*, Barcelona, Spain, July 2007.

[5] J. Ascenso, C. Brites and F. Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, July 2005.

[6] T. Berger. *Rate-distortion theory: A mathematical basis for data compression.* Prentice-Hall, Englewood Cliffs, 1971.

[7] R. E. Blahut, *Theory and Practice of Error-Control Codes*. Addison-Wesley, Massachusetts, 1983.

[8] C. Brites, J. Ascenso and F. Pereira. Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities. *14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, September 2006.

[9] C. Brites and F. Pereira. Encoder Rate Control for Transform Domain Wyner-Ziv Video Coding. In *Proc. IEEE Int. Conf. on Image Proc.*, San Antonio, Texas, USA, September 2007.

[10] T. M. Cover. A proof of the data compression theorem of Slepian and Wolf for ergodic sources. *IEEE Trans. Inform. Theory*, 22: 226–228, 1975.

[11] T. M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.

[12] M. Dalai and R. Leonardi. Minimal Information Exchange for Image Registration. In *Proc. 16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, August 2008.

[13] F. Dufaux and T. Ebrahimi. Error-resilient video coding performance analysis of motion JPEG2000 and MPEG-4. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, January 2004.

[14] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[15] N. Gehrig and P. L. Dragotti. Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes. *IEEE Comm. Letters*, vol 9, n° 1, pp. 61–63, January 2005.

[16] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero. Distributed Video Coding. *Proc. IEEE*, vol. 93, n° 1, pp. 71-83, January 2005.

[17] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, J. Ostermann. Distributed monoview and multiview video coding. *IEEE Signal Processing Magazine*, vol. 24, n^o 5, pp. 67 - 76, September 2007.

[18] D. Kubasov, K. Lajnef and C. Guillemot. A Hybrid Encoder/Decoder Rate Control for Wyner-Ziv Video Coding with a Feedback Channel. *Int. Workshop on Multimedia Signal Processing*, Crete, Greece, October 2007.

[19] A. Vetro, P. Pandit, H. Kimata, A. Smolic, Y.-K. Wang. Joint Multiview Video Model JMVM 8.0. ITU-T and ISO/IEC Joint Video Team, Doc. JVT-AA207, April 2008.

[20] B. McMillan. The basic theorems of information theory. *Ann. Math. Stat.*, vol. 24, n° 2, pp. 196–219, 1953.

[21] P. Merkle, K. Müller, A. Smolic and T. Wiegand, Efficient Compression of Multi-View Video Exploiting Inter-View Dependencies Based on H.264/MPEG4-AVC. In *IEEE International Conference on Multimedia and Exposition (ICME 2006)*, Toronto, Ontario, Canada, July 2006.

[22] F. Pereira, C. Brites, J. Ascenso, M. Tagliasacchi. Wyner-Ziv video coding: a review of the early architectures and relevant developments. *IEEE International Conference on Multimedia & Expo (ICME 2008)*, Hannover, Germany, June 2008.

[23] S.S Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. *IEEE Trans. Inform. Theory*, vol. 49, n° 3, pp. 626–643, 2003.

[24] S.S Pradhan and K. Ramchandran. Generalized coset codes for distributed binning. *IEEE Trans. Inform. Theory*, vol. 51, n° 10, pp. 3457–3474, 2005.

[25] R. Puri, and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proc. Of 40th Allerton Conf. on Comm., Control and Comp.*, Monticello, October 2002.

[26] R. Puri, A. Majumdar and K. Ramchandran, PRISM: A Video Coding Paradigm with Motion Estimation at the Decoder. *IEEE Trans. on Image Process.*, vol. 16, n^o 10, pp. 2436–2448, Oct. 2007.

[27] Iain E. G. Richardson. *H.264 and MPEG-4 Video Compression*. Wiley & Sons, UK, 2003.

[28] D. Schonberg, S. S. Pradhan, and K. Ramchandran. Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources. *Data Compression Conference*, Snowbird, UT, USA, March 2004.

[29] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, vol. 19, n° 4, 471–480, 1973.

[30] B. Song, E. Tuncel and A. K. Roy-Chowdhury. Towards a Multi-terminal Video Compression Algorithm by Integrating Distributed Source Coding with Geometrical Constraints. *Journal of Multimedia*, vol. 2, n° 3, pp. 9–16, June 2007.

[31] C. Yeo and K. Ramchandran. Robust distributed multi-view video compression for wireless camera networks. In *Proc. of SPIE. Visual Communications and Image Processing (VCIP 2007)*, vol. 6508, 2007.

[32] C. Yeo, J. Wang, K. Ramchandran. View Synthesis for Robust Distributed Video Compression in Wireless Camera Networks. In *Proc. IEEE Int. Conf. on Image Proc. 2007 (ICIP 2007)*, San Antonio, Texas, USA, September 2007.

[33] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, vol. 14, n° 4, pp. 31-44, 1991.

[34] T. Wiegand, G. J. Sullivan, G. Bjntegaard and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, n° 7, pp. 560–576, July 2003.

[35] A. D. Wyner. Recent results in the Shannon theory. *IEEE Trans. Inform. Theory*, vol 20, n° 1, pp. 2–10, 1974.

[36] A. D. Wyner and J. Ziv. The rate distortion function for source coding with side information at the receiver. *IEEE Trans. Inform. Theory*, vol. 22, n° 1, pp. 1–11, January 1976.

[37] A. D. Wyner. The rate-distortion function for source coding with side information at the decoder-II: General sources. *Inform. Contr.*, vol. 38, n° 1, pp. 60–80, 1978.

[38] R. Zamir. The rate loss in the Wyner-Ziv problem. *IEEE Trans. Inform. Theory*, vol. 42, n°6, pp. 2073–2084, 1996.

[39] X. Zhu, A. Aaron and B. Girod. Distributed compression for large camera arrays. In *Proc. IEEE Workshop on Statistical Signal Processing*, St Louis, Missouri, USA, October 2003.

[40] “Video Codec for audiovisual services at px64 kbit/s”, ITU-T Recomendation H.261, Geneva, Switzerland, 1990.

[41] “Special issue on scalable video coding-standardization and beyond”, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, n° 9, September 2007.

[42] ITU-T Rec. & ISO/IEC 14496-10 AVC, Advanced video coding for generic audiovisual services, 2005.